A novel Classification-based Hybrid IDS

Oscar Rodas H., Jose A Alvarez Aldana and Gerardo Morales
RLICT / Universidad Galileo
7Av. Final Calle Dr. Eduardo Suger Zona 10
Guatemala, Guatemala
Email: {orodas,josealfredo1515,gmorales}@galileo.edu

Stephane Maag
Institut Mines-Telecom / Telecom SudParis
CNRS UMR 5157
9, rue Charles Fourier, 91011, EVRY Cedex, France
Email: Stephane.Maag@telecom-sudparis.eu

***Abstract*—For years, the IDS industry has worked on bringing a solution to anomaly-based attacks on computer networks. The main concerns related to the IDS implementations have been: low detection rate of anomaly-based attacks that causes low usability and high rate of false positives that causes low acceptability. Researchers in the field of IT have proposed different approaches using numerous techniques to improve these rates. This paper presents an approach based on the problematic faced by networks when anomaly-based attacks emerge. The approach proposes a novel framework based on analyzing real-time information and classifying traffic in a binary way, *legitimate* or an *intrusion*. Log information will be correlated with the same customized format, filtered and stored in mongoDB by Collaborative Intrusion Detection System (CIDS). This CIDS will be the main entity for examining correlation in the log information. While correlating the information with the help of mongoDB, the framework will be able to rapidly determine the existence of anomaly-based intrusions and will notify the different entities in the network about the intrusion. The framework will dynamically adapt to the kind of traffic present in the network. The case study for this paper will be based on the typically Brute Force Attack launched to any server running the SSH protocol. The SSH protocol was chosen because it is the default remote protocol sysadmins use for remote connections on networks worldwide.**

I. Introduction

Private and public networks, like The Internet, have become a place where people do business, check email, perform transactions and bring everyday life activities to a virtual environment, making it important to follow the three pillars of Information Security, which are, *confidentiality, integrity and availability*. When a transaction fails to meet this kind of compliance, it means that the network is vulnerable to a security breach. To mitigate these kinds of problems, companies need a comprehensive security system on their network. Protection should start with a component that runs analysis on traffic, coming in and out of it, and that also has the capacity to classify this traffic as legitimate or intrusion. Classifying traffic should be handled by an Intrusion Detection System (IDS) on the network. The IDS industry has developed different approaches for the intrusion detection in networks.

Commonly, researchers working in Information Security [8] [6] classify IDSs into two different types depending on the kind of implementation:

- **HIDS (Host-based Intrusion Detection System)**: This IDS is installed on every device (host) that is connected to a network. Its purpose is to capture and analyze network data from the device.
- **NIDS (Network-based Intrusion Detection System)**: This IDS is either installed at the network gateway, with the purpose of analyzing traffic on all devices in the network. The IDS will then inspect packets and will be able to determine an attack before it reaches the endpoints.

Various factors should be considered when deciding the best place in the network to implement an IDS solution. *Chen et al.* [6] state that HIDSs imply high costs of installation and maintenance. NIDSs imply a single point of failure existing in the architecture designed for implementing security on the network. NIDSs also imply a single resource on the network that spends a lot of time processing and analyzing packet content and could degrade the system performance. Deciding the right spot to implement the IDS will determine if it should be installed either at a centralized point or at a particular host. NIDS, installed on a gateway, router or server are able to analyze all the traffic incoming or outgoing to the network. HIDS installed on particular host (computer, laptop, etc.) are able to analyze its particular traffic. In both implementations the IDS will always be analyzing traffic and sending alarms if any intrusion attack has been detected.

IDS can also be categorized based on the approach they use to determine if traffic can be considered as "legitimate" or "anomalous" [8] [7]. These different approaches are based on two main techniques. *a.) Misuse Detection*: This detection is based on the use of signature-based intrusion detection. The packet is first examined to determine if it can be categorized as an anomaly. If the packet does not comply with any of the signatures, then it can be categorized as "legitimate" traffic. *b.) Anomaly Detection*: This detection is based on the use of a profile of a user, server or network that after some criteria is denoted as "legitimate". Starting from this profile, every other traffic that does not comply with the profile will be considered an anomaly.

Even though, the IDSs in the recent years were improved by the intrusion detection industries, they still have two major concerns [12]: *Low detection rate of anomaly-based intrusion attacks* that causes low usability and *High rate of false positives* that causes low acceptability.

There has been numerous approaches using different techniques to try to improve these rates. Recently, researchers have proposed approaches using machine learning techniques to de-

tect intrusions [13] [4] [9] [12] [5]. Their main contribution is the implementation of hybrid approaches that detect anomaly-based intrusions after filtering traffic that complied with signature-based techniques. Other researches have approaches that correlate information from different traffic observations to determine anomaly-based attacks after long periods of time [14].

Regarding new kinds of attacks like coordinated attacks, *Collaborative Intrusion Detection (CIDS)* Systems have been a solution to address this kind of problem. A survey by *Zhou et al.* [15] describes the need of having various IDSs in a network collaborating to correlate log information from different parts of a network. Analyzing correlated data from different logs will make easier the detection of coordinated attacks like large-scale stealthy scans, worm outbreaks and Distributed Denial-of-Service (DDos) attacks. Contrary to having isolated IDSs that only monitor a limited portion of the network and make extremely difficult the detection of these kind of attacks.

The main contributions of this paper after examining different approaches to detect anomaly-based attacks are:

- *Definition of a novel framework to detect anomaly-based intrusion attacks*. The framework is based on anomaly-based intrusion detection systems (A-IDS) embedded in different areas of a distributed system. All A-IDS send log information to a central repository with mongoDB [3], a database, for further collaborative anomaly-based intrusion attack detection. This correlation is handled by a Collaborative Intrusion Detection System (CIDS) analyzing all data in real time.
- *A more precise traffic classification methodology*. While correlating all the information of the different IDSs installed on the network, the framework enables to classify traffic in a binary way, as *legitimate* or as an *intrusion*.

The remainder of the paper is organized as follows. In Section II, we review the related work of other researchers and how they motivated and help the research presented in this paper. In Section III, we describe our novel framework and in Section IV we present the experimental studies and the results obtained with our testbed. We conclude and give perspectives in Section V.

II. Related Work

A. Machine Learning Techniques

In [7], *Garcia-Teodoro et al.* described the techniques, systems and challenges in anomaly-based network intrusion detection. They mention the existence of the IDWG (Intrusion Detection Working Group) that was based on the CIDF (Common Intrusion Detection Framework)[1].

Tsai et al. [13] examined 55 related studies based on the type of classifier. The review promoted the research of combining hybrid and ensemble classifiers to create better IDS that would increase the detection rates on anomalous traffic going through networks. Furthermore, it also mentioned the importance of feature selection for analyzing traffic and reducing the workload for analysis by IDSs.

[1]http://www.ietf.org/wg-descriptions/cidf.desc.txt

B. Hybrid Techniques

The proposal of *Aydin et al.* [4] describes a hybrid approach, concatenating anomaly-based and misuse intrusion detections systems. This approach delivered results that increased the anomaly detection rate in SNORT [10]. Our approach differs from the way of implementing the hybrid IDS. This implementation is based on a single Network IDS, which makes the a solution to be based on a single point of failure. Additionally, the implementation was only tested on an *IDEVAL* [1] testbed and not on real-time traffic, unlike our approach.

C. Other Techniques

One of the challenging issues of network and system management operators is the every day dealing with huge amount of alerts generated [11]. These alerts can obfuscate the appearance of anomalous behaviors. The use of alert correlation has become one of the most popular resources used for detecting collaborative attacks. Our framework deals with the problem of scalability, it can handle as much network entities as the distributed system needs to be consider secure.

Chen et al. [6] describe an efficient network intrusion detection approach based on a Lightweight Network Intrusion Detection system (LNID). Our approach differs from their solution that is based on a single Network IDS.

Zhou et al. [15] mentions that the major problem is *Coordinated Attacks*, which are a large-scale stealthy scans, worm outbreaks and distributed denial-of-service (DDoS) attacks that occurred in different networks at the same time. In their survey they focus on two important factors for detecting anomaly-based intrusions:

- *Collaborative Intrusion Detection Systems (CIDSs)* and their architecture.
- *Alert Correlation Algorithms* to detect this kind of attacks.

It is mentioned that every attack has a common stage in which all traffic either comes from the same source (stealthy scans or worm outbreaks) or the traffic goes to a particular host or server (DDoS attacks). Researchers describe that improving the techniques for HIDSs or NIDSs can help detect anomaly-based traffic in a more precise manner. Also, that there is a need to correlate all the logs of the different IDSs on large-scale networks, like the Internet, to determine new kind of attacks. Our approach enhances this feature by having a CIDS that will only be managing the correlation of the logs and, in this first approach, our correlation is based on the source IP address of the attacks.

Table 1 highlights that neither a Centralized nor Hierarchical approach for system architecture is the best for collaborative intrusion detection. Our approach takes into consideration the advantages of the Centralized and Hierarchical approach mentioned before and implements a hybrid architecture. This hybrid architecture is based on hierarchy and centralization.

Hierarchy will have two different phases to detect anomalous traffic and will be implemented as follows:

- **Phase 1:** IDSs will be installed across the network and will be responsible for intrusion detections at their particular areas (host or network).
- **Phase 2:** All log information will be sent to a CIDS that will manage the correlation of the information to determine the possibility of coordinated attacks present in the network and to classify traffic as *legitimate* or as an *intrusion*.

Centralization is used for managing the correlation of log information to determine the possibility of collaborative intrusion attacks. The merging of both architectures into this novel hybrid approach makes the whole new framework to be able to protect the network without having a single point of failure, like a centralized approach. Moreover, we drastically save processing power and would not have the same detection speed and capacity, like a hierarchical approach. Detailed information can be found in Table I.

TABLE I
SUMMARY OF CIDS RESEARCH - SYSTEM ARCHITECTURE POINT OF VIEW

Classification	Advantages	Disadvantages
Centralized	Efficient for small scale cooperation	Single point of failure; Poor scalability
Hierarchical	No single point of failure	Limited Scalability; Reduced detection capacity during attacks
Distributed	No single point of failure; Better scalability	Load imbalance during attacks; Uncertain detection accuracy; Simplistic alert correlation

In this work, we are interested in describing a new approach to handle collaborative anomaly-based intrusion attacks. The proposed framework has different IDSs being installed all over the distributed system protected. These IDSs will send log information to the Collaborative Intrusion Detection System (CIDS), which is the central repository responsible for the correlation of all the traffic incoming and outgoing from the whole network. By implementing the IDSs in this architecture the framework can:

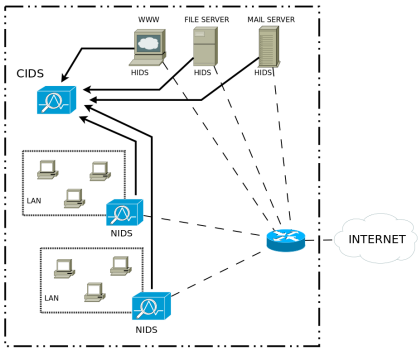
- Analyze data in real-time about the whole network and search for anomaly-based intrusion attacks. The results are obtained in a fast way using mongoDB [3] for compilation.
- Correlate all the logs from the network to determine the existence of collaborative anomaly-based intrusion attack. If an attack is detected in the network, the framework will automatically block the source IP address and inform the rest of the network entities to block that IP address. This kind of information sent to other network entities and the automatic process of determining an attack in the network by the framework, is what gives this novel framework a sense of intelligence.
- Assure a high detection rate based on the log information of the distributed IDSs and low false alarm rate (correlation of log information and classification of legitimate and anomalous traffic).

III. HYBRID INTRUSION DETECTION SYSTEM

A. IB Framework

Figure 1 describes the proposed framework for *correlative anomaly-based intrusion detection* in a network. The dash lines indicate the connections inside the network for operation. The solid bold lines indicate the connections inside the network for collaborative intrusion detection.

Fig. 1. Proposed Framework



The framework is composed by the following components:

- **NIDS:** The NIDS in the network is responsible for stopping intrusion attacks in the LAN area of the network they were installed for protecting. Installing NIDS to secure a network is easier and the cost of installation and maintenance are decreased. These NIDS send log information to the *Collaborative Intrusion Detection System (CIDS)* installed on the network.
- **HIDS:** The HIDS in the network is responsible for stopping intrusion attacks for each server or host in the network. By having each server or host with its own HIDS will assure that each of the services used in the network will be protected. Another benefit will be to prevent a bottleneck for the services attended by the servers.
- **CIDS:** The CIDS is the IDS specialized to correlate log information between the above mentioned components.
- **IB Framework:** The IB framework is our novel approach to be able to detect anomaly-based intrusion packets. This framework correlates all the information sent by the different NIDSs or HIDSs mentioned above. The correlation will be managed and centralized in the CIDS. The framework is able to analyze packets in real time and will have the capacity to classify them as either *legitimate* or an *intrusion*. The IB framework will classify all the packets based on a *trust coefficient (TC)*. This TC at first marks every single incoming packet in an ”unknown” state. After more incoming traffic of the same kind will be present on the network, and its behavior has been analyzed, the TC will converge to a *legitimate* or *intrusion* state.

As a summary, the input to the IDSs will be the incoming traffic passing through each one. Each IDS will be responsible for protecting the part of the network it guards. The log information capture by each of the IDSs will then be sent to the CIDS to correlate it. The CIDS will manage the information in a database for easier access and doing queries. To be able to acquire better results on analyzing the data, the solution is implemented using mongoDB. The CIDS processes lots of information because all the log information from each NIDS or HIDS have to be correlated and then determine the possibility of a coordinated / collaborative attack. The information managed by all the IDS will be in the same format to facilitate its processing.

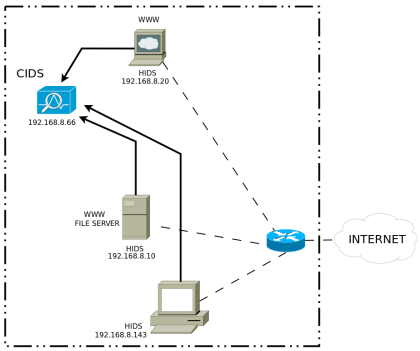
f

B. The Implemented Framework

1) Description: The implementation of IB framework is simplified for this paper. The scenario was built to show the methodology proposed to rapidly decide if an ssh login attempt can be consider *legitimate* or an *intrusion*. The scenario shows two important topics to be considered. a.) The importance of log correlation and how this helps to determine anomaly-distributed attacks and b.) How the proposed framework determines that a login attempt is either *legitimate* or an *intrusion*. The classification method use in the IB framework is based to ensure that the anomaly detection rate is increased and the false positive rate is decreased.

In the scenario, there are three servers: a.) A web server (192.168.8.20) with 10 virtual hosts, b.) A server (192.168.8.10) that has web and file services running, c.) A server (192.168.8.66) that has a web service running and is used as the correlation server. This server is the most powerful in resources. For this scenario it uses 4 vCPUs and 8GB of RAM. and a d.) Host (192.168.8.143) running an IDS and some basic services.

Fig. 2. **Implemented Scenario**



Our framework has been implemented with regards to anomaly intrusion detection for brute force attacks on a network with SSH capabilities on its hosts. Based on the logs of the network entities such as servers, appliances or end user clients, the data will be stored and analyzed. For this first approach, the key parameters are the source IP address to whom the login attempt was directed and the trust coefficient (TC) calculated after each new login attempt. The centralized and correlated logs will be the information needed to extract and filter login attempts by its source IP address. The trust coefficient (TC) is the value that determines if the login attempt is considered (i) an unknown package (first time information is in the framework), (ii) regular access (legitimate) or (iii) an intrusion. The trust level of a source IP address doing a login attempt is determine, manage and filter depending on the trust coefficient (TC) that was describe above and how it responds to two thresholds, a *minimum (MINTHR)* and a *maximum (MAXTHR)* present in the decision making. The characteristics of these thresholds are:

- MINTHR and MAXTHR are in the range [0, 100],
- MAXTHR>MINTHR.

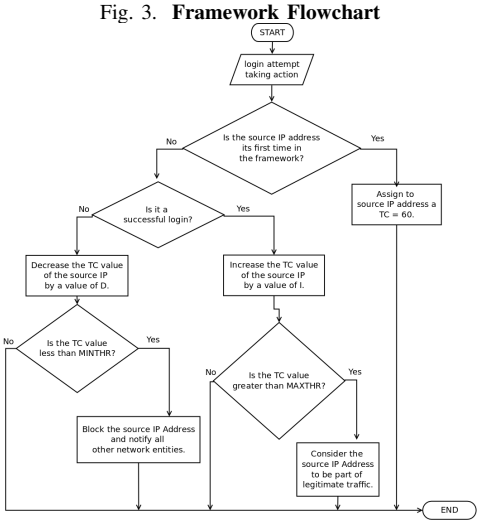
These thresholds will be of use to determine the status of a login attempt in one of the next classifications:

- If the trust coefficient is between 100 and MAXTHR it means that the login attempt is consider a *legitimate authentication*. In this case, the source IP address doing the login attempt is considered to have *high trust level*.
- If the trust coefficient is between MAXTHR and MINTHR it means that the login attempt is an unknown authentication state. This might happen when the source IP address is added for the first time to the framework analysis or still gaining / losing its trust level.
- If the trust coefficient is between MINTHR and 0 the login attempt is considered part of an attack and the source IP address has a low trust level. Automatically the framework will add the source IP address to a list of IPs that should be blocked. A notification will be sent to every other network entity to inform them about this new source IP address that is consider to be part of an attack and that has to be blocked. By implementing this notification information to all the network entities in the network and reconfiguring their IDSs, the whole framework projects a sense of intelligence by being able to reconfigure its behavior to new anomaly intrusion behaviors that will be considered attacks.

The trust coefficient will either increase or decrease during the login attempts. The increasing or decreasing values are considered I and D, respectively. The computation done for determining an increment or decrement of the trust coefficient is based on the fact that an ssh login attempt can be either considered as a *successful login* or a *failed login*. *P* is a coefficient that will be useful to determine the relationship between a successful login or a failed login. Define $D = P*I$, the greater this P value is, the more secure the sysadmin or IT manager wants its hosts to be. P value should always be greater than 0.

2) An Example: At first, a source IP address doing a login attempt will be considered *unknown*. The second and next login attempts will be considered either a successful login or a failed login. If the login attempt is a *successful login*, the trust coefficient (TC) will increase by a value of

Fig. 3. **Framework Flowchart**



I. If the login attempt is consider a *failed login*, the trust coefficient (TC) will decrease by a value of D. Notice that since D will always be greater than I, it will be easier for the trust coefficient (TC) to fall into the zone considered as an intrusion rather than into a legitimate zone.
After determining the temporal TC value of each login attempt as mentioned before, the framework will determine the status of the source IP address regarding its last login attempt. Any source IP address analyze by the framework could either be in one of the next statuses. a.) Still at an *unknown stage* if the TC value is between the MAXTHR and the MINTHR. b.) Considered to be part of *legitimate traffic* because the TC value is greater than the MAXTHR. and c.) Considered to be *an intrusion attempt* because the TC value is less than the MINTHR.

All the authentication logs are being centralized in one server running syslog. This server is what in Figure 1 is considered as the CIDS (Collaborative Intrusion Detection System). This server is analyzing all the logs and based on the results will determine the temporal classification of the host. Calculating the trust coefficient (TC) with this methodology helps the false positive rate to decrease and the anomaly intrusion detection rate to increase for this specific case scenario running *ssh*. A false positive in IDS terms, is the case when an event in a normal behavior is considered an anomaly intrusion or attack. The reasons why the IB Framework is making an improvement to these both rates are:

- The false positive rate is decreasing because the framework is exactly blocking the source IP addresses that have committed various failed logins. There are no other events that can be misinterpreted in the analyzing neither blocking. Either the login attempt is considered a successful or failed login.
- The anomaly intrusion detection rate will increase because of all the correlated logs present in the framework. Any event that seems suspicious and causes a failed login will be centralized, analyzed and reported as anomaly intrusion and is easily detected.

IV. Experimental studies and Results

A. Testbed description

The experiments done for this scenario were first tried with simulated attacks and then with real attacks. The simulation ran for a week and was useful to determine the correct trust coefficient value each host should have as a starting point and the values of I and D for this first approach. The simulation was composed of a series of login attempts done automatically by a script running from a server. A login attempt was automatically performed every 5 minutes. The values taken into consideration to run tests in the testbed environment are detailed here:

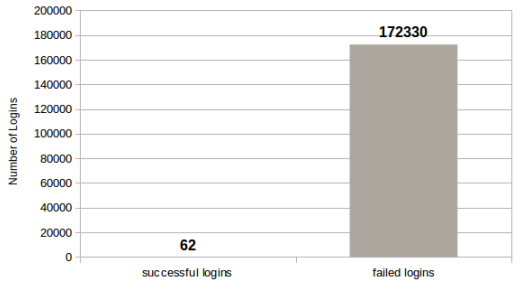
- First, D (TC Decreasing Value) has a value of 10 and I (TC Increasing Value) a value of 5. With these considerations, the value of P is then 2, meaning that a user needs to do 2 successful logins in order to set its trust coefficient (TC) back to the value it was before he did his first failed login. In other words, the cost of a failed login is twice the cost of a successful login.
- TC should have a starting value of 60 based on the usual configuration for ssh remote access that after 3 failed logins the login session will be closed. So, we considered that 2 sessions should be what any regular user may need to exactly type his right password and do a successful login.
- MINTHR will have a value of 20, meaning that after 6 failed login attempts the trust coefficient for the source IP address will be below MINTHR and the trust coefficient will be in the *anomaly intrusion attack zone*.

B. Case Study

After determining the best values for D and I and the starting value of the TC we gave full access to the Internet to the servers and gave them a sense of honeypots. By simply publishing to the Internet the existence of some of the webservers and doing some remote connections, the brute force attacks from different parts of the world began. At first, we only considered the option of simply loging and be able to grasp lots of information from the real environment. The results obtained were amazing having almost 172,000 login attempts during 45 days. The successful login attempts during these 45 days were only 62 in contrast to the more than 170,000 failed logins. In Figure 4 you can barely distinguished the successful logins done during these 45 days.

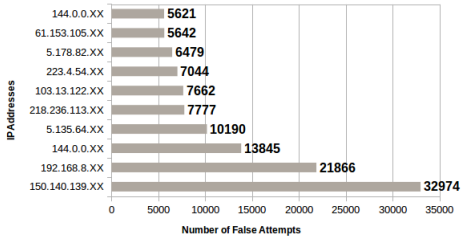
Interesting fact, from the more than 172,000 failed logins are that after launching the honeypot servers to the Internet, the number of source IP addresses attacking the server were around 428. From the top 10 source IP addresses that attacked our servers we can also highlight that, in the period where blocking was still not activated, the *number 1* source IP address made a brute force attack of almost *32,974 login attempts*, as seen on Figure 5. Notice that even *number 10* of our Top 10 Attackers List still had the chance to launch a

Fig. 4. **Login Attempts**



brute force attack of almost *5,621*. A total of *119,100* failed login attempts were inserted into the log files with only 10 IP addresses.

Fig. 5. **Top 10 IP Addresses with Number of Failed Logins**



V. Conclusions

After correlating all the information in the CIDS the results obtained for a brute force attack on an environment running ssh confirmed that:

- The anomaly-based intrusion detection rate can be increased by doing correlation of all the logs of the different network components. Correlating logs from different network entities let the framework in advanced determine possible collaborative attacks based on the failed attempts an specific source IP address is committing.
- Based on the correlation and early discover of an attack, the framework is able to rapidly determine and confirm an attack, and also to notify the rest of network entities on the network to block the IP source address of the attacker. This process of determining an attack, filtering, identifying and notifying other entities is one the main contributions of this paper.
- With the process mentioned above, the IB framework is capable of reconfiguring itself and learn from the environment it is acting. This other main contribution makes the process of detection and prevention of intrusions and attacks to be intelligent. Important that no human intervention is needed to determine the right behavior of traffic for this case study.
- The false positive rate can be decreased by knowing the exact profile of a thread of communication between the network entities. In this case study, we are following the thread of communication to establish a remote session between network entities as described in the RFC 4254 [2].
- In this case study, the log message was modified to be able to run scripts on it and parse the information useful to count the exact successful and failed login attempts each source IP address committed. This modification let us understand that mainly all the information we need to implement the framework was already in the environment, but did not have the correct analysis nor correlation, to be able to make it useful information to protect hosts, servers and the whole network.

As a future work, we will include more variables to determine the possibility of being part of collaborative or anomaly-distributed intrusion attacks. We will also work in making the framework to be able to reconfigure its normal profile using machine learning techniques and be able to define new collaborative or anomaly-distributed intrusion attacks. It is important to highlight that the new classification of traffic will be shared with the rest of the IDS to make a more precise binary classification in further analyses.

References

[1] Data set, 1999 DARPA Intrusion Detection Evaluation Data Set. http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/1999data.html, 1999.

[2] The secure shell (ssh) connection protocol. http://www.ietf.org/rfc/rfc4254.txt, 2006.

[3] mongodb. http://www.mongodb.org, 2009.

[4] M Ali Aydın, A Halim Zaim, and K Gökhan Ceylan. A hybrid intrusion detection system design for computer network security. *Computers & Electrical Engineering*, 35(3):517–526, 2009.

[5] Karan Bajaj and Amit Arora. Improving the intrusion detection using discriminative machine learning approach and improve the time complexity by data mining feature selection methods. *International Journal of Computer Applications*, 76(1):5–11, 2013.

[6] Chia-Mei Chen, Ya-Lin Chen, and Hsiao-Chung Lin. An efficient network intrusion detection. *Computer communications*, 33(4):477–484, 2010.

[7] Pedro Garcia-Teodoro, J Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1):18–28, 2009.

[8] Sapna S. Kaushik and Dr. Prof. P. R. Deshmukh. Detection of attacks in an intrusion detection system. *International Journal of Computer Science and Information Technologies*, 2(3):982–986, 2011.

[9] Mrutyunjaya Panda, Ajith Abraham, and Manas Ranjan Patra. A hybrid intelligent approach for network intrusion detection. *Procedia Engineering*, 30:1–9, 2012.

[10] M. Roesch and C. Green. Snort. http://www.snort.org, 2004.

[11] Saeed Salah, Gabriel Maciá-Fernández, and Jesús E. Díaz-Verdejo. A model-based survey of alert correlation techniques. *Computer Networks*, 57(5):1289–1317, 2013.

[12] Ciza Thomas. Application of machine learning for intrusion detection: Challenges and solutions. 2013.

[13] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36(10):11994–12000, 2009.

[14] Chih-Fong Tsai and Chia-Ying Lin. A triangle area based nearest neighbors approach to intrusion detection. *Pattern Recognition*, 43(1):222–229, 2010.

[15] Chenfeng Vincent Zhou, Christopher Leckie, and Shanika Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. *Computers & Security*, 29(1):124–140, 2010.