

State-of-the-Art in Intrusion Detection Systems

Hyung-Joon Kim, Abhishek Pamnami, Milin Patel
Dept. of Electrical and Computer Engineering
Stevens Institute of Technology, Hoboken, New Jersey

Abstract

With the astonishing evolution of computer systems and the Internet, recent years have seen the explosive growth in the number of internetworking systems. These days, the networked information systems play crucial roles for most governments, enterprises, and even individuals; thus, the systems must remain not only “up-and-running” but also “secure” – against any form of unwanted harmful actions such as attack, misuse, and abuse. However, the networked information systems have historically undergone ever-increasing events of harassment by hostile parties despite the existence of a variety of security technologies. In general, security can be achieved in three phases: prevention, detection, and correction. Prevention is the most ideal solution, but unfortunately can’t be achieved perfectly. Then, the next phase is detection, much preferable to correction, in which intrusion detection technology is used to monitor all traffic, detect malicious attempts, and respond to the attempts. Such intrusion detection scheme has gradually gaining its gravity as an essential component for a complete security package. Not surprisingly, an intrusion detection system (IDS) requires sophisticated techniques to examine massive traffic passing through the network. Besides, there are many approaches to designing a novel IDS depending on specific purposes or network environments. This paper will present the basic concepts of IDS as well as the current state-of-the-art in IDS, and introduce some case studies that apply the state-of-the-art in IDS to network infrastructure.

1. Introduction

The historical evolution of computer systems along with the Internet has drastically reduced the cost of transporting information over the world by means of internetworking. With the apparent effectiveness of internetworking, today’s networked information systems are expanding ubiquitously. Indeed, recent years have seen the explosive growth in the number of devices interconnected to computer and telecommunication networks. As a result, tremendous amounts of data are transmitted from and to, processed, and stored at central networked information systems. Without doubts, the networked information systems play crucial roles for most governments, enterprises, and even individuals. Therefore, they must remain not only “up-and-running” but also “secure” – against any form of unwanted harmful actions such as attack, misuse, and abuse. However, the information systems have undergone ever-increasing events of malicious activities despite the existence of a variety of security technologies. Attackers attempt to steal, modify, and destruct valuable information and at worst damage the victim systems. And also, attackers often attempt to make services merely unavailable to intended legitimate users by exhausting the system resources – DoS/DDoS. In many systems, security cannot allow these threats because the impact of such attacks is immeasurably gigantic and irrevocable.

Despite of the existence of a variety of security protections, attackers eventually manage to get through them and attacking techniques are speedily evolved. Thus, even though the security systems fail to defend against attacks, they must be well aware of being attacked and have a mechanism to perform countermeasures in order to prevent further attacks and reduce the

damage and loss resulted from the attacks. This is what Intrusion Detection System (IDS) for. Detecting intrusion is not easy because of the complication of modern network technologies. It's more difficult when a network experiences a heavy volume of traffic, especially with high-speed network connectivity. For this reason, one simple solution cannot solve the difficulty of detecting intrusion. Therefore, there exist various types of intrusion detection systems in terms of their "clients" such as network, host, and application. Lately, hybrids of those two or more types also exist. Furthermore, each type of state-of-the-art IDS usually employs various intrusion detection methodologies together for better detection performance.

2. Background

In general, security can be achieved in three phases: prevention, detection, and correction. Prevention is the most ideal solution, but unfortunately the history shows us it can't be achieved perfectly. Even so, it is a very bad idea to entirely rely on prevention. It's because in case an attacker somehow finds a way to make a security whole on prevention and make it a good-for-nothing, the cost for fixing the vulnerability and restoring the system back to normal condition must be incredibly expensive in the later phase if there is no preparation for that. Therefore, security protection systems are better off having ready-to-go correction mechanisms as well. By well-designed correction mechanisms, compromised or malfunctioning systems can be quickly repaired and restored to normal condition. Prevention is effective before successful intrusions. Correction is active after successful intrusions. However, once successful attacks eventually manage to get through prevention, it's a matter of time that the whole system is attacked, compromised, and malfunctioned. Thus, we need to have an interim stage such as detection phase, which is positive during intrusion. By a detection mechanism, even if prevention fails to stop intrusion, a protected system can be at least aware of being attacked so that the system can take some actions to reduce the probability of propagating damage and loss. By doing so, detection can shorten the gap between prevention and correction for minimizing the *net* cost. "The earlier one intervenes, the more cost effective the solution will be [1:77]." For this reason, the detection phase of security has gradually gaining its gravity as an essential element for a complete security solution.

In such detection phase, Intrusion Detection System (IDS) is used to monitor all or partial traffic, detect malicious activities, and respond to the activities. Malicious activities include network attacks such as DoS/DDoS, data-driven attacks, privilege escalation, unauthorized logins, illegitimate data access/modification, malware proliferation, and all other potentially harmful actions. All these malicious activities are treated as "intrusion" by IDS whether they are intentional or not. Intrusion can occur through a variety of vulnerabilities of a system, such as software bugs and improper configurations, or simply brute force attempts. Intrusion can come from either outside network or inside network.

All these characteristics of intrusion and intrusion detection help differentiate IDS from firewalls and anti-viruses. Firewalls are based on filtering mechanisms specified by a set of rules, known as a policy. The basic operations of firewalls are filtering packets passing through specific network ports or hosts. In other others, firewalls prevent unauthorized traffic from passing through a protected system. However, some traffic must pass in and out of a firewall in order for the protected system to be useful. If a system is kept completely isolated, the system may be secure but not useful at all. That said, firewalls provide an all-or-nothing type of security. In other words, a firewall cannot protect against data that are allowed to pass through the firewall.

For example, well-known port for HTTP service, port 80, is usually open in most systems. Given this policy, malicious web traffic can pass through the firewall and hit the protected system without a hitch. Besides, firewalls cannot stop insider attacks. On the contrary, anti-virus inspects executables (viruses or worms) or running process in memory of a protected system rather than network traffic to the system. Although anti-viruses monitor file integrity to protect against illegal data modification or so, they cannot stop malicious network traffic intended for network attacks, data-driven attacks, and so forth. Having all said that, typical functions of IDS include: (1) monitoring all or partial packets to detect hostile activities in real time (2) recording related event for later analysis (3) reporting network administrator to perform follow-ups.

3. State-of-the-Art in Intrusion Detection Systems

Intrusion detection has never been easy nor will be. Even current state-of-art IDS is still far from being as accurate as other security protections. For any security protection, it's impossible to achieve 100 percent security. For IDS, the fact is very true. Even though IDS has drastically improved over time, many still suffer from the difficulty of detecting suspicious activities in severe network environments and from the inaccuracy of detection mechanisms. As a result, many IDSs often generate intolerable quantity of false negative and false positive. The difficulty of designing and implementing a novel IDS results from numerous characteristics of modern computer network.

High loads and high-speed – IDS is mostly used to protect central networked information systems that operate as servers for large organizations or service providers. These central systems are typically connected with high-speed Gigabit Ethernet or optical fiber links. Besides, not only does massive traffic pass in and out of the systems, but also heavy computing processes are performed at the center of the systems. Given this circumstance, it's not easy for IDS to handle all activities.

Real-time processing – IDS's reason for being is to stop hostile activities as early as possible in order to minimize the probability of succeeding attacks and the damage and loss that could possibly result from the hostile activities. Thus, intrusion detection had better be real-time or near real-time processing. Under high loads and high-speed, however, it's very difficult to perform intrusion detection in real time.

Increasing complexity for defenders – Today's networked systems comprise a wide range of computer systems and information depositories, connected by a variety of network technology. The higher complexity the systems have, the more vulnerabilities the systems will have. Besides, applications and services are becoming more diverse and complex over time. Such a high complexity of the systems and services makes it very difficult for IDS to not only gather static information about wrong behaviors but also identify deviations from normal behaviors.

Decreasing complexity for attackers – A simple reasoning applies here for security. Increasing complexity for defenders turns into decreasing complexity for attackers. In addition, the advance in technology feeds up the attackers, too. Attacking techniques have been very quickly evolved and mutated, and the knowledge and high-quality tools are widely available to attackers. As a result, the expertise levels for successful attacks have been decreasing.

Given these circumstances, one simple solution cannot solve the difficulty of intrusion detection. For this reason, there exist various types of intrusion detection systems in terms of their “clients” such as network, host, and application. Lately, hybrids of those two or more types also exist. Each type of state-of-the-art IDS usually employs various intrusion detection methodologies together for better detection performance. In other words, an IDS is designed differently – specialized – depending on where, what, and how it detects.

Certainly, IDS with a high need for sophisticated techniques can be reinforced and bridge over the difficulties by means of smart technology such as machine learning, data mining, and artificial intelligence. In fact, such smart technology sees gigantic potential to be applied in intrusion detection. The reason is that intellectual reasoning, dynamic decision-making, and self-adaptive attitude are necessary for IDS to survive in increasing complexity of computer network. To implement intelligent capabilities, many researches pull the ideas from AI, probabilistic modeling, computational complexity theory, cognitive psychology, game theory, and empirical machine learning; for instance, among some popular machine learning methods, neural networks, clustering methods, decision trees, and Bayesian networks are popularly applied to intrusion detection [1:82]. Nevertheless, the machine learning methods that have been used are still not competent to ensure tolerable accuracy, and “its full potential has not yet been brought to bear in addressing computer security problems [1:90].” In addition to the machine learning, artificial immune systems based on the negative-selection algorithm (NSA), the negative-selection with detection rules (NSDR), and the negative-selection with fuzzy detection rules (NSFDR) are discussed in [1:Ch.7]. All these smart technology are exceedingly likely to appear more and more in future IDSs.

In addition to sophisticated detection techniques, there are other requisites for state-of-the-art IDS. Firstly, better IDSs should provide overall simplicity of operation and management in spite of the complexity of its underlying tasks. With overwhelming complication of network technology, there are operational challenges with IDS; that is, lack of adequate evaluation measure and necessary knowledge for IDS makes it difficult for administrators to evaluate, maintain, and deploy IDSs [2]. Secondly, future IDSs should provide a better way to involve human in intrusion detection. IDSs nothing but generate reports based on *raw* data analysis and diagnosis and raise alarms, without providing a way for human to interact and reason about malicious activities; unquestionably, human have excellent intuition, symbolic reasoning, and graphical thinking that could complement IDS, so there is a clear need for enhanced human-computer interaction (HCI) [2]. In addition, the intrusion-related reports and information that IDS produce are not human-friendly while most IDSs lack support for human analysis; thus, graphical representation is a must-have HCI for state-of-the-art IDS [2]. For an example of such need for graphical representation, a prototype of visualizing network traffic for IDS is introduced in [3]. Thirdly, there is the need to bring seamless interoperability to IDS [2]. To efficiently cover today’s large-scale network, it’s crucial for different types of IDSs to be interoperable and cooperate with network components and operation systems. As a traditional solution to these problems, some efforts have been put on developing standards for intrusion detection: Intrusion Detection Message Exchange Format (IDMEF) [4]. Thirdly, any modern network component should have high scalability and flexibility to accommodate the explosive growth in today’s network environments. There has been recently a popular concept of Service-oriented Architecture (SOA) in software design and architecture, and IDS is not an exception for a customer of SOA. For sure, a large-scale network needs to composite multiple IDSs. And also, if

a new IDS need to be added to or replace the existing one, there must be a novel way to do that seamlessly without alteration or rearrangement of the existing components.

For that, the dynamic composition to build more flexible and adequate IDSs for heterogeneous distributed systems is proposed in [5]; in this proposal, the extensive use of XML (Extensible Markup Language) and web services are suggested for the components search, selection, composition, and communication of IDSs. Finally, IDSs like other security services such as firewalls and anti-viruses are expected to evolve to desktop software for smaller businesses and home users. Not only can they protect themselves from intrusion, but they can also prevent the proliferation of malware, which reside on anonymous systems and are used for serious network attacks, for example DDoS (Distributed Denial of Service), targeting core networks. Even more, in near future, IDS technology will be integrated in operating systems and network software to provide seamless security services [4].

4. Types of Intrusion Detection System

There exist numerous types of IDSs, depending on where and how an IDS detect what. The most important strategy of deploying intrusion detection technology is to apply a right IDS to a right place, optionally employing a combination of two or more at one place. The reason is that different IDSs aim for different functionalities, and consequently, each type of IDSs has inherent advantage and limitation. Among many, this paper overviews the most typical types of IDSs: network-based and host-based.

Network-based intrusion detection system (NIDS) resides on the perimeter of a protected network to detect suspicious traffic coming from the outside network, namely the Internet. NIDS inspects and analyzes network traffic and devices at the network, transport, and application layer to detect malicious activities. It is a necessary security component because most of the attacks come from the Internet. Since NIDSs are particularly designed to work best in monitoring malicious traffic coming from the Internet, they can be efficiently used with firewalls and switches to protect network segments. The best practice to secure the large-scale network is to divide it into smaller networks using switches whenever feasible and then apply security technology such as firewall and IDSs to protect separate network segments.

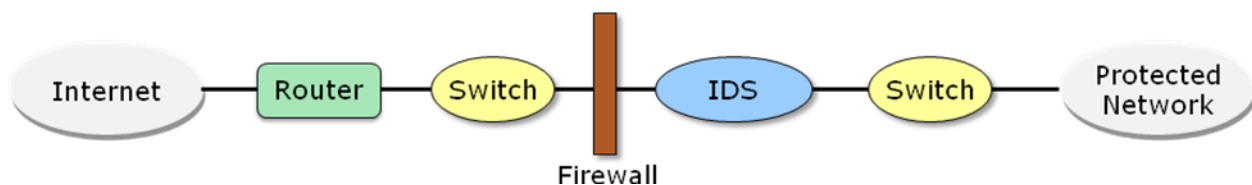


Figure 1. An example of network-based intrusion detection system

NIDS can operate on two modes – inline and passive. In inline mode, all network traffic is to physically pass through an IDS which monitors all incoming traffic to detect suspicious traffic. By contrast, in passive mode, a copy of actual traffic is made and monitored by an IDS. The copy of the traffic may be conveniently made in selective network locations according to high priority; for example, as the divisions between networks, and key network segments, and a demilitarized zone (DMZ) subnet [6]. To gain access to network traffic, an IDS in passive mode uses several methods: a spanning port of a network switch, a network trap, and IDS load balancer

[6]. NIDS can detect a vast range of malicious events and attacks occurred in the network, transport, and application layer as well as unexpected application services and policy violations, and the details of these events detected by NIDS are listed in [6].

Despite of the efficiency for network segment security, NIDS has some inherent limitations, mainly caused by the characteristic of its operational location. Firstly, it often fails to perform “full” analysis for entire packets or cause significant delay under high loads and high-speed. It’s not surprising because NIDS sits on the perimeter of a network and encounter massive traffic. To overcome such limitation, one may use some techniques to drop some packets or perform load balancing; however, the techniques should be cautiously applied in order to avoid the high rate of undetected incidents in passive mode and the possibility of disruptions in network availability in inline mode [6]. Secondly, NIDS itself is subject to DoS/DDoS attacks, much like those against IPsec gateways. The first limitation of NIDS mentioned above can paradoxically lead attackers to target on NIDS itself. In addition, so-called blinding attack can deceive about a real attack by letting NIDS generate excessive alarms using false packets and injecting the real attack into the traffic; thus, the alarm for the real attack can be overlooked. Thirdly, NIDS cannot inspect encrypted packets because they are in the middle of connection, much like eavesdroppers in the middle cannot understand encrypted traffic. Therefore, NIDS is better off accompanying with host-based intrusion detection systems (HIDS) at the end-points.

Host-based intrusion detection system (HIDS) is installed on a particular host to monitor suspicious events occurring within the host. In other words, HIDS resides on the end-points of the network. Unlike NIDS, HIDS monitors not only malicious network traffic but also various events occur within the protected host. The capabilities of HIDS include code analysis, buffer overflow detection, system call monitoring, application and library lists, filesystem monitoring, privilege misuse and abuse, system log analysis, system and application configuration analysis, and many others [6]. This can be done because HIDS is designed to operate on a specific host – “specialized” – for example, web server, mail server, database server, file server, or DNS server. HIDS is often integrated into server software and can be relatively easily implemented to communicate with other network components and operating systems. Besides, since HIDS has capability to analyze packets at the application ends, it can inspect encrypted traffic, strictly saying plaintext before encryption or after decryption. For this reason, HIDS can complement NIDS when they’re used together.

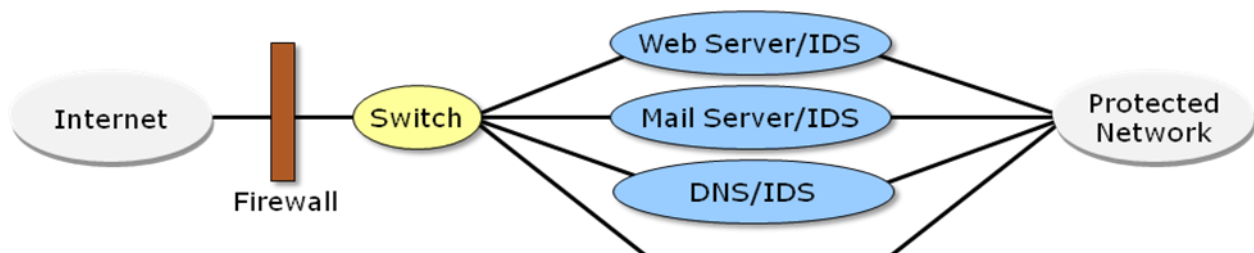


Figure 2. An example of host-based intrusion detection system

HIDS also has some inherent drawbacks caused by the characteristic of HIDS’s operational location. Firstly, since HIDSs are installed on a host, it actually consumes system resources that should be allocated for services as much as possible. Secondly, HIDS may result in conflicts with existing security policies of operating systems or firewalls. Thirdly, HIDS may require

hosts to reboot after its installation or updates. This operation is often intolerable for many critical servers.

There are also application-based and protocol-based. They sit on the front end of or within servers, respectively, monitoring dynamic behavior and state of a particular application or protocol. As mentioned earlier, one particular type of IDS cannot fulfill all the needs for intrusion detection because any type of IDS has both inherent virtue and shortcoming. Thus, a combination of two or more types of IDSs, called hybrid IDS, also exists to provide a comprehensive approach for intrusion detection. Without hybrid IDS, security-aware network should at least consider combining multiple approaches in the use of IDS for a complete solution. Besides, this hybrid approach is also applicable to the adoption of various detection methodologies and techniques for any one of IDS.

5. Methodologies for Intrusion Detection

Most of the current Network Intrusion Detection Systems (NIDSs) employ either *Misuse detection* or *Anomaly detection*. Each type has its own advantages and disadvantages. The misuse type of detection works in a way that it looks for patterns of signatures or known intrusions. It is also known as *Signature-based*. For this reason it is quite fast in the detection speed and has a lower false positives rate. On the other hand an anomaly detection system is intelligent as it is able to detect unknown intrusions. It works on profiles to do this job [6]. The profiles represent the normal behavioral activities of the users, systems, or network connections, applications. These profiles are developed by monitoring the characteristics of typical activity over a period of time. Profiles can be created based on a number of behavioral attributes like the number of emails sent by a user, the number of failed login attempts for a host, and the level of processor usage for a host in a given period of time. Great care should be taken while defining profiles because currently there is no effective way to define normal profiles that can achieve high detection rate and low false positives at the same time. If the profiles are too broadly defined, some attacks might not be detected. This leads to a low detection rate. On the other hand, if the profiles are too narrowly, some normal activities might be detected as intrusion [7].

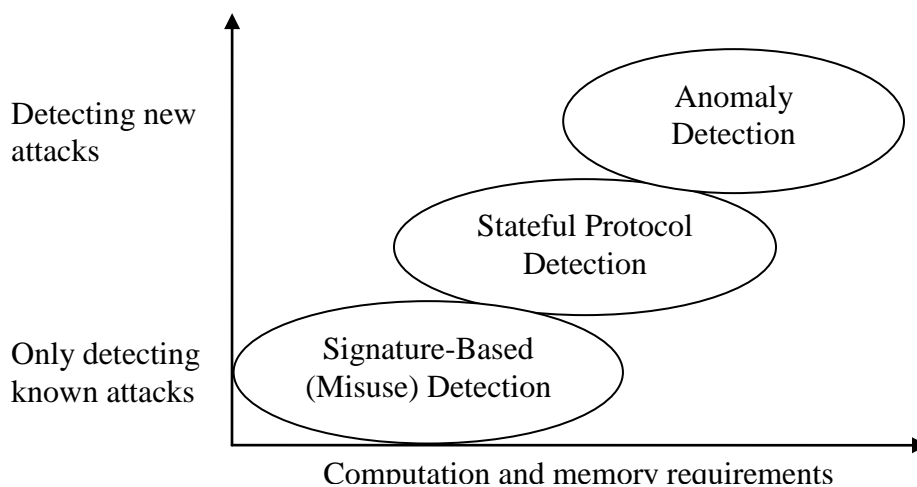


Figure 3. Three general approaches to intrusion detection, taken from [1:95]

Stateful Protocol Analysis [6] is the process of comparing predetermined profiles of generally accepted definitions of benign protocol activity for each protocol state against observed events to identify deviations. Unlike anomaly based detection, stateful protocol type relies on vendor-developed universal profiles that specify how particular profiles should and should not be used. The “stateful” in stateful protocol analysis means that the intrusion detection system is capable of understanding and tracking the state of network, transport, and application protocols that have a notion of state.

To address the problems of misuse and anomaly detection, a hybrid detection system is thought as a possible solution. The hybrid system can reach the accuracy of a misuse detection system and have the ability to deal with new attacks. A number of hybrid detection systems have been developed combining misuse component and anomaly component. The first is the one in which misuse is first used to obtain suspicious activities. Then, misuse detection is used to detect intrusions from the suspicious activities. ADAM [8] is one such system. The second approach is the parallel approach where the observed activities are fed simultaneously to the misuse and anomaly component, the suspicious activities found if any are analyzed by the correlation component to detect intrusions.

The method proposed in this paper combines the advantages of misuse and anomaly detection components. A *Random Forests* algorithm is applied in the misuse detection component to detect known intrusions. Then, an outlier detection technique provided by the same algorithm is used to detect unknown intrusions. The random forests algorithm is an ensemble classification and regression approach, which is unsurpassable in accuracy among current data mining algorithms [9].

In this section, we describe the proposed framework of the hybrid NIDS along with other approaches that combine misuse and anomaly detection – forming *hybrid detection*.

Anomaly followed by Misuse – The observed activities are fed to the anomaly detection component first and if any suspicious activities are found then they are fed to the misuse component as seen in Figure 4 from [7]. Note that here the condition is that the anomaly detection component needs to have a very high detection rate because if there are any intrusions that are missed by the anomaly component then they cannot be detected by the misuse component.

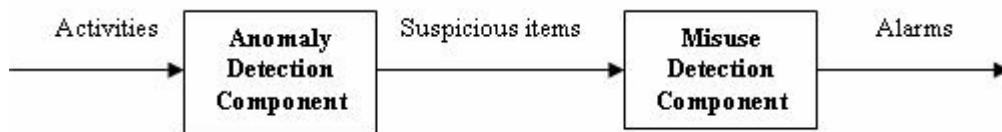


Figure 4. Framework of Anomaly detection component followed by Misuse detection.

Parallel Approach – The misuse detection component and the anomaly detection component are in parallel as the name suggests. The observed activities are simultaneously fed to both the components the suspicious activities found if any are analyzed by the correlation component to detect intrusions as seen in Figure 5 from [7].

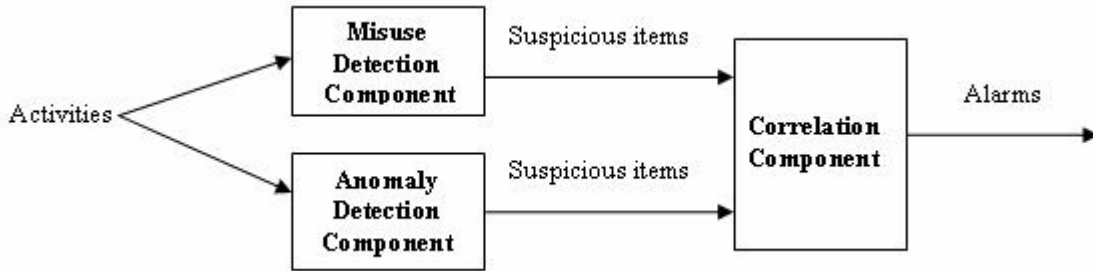


Figure 5. Framework of the Parallel Approach

Misuse followed by Anomaly – This is the proposed framework of a hybrid detection system which employs the random forests algorithm in the misuse detection component to detect known attacks. The unknown attacks are detected by the anomaly detection component using the outlier detection technique as seen in Figure 6 from [7].

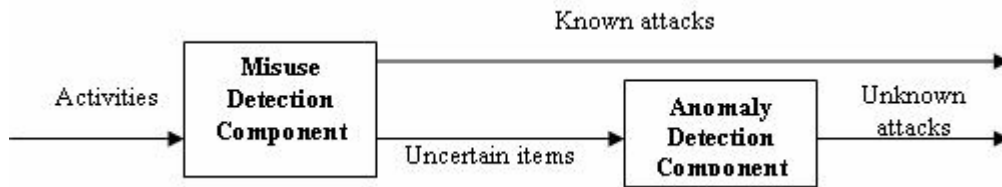


Figure 6. Framework of Misuse followed by Anomaly

The anomaly detection component should have low false positive rate, otherwise the overall false positive rate of the hybrid system will be high. High false positive rate makes the detection system useless. Another advantage of our proposed framework is that the hybrid system can detect known intrusions in real time, since the misuse detection by the random forests algorithm has high speed performance. However, low speed performance of the anomaly detection makes real time detection impossible with other approaches to combine both the detection components.

To discuss what exactly happens inside both the components is also important. The internal architecture of the misuse detection component and anomaly detection is shown in the Figure 7 from [7]. The system basically has two phases, an on-line phase and an off-line phase. The system can build patterns of intrusions for the misuse detection component and can detect unknown intrusions using the anomaly detection component in the off-line phase. It detects known intrusions using the misuse detection component in the on-line phase. In the off-line phase, the Intrusion Pattern Builder module is trained by labeled data and outputs patterns of intrusions to the Misuse Detector module. In the on-line phase, network traffic is captured and fed to Misuse Detector. Misuse Detector raises an alarm to the Misuse Alarmer module if any connection matches an intrusion pattern. Then, Alarmer will deliver alarms to security analysts. If the connection does not match any In the off-line phase, the system can detect novel intrusions using the anomaly detection component. First, the Service Pattern Builder module retrieves data from the anomaly database to build patterns of network services, and outputs the built patterns to the Outlier Detector module. With the patterns, Outlier Detector retrieves the data from the anomaly database and uses outlier detection technique to detect attacks. If it detects any attack, it raises alarms to the Anomaly Alarmer module. Anomaly Alarmer can deliver the alarms to

security analysts. It also can store the new detected intrusions in the training database. Thus, new patterns of these intrusions can be built for misuse detection.

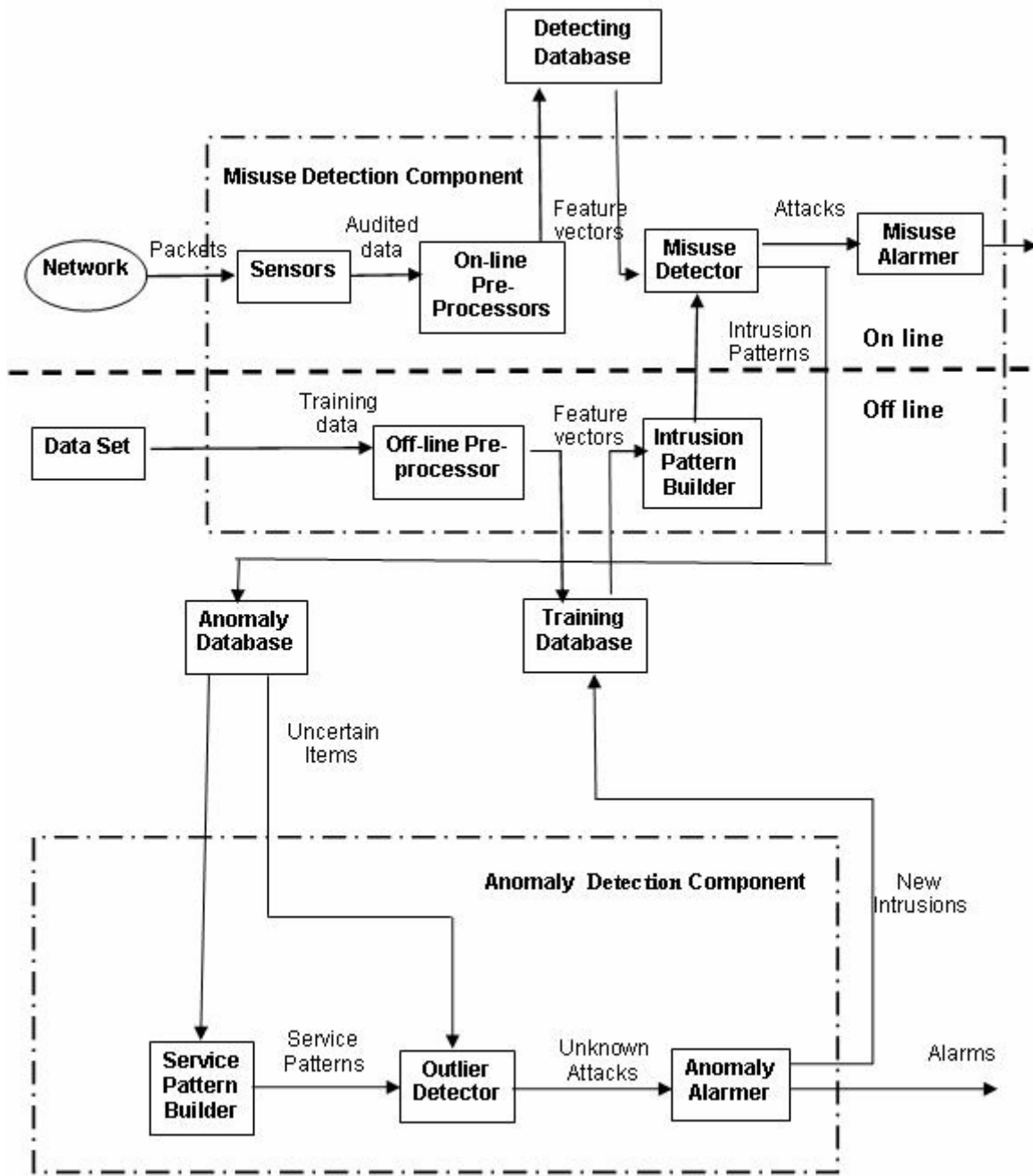


Figure 7. Architecture of the Hybrid System

6. Architecture and Processing of Intrusion Detection System

6.1. IDS Architectural Schema

As we have seen different types of IDS and methodologies used to detect intrusions, let us see a real life example of an IDS deployed in a network that protects a target system. As shown in Figure 8 taken from [10], we have a *target system* that must have assets worth protecting.

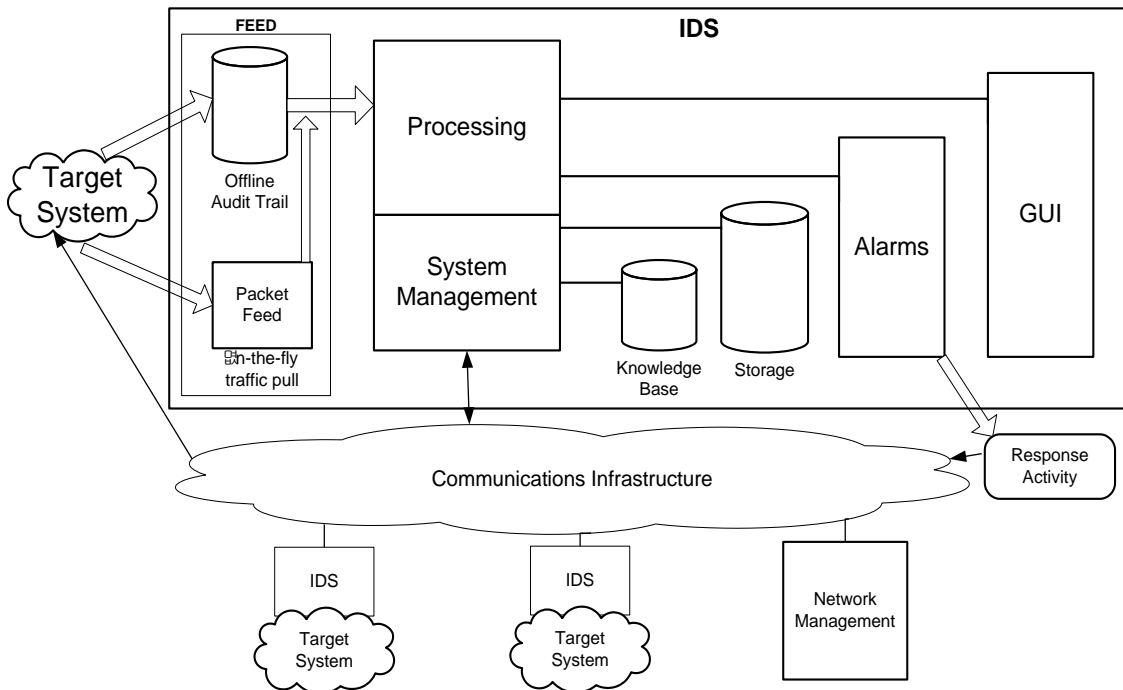


Figure 8. IDS Architectural Schema

Each component in IDS architectural schema is summarized as below [10]:

Feed from target system is a means for deriving information which could be either “*offline audit trail*” or “*on-the-fly traffic pull*”. The IDS portion starts from here and feeds are a part of the IDS.

Processing is the execution of algorithms designed to detect malicious activity to some target system.

Knowledge bases are used to store info about attacks as signatures in case of misuse detection and system profiles in case of anomaly detection.

Types of information that must be **stored** in IDS will vary from short term cached information about an on-going session to longer term event-related session information that can be sent to audit trail.

Response from IDS would be sending an alarm to the interested party, processing component or other IDS. The input feed and the output response activity are the primary interfaces between the IDS and target system.

GUI is for displaying information to the administrator.

Different components and different IDSs communicate through the communication infrastructure. It would involve protections such as encryption and access control to protect information such as alarms in transit etc. Some environments may involve multiple IDS and efforts such as CIDF (Common Intrusion detection framework) are being organized.

Network Management systems may be embedded into the intrusion processing or could be used to complement intrusion detection via remote monitoring and administration activities.

Response can be initiated based on detected intrusion information.

6.2. Example of Baseline Algorithm for Processing

There is no single approach for intrusion detection system processing. A collection of processing algorithms can be identified from recent research and development. These algorithms collectively comprise the present state-of-the-practice and state-of-the-art in intrusion detection system processing internals. A given intrusion detection may implement several algorithmic approaches to intrusion processing. We would describe one such approach to intrusion detection processing at a high level as shown below [10]. The notation used is standard pseudo-code and we will ignore the low level details and illustrate the high-level idea.

```
repeat          /*iterative loop forever*/
    target_system_feed (info)
    intrusion_processing (info, result)
    if (result = intrusion) then
        initiate_response (result)
forever
```

The repeat-forever loop is used to show that these systems function continuously and constantly monitor traffic. The function `target_system_feed (info)` is to represent a means by which information is obtained from some target system. The function `intrusion_processing (info, result)` is to represent the manner in which processing accepts an information stream, processes it, and returns a result. The comparison of the result to some intrusion activity is done in the next step and is an important component of intrusion detection algorithms. The function `initiate_response (result)` represents whatever activity is required for the result. The response would be dependent on the result and the system would respond differently for different results.

6.3. Case Study: Finding Intrusions in a Hypothetical Trail

As we have seen the architecture and processing algorithms, now we would take an example case study in which an IDS is finding intrusions in a hypothetical audit trail [10]. We assume that the audit records in the trail log the initiation of all TCP sessions. Let us consider a simple design as shown in below figure wherein audit records are coming from a gateway on the perimeter network of Intranet A which is connected to the internet.

IP addresses from A are the *in* addresses and those from the Internet are *out* addresses. Gateway IP address is *gw*. We also assume that *inbound* directed packets come into A, whereas *outbound* packets leave A for the internet.

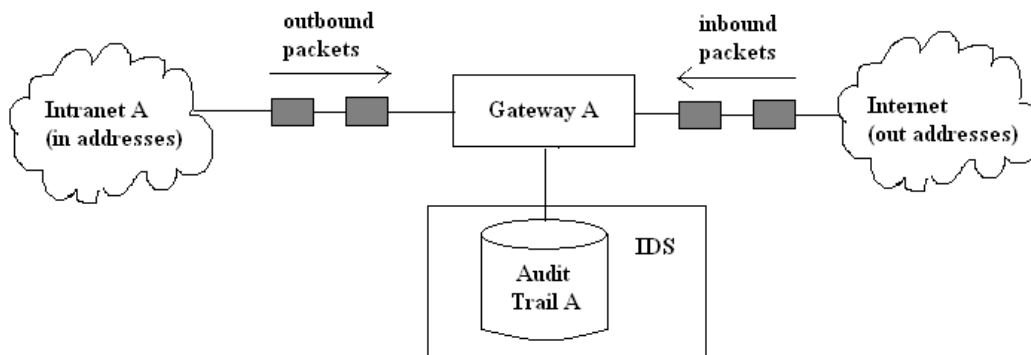


Figure 9. IDS Architectural Schema

Let us suppose that the audit trail format is as below:

<src IP add, dest IP add, src port, dest port, protocol, time, direction, success/failure of session>

Let us examine a collection of sample audit records and see if we can derive any intrusion related information.

```

<in      , in  , 4050, 80, TCP, 07:36:04, inbound, success>
<out (X), gw, 6025, 23, TCP, 07:51:12, inbound, failure>
<out (X), gw, 6025, 23, TCP, 07:51:55, inbound, failure>
<out (X), gw, 6025, 23, TCP, 07:52:17, inbound, failure>
<out (X), gw, 6025, 23, TCP, 07:52:58, inbound, failure>
<out (X), in  , 3000, 23, TCP, 13:04:22, inbound, success>
<out (Y), gw, 5000, 23, TCP, 23:54:22, inbound, success>

```

The IDS is monitoring the above audit records and by looking at these records, we can presume that something suspicious is occurring. First record shows an *in* source and *in* destination IP address for an *inbound* session. This is not right as inbound sessions should have out source IP addresses and someone is changing the source IP address here. This intrusion is referred to as IP gateway spoof. The second, third, fourth and fifth records show repeated attempts by *out* address X to telnet A's intranet gateway. This repetitive attempt warrants some attention by the administrator and the IDS. The sixth record shows that user at address X managed to telnet an inside address of A. This might raise some suspicion because the source IP address correlates to previous unsuccessful activity that was considered suspicious. The last record shows that some out IP address Y tried to telnet inbound at midnight and was successful to telnet the gateway. This resembles that the gateway has been compromised. From all the above discussion, we can say that the attacker had been trying different ways to penetrate the network and after 17 hours since the first try, he was successful. The state-of-the-art IDS have to be smart enough to detect anomalies in these kinds of sessions and alarm the administrator or provide appropriate

responses. Some action has to be taken by the IDS in the initial 6 sessions so that the last session wherein gateway is compromised never occurs.

7. Conclusion

In our paper, we have given the basics of an Intrusion Detection System, discussed the state-of-art in intrusion detection systems, its different types, methodologies used to detect intrusions and a real life example of how an IDS would function in a network infrastructure. From all the discussion, we can say that attack scenarios are never stagnant and attackers are constantly looking for new vulnerabilities for which the state-of-the-art IDS have to be sophisticated and smart enough to detect new attacks and take appropriate response actions. In an environment where security incidents are rising sharply and is costing millions of dollars, it makes sense to build state-of-the-art Intrusion Detection systems to reduce the effects of these attacks. Smart technologies such as machine learning, data mining and artificial intelligence have to be an integral part of these machines. IDSs need to maximize user defined security goals while minimizing costs and it has to be directed to respond to the most damaging intrusions when adversaries are launching a large amount of automated attacks. Hence, given the requirements and complexities of the current network environments, state-of-the-art IDSs have to be expandable, scalable and interoperable between different standards and protocols.

References

- [1] V. Rao Vemuri, *Enhancing Computer Security with Smart Technology*, Auerbach Publications, 2006.
- [2] Julia Allen, Alan Christie, William Fithen, John McHugh, Jed Pickel, and Ed Stoner, "State of the Practice of Intrusion Detection Technologies," Carnegie Mellon Software Engineering Institute, Technical Report CMU/SEI-99-TR-028, Jan. 2000. [Online] Available: <http://www.sei.cmu.edu/pub/documents/99.reports/pdf/99tr028.pdf>. [Accessed: Apr. 15, 2007]
- [3] John R. Goodall, "Visualizing Network Traffic for Intrusion Detection," *Proceedings of the 6th ACM conference on Designing Interactive systems DIS '06*, ACM Press, Jun. 2006, pp. 363-364.
- [4] Richard A. Kemmerer and Giovanni Vigna, "Intrusion Detection: A Brief History and Overview," *Computer*, IEEE Computer Society Press, Apr. 2002, pp. 27-30.
- [5] José Eduardo M. S. Brandão, Joni da Silva Fraga, and Paulo Manoel Mafra, "A New Approach for IDS Composition," *Proceedings of 2006 IEEE International Conference on Communications*, Vol. 5, IEEE Press, Jun. 2006, pp. 2195-2200
- [6] Karen Scarfone and Peter Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," National Institute of Standards and Technology, Special Publication 800-94, Feb. 2007. [Online] Available: <http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf>. [Accessed: Apr. 15, 2007]
- [7] Jiong Zhang and Mohammad Zulkernine, "A Hybrid Network Intrusion Detection Technique Using Random Forests," *Proceedings of the First International Conference on Availability, Reliability and Security ARES '06*, IEEE Computer Society Press, Apr. 2006.
- [8] Daniel Barbarra, Julia Couto, Sushil Jajodia, Leonard Popyack, and Ningning Wu, "ADAM: Detecting Intrusions by Data Mining", *Proceedings of the 2001 IEEE, Workshop on Information Assurance and Security TIA3 1100 United States Military Academy*, West Point, NY, June 2001.
- [9] Leo Breiman, "Random Forests," *Machine Learning*, Vol. 45, Issue 1, Kluwer Academic Publishers, Oct. 2001, pp. 5-32.
- [10] Edward Amoroso, *Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response*, First Edition, Intrusion.Net Books, New Jersey, 1999.