Security Patterns for Intrusion Detection Systems

Ajoy Kumar
Florida Atlantic University, Boca Raton, Florida, USA, ajkuff@yahoo.com

Eduardo B. Fernandez
Florida Atlantic University, Boca Raton, Florida, USA, ed@cse.fau.edu

Abstract

In our world of ever-increasing Internet connectivity, there is an on-going threat of intrusion or denial of service attacks. These intrusions may bring all kinds of misuses. *Intrusion Detection Systems* (IDS) play a very important role in the security of today's networks by detecting when an attack is happening. IDS have evolved into an integral part of network security which monitors the network traffic for attacks based either on existing attack patterns or signatures (Signature-based IDS) or on anomalies or abnormal behavior (Behavior-Based) in the system. We present here a pattern for abstract IDS that defines their general features and patterns for Signature-Based IDS and Behavior-Based IDS.

Keywords: Security Patterns, Intrusion Detection Systems, Signature-based IDS, Behavior-based IDS.

1. Introduction

A system intrusion is any attempt to attack a system and compromise its security aspects such as integrity, confidentiality, or availability. Intrusion Detection Systems (IDS) are implemented to detect an intrusion when it occurs and on detection they should trigger appropriate recovery measures [Bie01]. IDS monitor all traffic as it passes through a network, analyze it, reconstruct sessions and detect predefined patterns of attack or abnormal behaviors that could be caused by system attacks.

The Abstract IDS pattern defines the basic features of any IDS. An abstract pattern defines only fundamental, implementation-independent functions and threats [Fer08]. Concrete patterns add functionalities and threats and take into account the characteristics of their specific concrete environment. In this case, the abstract functions are realized by concrete IDS which operate based on known attack signatures or based on abnormal behavior or anomaly in the network, i.e, Signature-Based IDS or Behavior-Based IDS. We present here patterns for all these three types of IDS.

Section 2 describes an Abstract IDS pattern, defining the common features and threats of all IDS. Sections 3 and 4 describe concrete IDS for the Signature-Based IDS and Behavior-Based IDS respectively. For the latter two, we show only their differences with respect to the abstract pattern, the assumption being that all their other aspects are inherited from the abstract pattern. Our patterns are intended for system designers, who can use these differences to select the type of pattern they need.

2. Abstract IDS

Intent

Monitor all traffic as it passes through a network and analyze it to detect possible attacks and trigger an appropriate response.

Example

Our company has a firewall to control traffic from the Internet. However we are still plagued by viruses and other attacks that penetrate the firewall. These attacks could be already existing attacks or they could be new attacks. We need to improve our defense against such attacks.

Context

Nodes for local system that need to communicate with each other using the Internet or another insecure network.

Problem

An attacker may try to infiltrate our system through the Internet and misuse our information by reading or modifying it. We need to know when an attack is happening and take appropriate response.

The solution to this problem is affected by the following **forces:**

- *Communication:* The system is usually more secure if we have a closed network. However in today's world it is better and realistic to use the Internet or other insecure network to reduce costs, which may subject our network to security threats.
- *Real time behavior:* Attacks should be detected before the attack completes the purpose of attack so that we can preserve our assets and save time and money. It is difficult to detect an attack when it is happening. But such detection is imperative so we can react timely and appropriately.
- *Incomplete security:* Security measures such as encryption, authentication, etc, may not protect all our systems because they do not cover all possible attacks.
- *Non-Suspicious users:* Protecting our system through a firewall is quick and easy. However request coming from a non-suspicious address (permitted by a firewall) could still be harmful and should be further monitored.
- *Flexibility:* Hard coding the type of attack can be done easily. But it will be hard and time consuming to adapt to attack patterns that keep changing constantly.

Solution

Each request to access the network is analyzed to check whether it conforms to the definition of an attack. If we detect an attack an alert is raised and some countermeasures maybe taken.

Figure 1 shows the typical placement of IDS in a network complementing a firewall. The firewall filters requests for services and the IDS further checks for suspicious patterns in request sequences. If a suspicious pattern is detected, the network operator is alerted and the firewall may block some or all traffic.

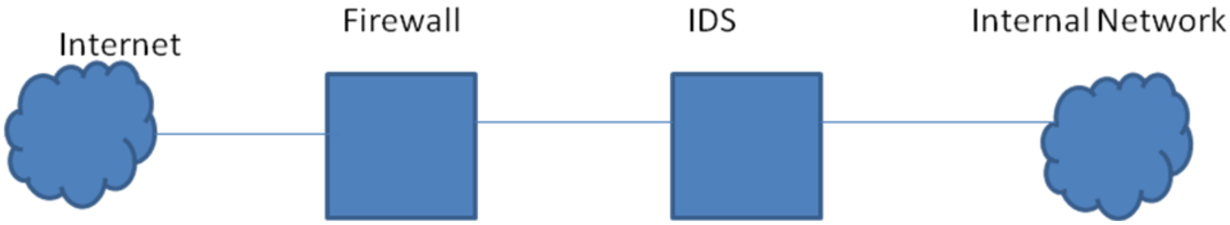

Figure 1. Possible Placement of Network IDS to complement a Firewall

Structure

In Figure 2, a **Client** requests some service from the **Server**. The **IDS** intercepts this request and sends it to an **Event Processor**. The **Event Processor** processes the event so that the **Attack Detector** can analyze the event and implement some method of detection using some information from **Attack Information**. When an attack is detected a **Response** is created.

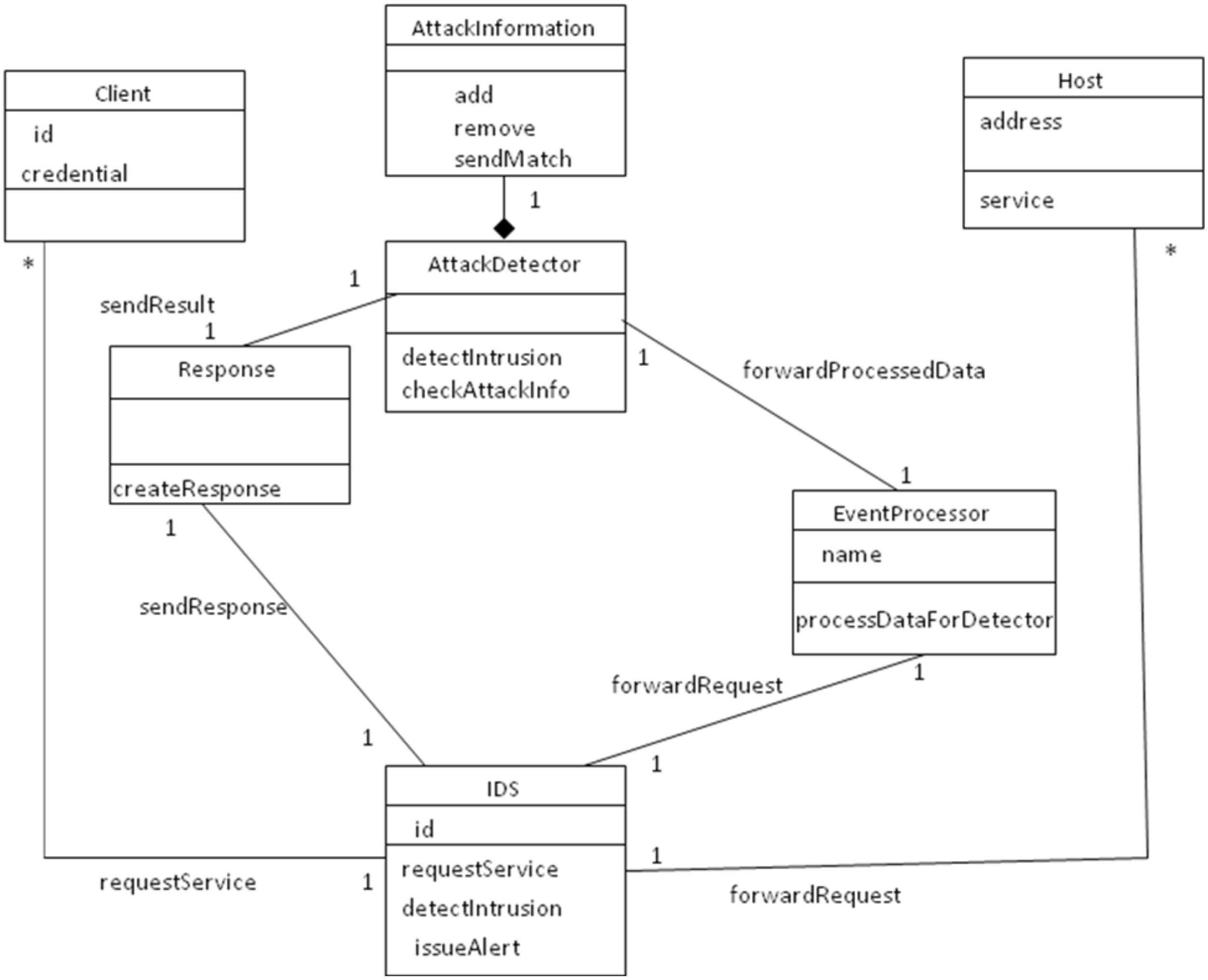

Figure 2. Class diagram for the Abstract IDS security pattern

Dynamics

We describe the dynamic aspects of the Abstract IDS Pattern using a sequence diagram for the following use case:

Detect an intrusion (Figure 3):

Summary: The client requests a service from the host. The IDS intercepts the message and checks whether the request is an attack or not and raises a response.

Actors: Client and Server

Precondition: We have attack information available.

Description:

a) A Client makes a service request to the host.
b) The IDS send the request event to an Event Processor.
c) The Event Processor processes the event data so that the Attack Detector can interpret the event.
d) The Attack Detector tries to detect whether this request is an attack or not by comparing with the available information in the Attack Information.
e) If an attack is detected a Response is created.

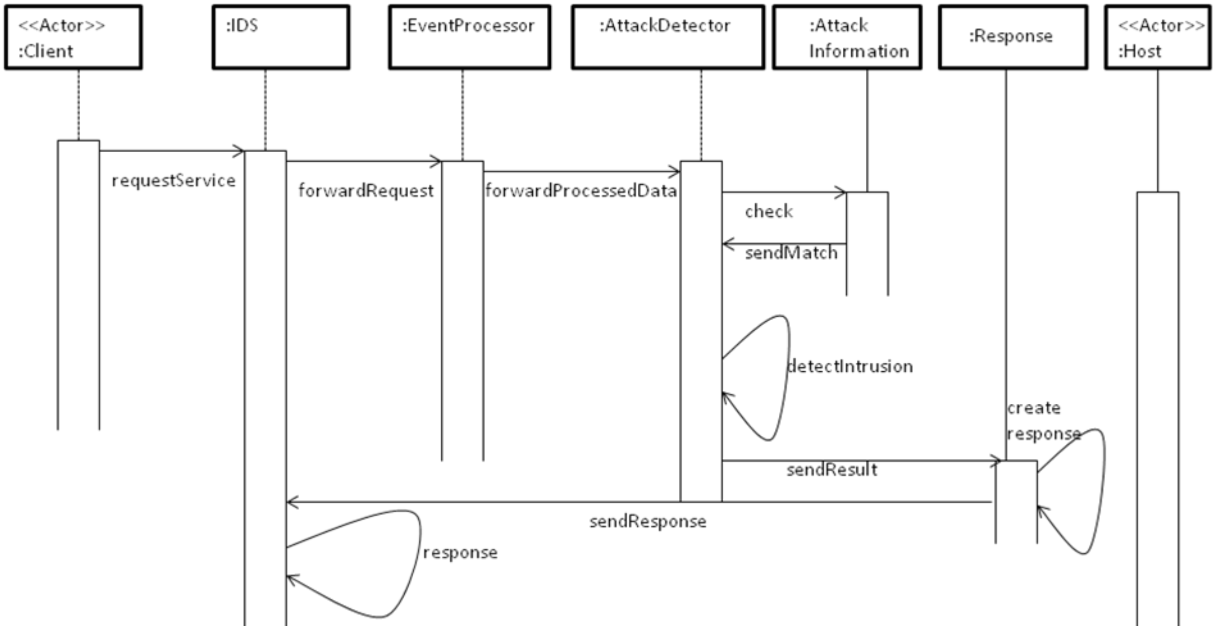

Figure 3: Sequence diagram for detecting an intrusion with Abstract IDS

Alternate Flows:

1) The Attack Information may not be able to detect an attack (a false negative).
2) The Attack Information may indicate an attack when no attack is present (a false positive).
3) If no attack is detected, the request for service is forwarded to the host.

Postcondition: If an attack is detected, suitable preventive measures may be applied.

Implementation

We need to create a database with attack information so we can check against this database and decide if an attack is happening. The incoming event is compared against the database and a decision is made whether the incoming event is an attack or not. The concrete versions of this pattern use different types of information to detect attacks.

The CIDF ("Common Intrusion Detection Framework"), a working group created by DARPA in 1998 mainly oriented towards creating a common framework in the IDS field. CIDFdefined a general IDS architecture based on the consideration of four types of functional modules as shown in Figure 4 [Gar09].

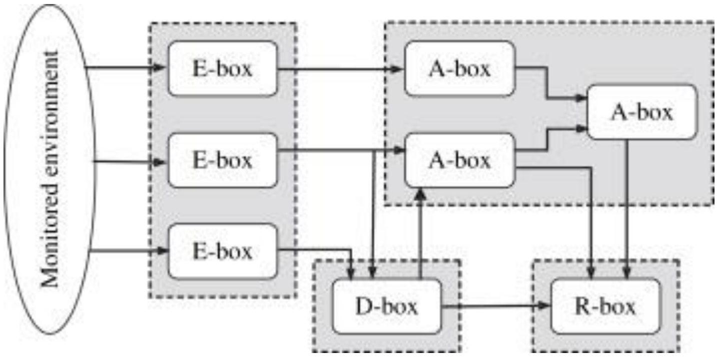

Figure 4. General CIDF architecture for IDS systems. [Gar09]

- *E blocks* ("Event-boxes"): This block is composed of sensor elements that monitor the target system, thus acquiring information events to be analyzed by other blocks.
- *D blocks* ("Database-boxes"): These are elements intended to store information from E blocks for subsequent processing by A and R boxes.
- *A blocks* ("Analysis-boxes"): Processing modules for analyzing events and detecting potential hostile behavior.
- *R blocks* ("Response-boxes"): The main function of this type of block is the execution, if any intrusion occurs, of a response to thwart the detected menace.

Concrete patterns

- IDS can be either behavior (rule) based or can be based on anomalies (abnormal behavior). There are significant differences in their use and effectiveness. The patterns for both the *Signature-Based* and *Behavior-Based* IDS are described below in the subsequent sections.
- A *hybrid model* of both the signature based and behavior based IDS together is now available. A Behavior-Based IDS detect the anomalies in traffic and then compare the anomalies with an attack signature in a Signature-based IDS.
- According to the resources they monitor, IDS systems are divided into two categories: *Host based IDS* systems and *Network Based IDS* systems. Host based IDS systems are installed locally on host machines. Host based IDS systems evaluate the activities and access to key servers within the host. The network based IDS systems inspect the packets passing through the network [Ozg05]. This classification is not discussed and is out of scope for this paper.

Known Uses

NID is a freely-available hybrid intrusion detection package that can be installed on a machine. NID monitors network traffic and scans for the presence of known attack signatures, as well as deviations from normal network behavior [Gra00].

Consequences

This pattern has the following advantages:

- *Communication:* If we can detect most attacks, we can safely use the Internet or other insecure networks to access other systems.
- *Real time behavior:* Attacks can be detected when the attack happens and the system can be alerted which saves the system both time and money from recovery measures and may prevent misuse of our assets. These attacks can be detected in real time if they have sufficient and appropriate information.
- *Incomplete Security:* This will be an added layer of security in addition to encryption, authentication, etc..
- *Non-Suspicious users:* A request coming from a non-suspicious address (permitted by a firewall) is further inspected and analyzed.
- *Flexibility:* The detection information can be modified to consider new attacks or new behavior.

This pattern has the following disadvantages:

- Some attacks may be so fast that it may be hard to recognize them in real time.
- Attack patterns are closely tied to a given environment (operating system, hardware architecture etc…) and cannot be applied easily to other systems. This means we need to define detection information tailored for an environment.
- There is some overhead in the addition of IDS to a system.

Related Patterns

- Firewalls can be added to complement the IDS [Sch06]. Firewalls usually deny requests made by unknown addresses. They can protect against attacks coming from distrusted sources and can block the addresses from where an attack originates.

- The response class could be implemented as a Strategy pattern [Gam94].

3. Signature-Based IDS

Intent

Check every request for access to the network against a set of existing attack signatures in order to detect possible attacks and trigger an appropriate response. **AKA:** Rule Based IDS, Knowledge-Based IDS.

Example

Our company has a firewall to control traffic from the Internet. However we are still plagued by viruses and other attacks that penetrate the firewall. We need to improve our defense against such attacks.

Context

Distributed systems executing applications that may provide services to remote nodes. Access to the network can be from the Internet or from other external networks.

Problem

Whenever data is accessed from the distrusted networks, there is always a possibility that this access can be harmful to the local node. We need to detect possible attacks while they are occurring. Security techniques such as authentication and firewalls are usually implemented to provide security, but we need additional defenses to detect whether an access request is a possible attack or not. The solution to this problem is affected by the following forces:

- *Known Attacks:* It is easier to protect the system against known attacks. Many attacks are new instances of known attacks and have a well defined attack signature.
- *Completeness:* If we have a complete collection of known attacks and their signatures, it is easier to detect an attack exhibiting one of these signatures.
- *Flexibility:* Hard coding the type of attack can be done easily. But it will be hard and time consuming to adapt to attack patterns that keep changing constantly.

Solution

Detect the occurrence of attacks by matching the current attack signature against the signature of previously known attacks.

Structure

Figure 5 represents the class diagram of this pattern. The IDS intercept an access request for a service. An **Event Processor** processes the information and feeds this processed information to a **Attack Detector** that tries to match the sequence of requests to the signatures in the **Attack Signature Information** and decides whether the request is an intrusion or not. If an attack is detected by getting a match of signatures, some appropriate **Response** is raised.

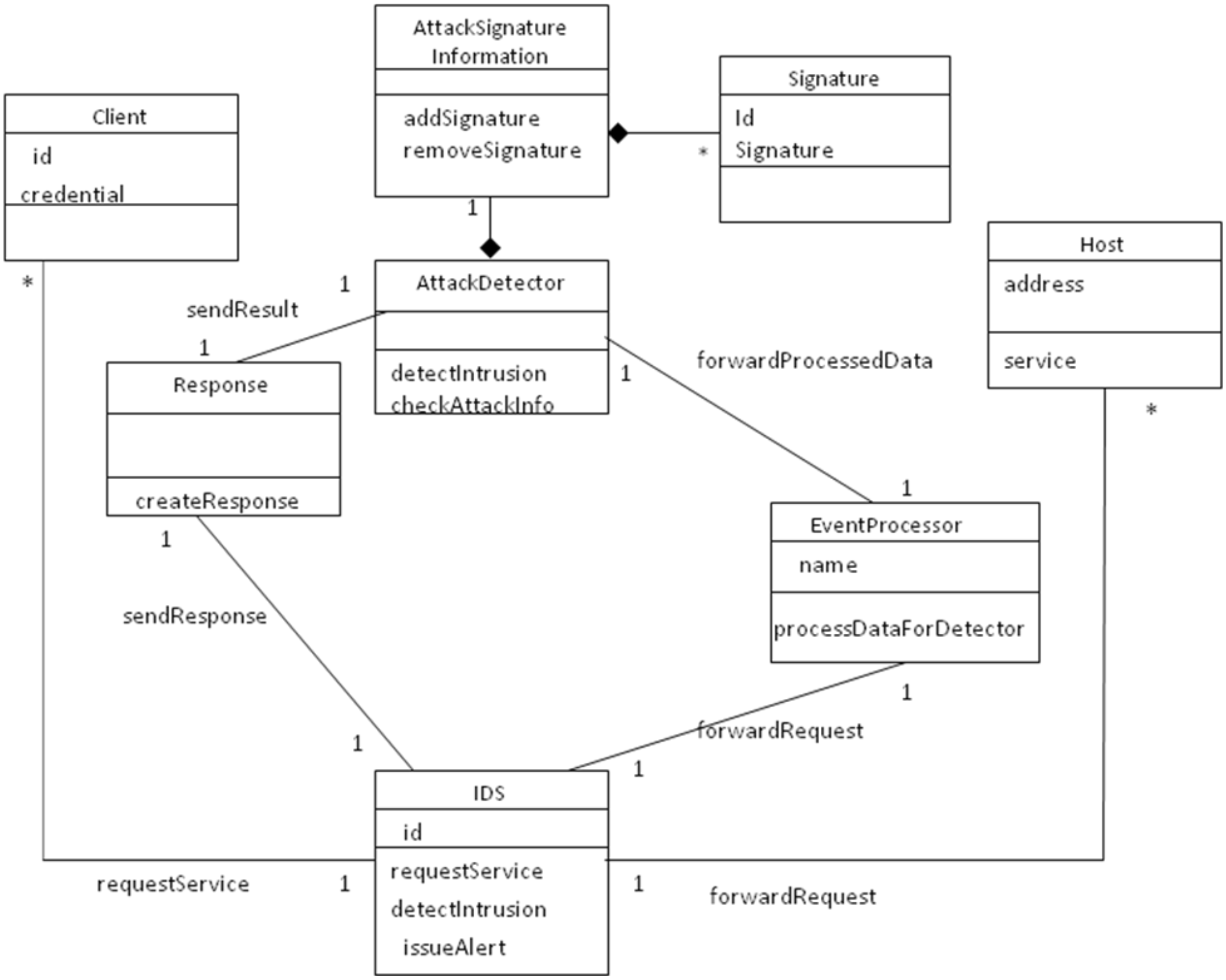

Figure 5. Class Diagram for Pattern for Signature-based IDS

Dynamics

We describe the dynamic aspects of the Signature-Based IDS Pattern using a sequence diagram for the following use case:

Detect an intrusion (Figure 6):

Summary: The client requests a service from the host. The Signature-Based IDS intercepts the message and determines whether the signature of the event matches an existing attack signature and if the request is an attack, appropriate response is raised.

Actors: Client and Server.

Precondition: We have information about attack signatures available.

Description:

a) A Client makes a service request for a service to the host.
b) The IDS send the request event to an Event Processor.
c) The Event Processor processes the event as required by the Attack Detector and passes the processed event data to the Attack Detector.

d) The Attack Detector tries to detect whether this request is an attack or not by comparing the signature of the event with the available signatures in the Attack Signature Information.
e) If a match is detected a Response is created.

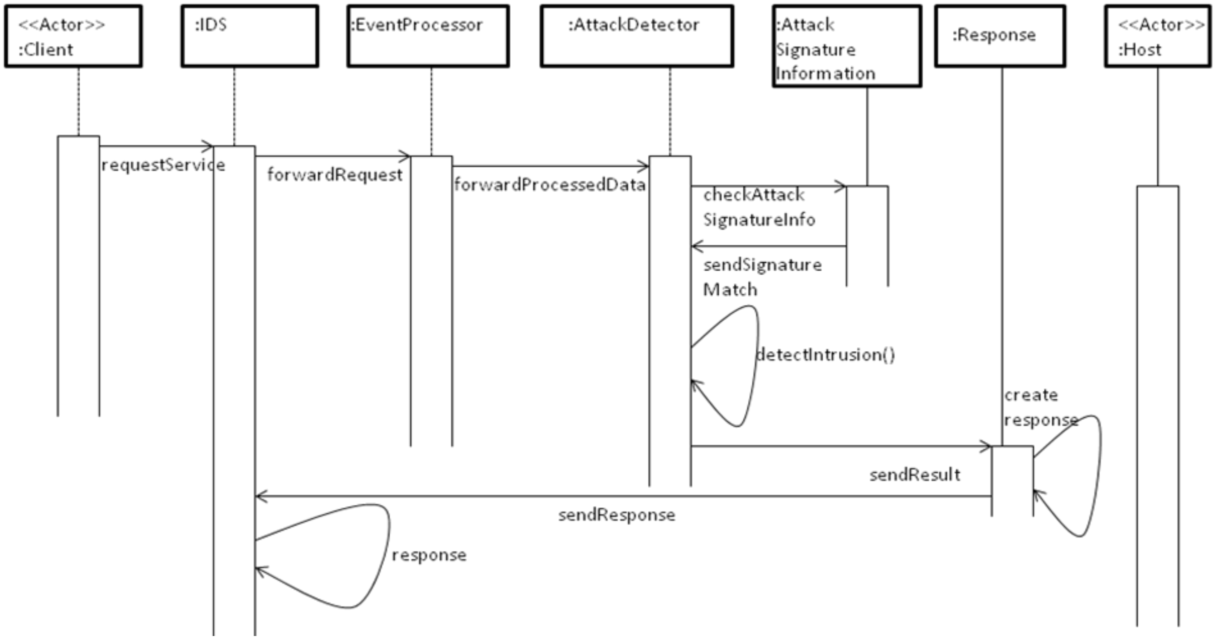

Figure 6. Sequence Diagram for detecting an intrusion with Signature-based IDS.

Alternate Flows:

1) The Attack Signature Information may not be able to detect an attack (a false negative).
2) The Attack Signatures can match and may indicate an attack when no attack is present (a false positive).
3) If no attack is detected, the request for service is forwarded to the host.

Postcondition: If an attack is detected while happening, suitable preventive measures can be adopted.

Example Resolved

We added an Intrusion Detection System besides the existing firewall to the system. Now any request authorized by the firewall is checked against the attack signatures to detect whether the access request is a possible attack. If we detect an attack, an alert can be raised and the firewall can block the request.

Implementation

We first need to create a database with a set of all the known or expected attack patterns. We then select a detection algorithm. Some possible detection algorithms are:

• Expression matching: The simplest form of misuse detection involves searching the event stream for known attack pattern expressions [Ver02].

• State transition analysis: The whole process is a network of states and transitions. Every observed event is applied to finite state machine instances (each representing an attack scenario), possibly causing transitions [Ver02].

• Dedicated languages: Some IDS implementations describe intrusion signatures using specialized languages varying from compiled expressions to programming languages such as Java. A signature takes the form of a specialized program, with raw events as input. Any input triggering a filtering program, or input that matches internal alert conditions, is recognized as an attack [Ver02].

• Genetic algorithms: A genetic algorithm is used to search for the combination of known attacks (expressed as a binary vector, each element indicating the presence of a particular attack) that best matches the observed event stream [Ver02].

Known Uses

An IDS can be combined with a firewall as done in Nokia's network systems [Nok01].

Cisco IDS utilizes detection techniques including stateful pattern recognition, protocol parsing, heuristic detection, and anomaly detection [Cis].

LIDS is a signature-based intrusion detection/defense system for the Linux kernel [Lid].

RealSecure[Rs] by Internet Security Systems is IDS adapted by IBM for intrusion detection packages on the market. It can monitor TCP, UDP and ICMP traffic and, if a match is found, counter-measures can be implemented along with read/write server locking, IP blocking and other measures. This productit is bundled with CheckPoint Software's Firewall [Che].

Consequences

This pattern has the following advantages:

- *Known Attacks:* Detection can be effective against known attacks.
- *Completeness:* If all known attack signatures are available in the database, attacks can be detected in real time.
- *Flexibility:* It is relatively easy to add new attacks to the detection set.

This pattern has the following liabilities.

- It only works for known attacks. A new attack will not be detected. We have to constantly update the database with new attack signatures.
- Some attacks don't have well defined signatures and the attacker may disguise the signatures. This may lead to false positives and false negatives.
- Some attacks may be so fast that it may be hard to recognize them in real time.
- Attack patterns are closely tied to a given environment (operating system, hardware architecture, etc…) and cannot be applied easily to other systems.

Related Patterns

- This pattern is a special (concrete) case of the Reference Monitor [Fer01].
- The patterns for firewalls in [Sch06] complement this pattern.
- The response class could be implemented as a Strategy pattern [Gam94].

4. Behavior-Based IDS

Intent

Check every request for access against patterns of network traffic in order to detect possible deviations from normal behavior (anomaly) which may indicate an attack, and trigger appropriate responses. **AKA:** Anomaly-Based IDS.

Example

A company uses a public network for its applications. The network is exposed to security threats, especially a variety of unknown attacks. Their business could be in jeopardy if their customers realize the fact that their system is not secure enough.

Context

Any network application, where the temporal behavior of network traffic is repetitive and predictable.

Problem

Whenever data is accessed from the Internet or other external networks, there is always a possibility that this access can be harmful to the network. We need to detect possible attacks while they are occurring.

The solution to this problem is affected by the following **forces**:

- *New Attacks:* In today's world, the networks are constantly bombarded with new attacks that do not have a specific attack signature. We need to detect these kinds of attacks.
- *Real-Time:* We need to detect attacks in real time while they are happening, and not after the attack has happened and it is too late to recover from the attack.
- *Increased Vulnerability:* Some networks, e.g. mobile networks are more vulnerable to unknown attacks because of their mobile nature.

Solution

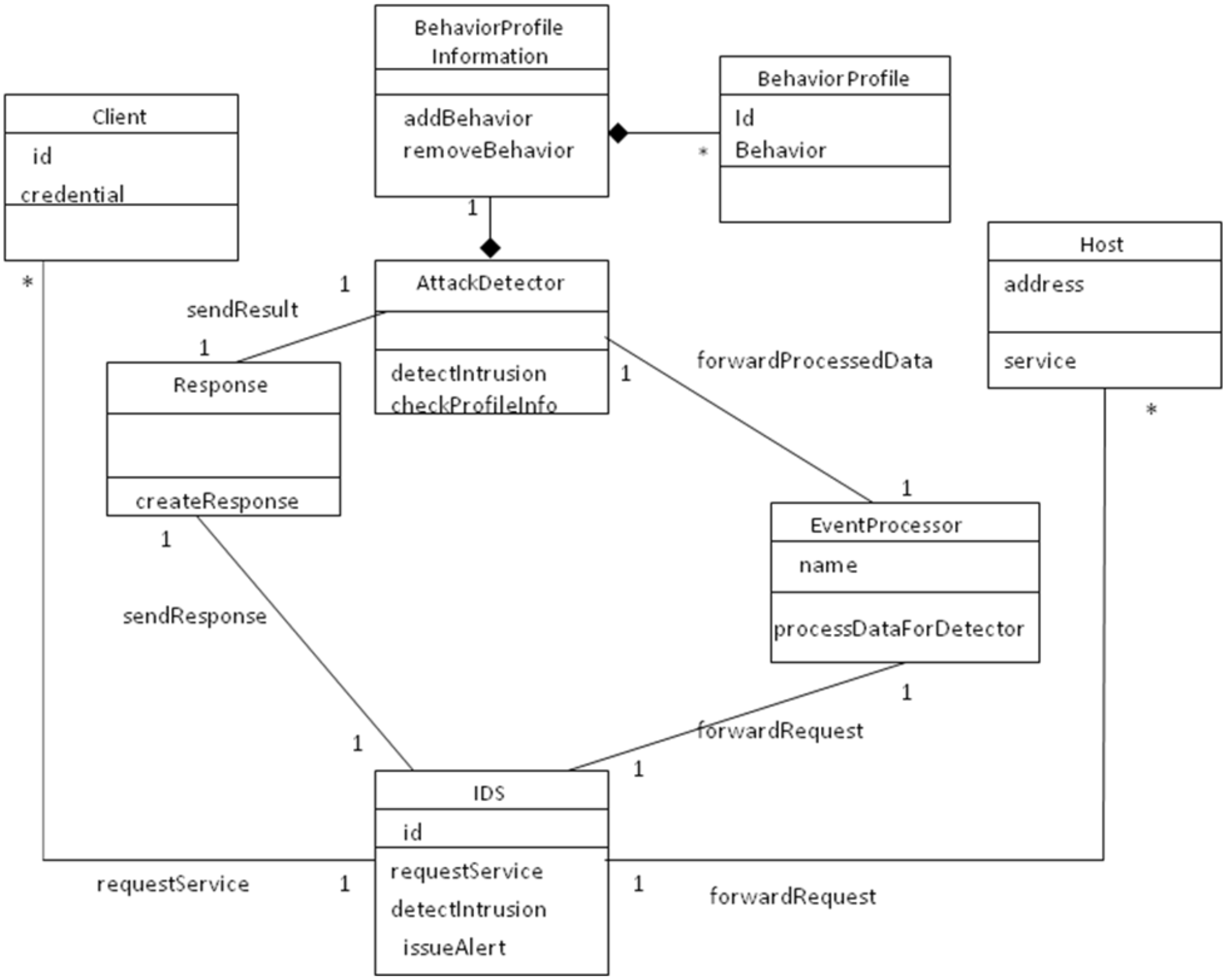

Figure 7. Class Diagram of Pattern for Behavior-based IDS

Observe the traffic over a network and try to find deviations from normal or expected behavior. Any deviation from normal behavior is treated as a sign of intrusion.

Structure

Figure 7 represents the class diagram of this pattern. A **Client** requests some service from the system. The **IDS** intercepts this request and sends it to an **Event Processor**. The **Event Processor** processes the event data as needed by the **Attack Detector** and passes the processed data to the **Attack Detector** which involves a process of establishing profiles of normal behavior which can be compared with the current load in **Behavior Profile Information**. When an attack is detected a **Response** is created.

Dynamics

We present here the dynamic aspects of the Behavior-Based IDS Pattern using sequence diagrams for the following use case:

Detect an intrusion (Figure 8):

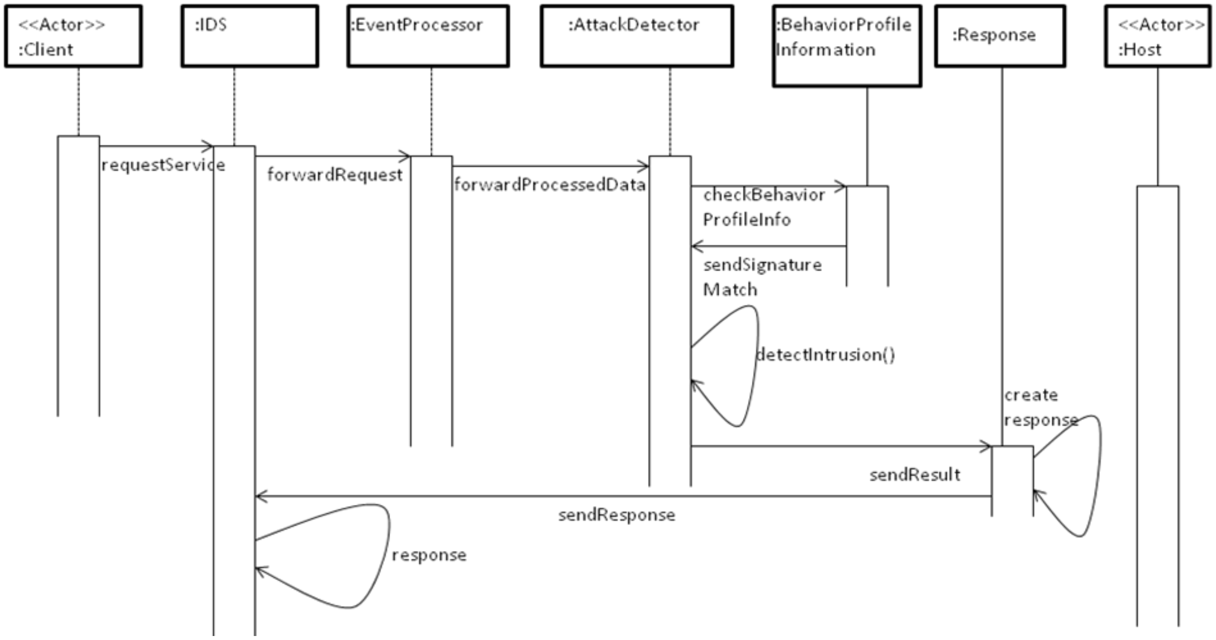

Figure 8. Sequence Diagram of detecting an intrusion with Behavior-based IDS.

Summary: The client requests a service from the host. The Behavior-Based IDS intercepts the message and compares whether the behavior of the request matches a normal behavior profile, if it does not, an attack is suspected and a response is raised.

Actors: Client and Server.

Precondition: We have a set of normal behavior profiles available.

Description:

a) Client makes a service request to the host.
b) The IDS send the request event to an Event Processor.
c) The Event Processor processes the event as required by the Attack Detector and passes the processed event data to the Attack Detector.
d) The Attack Detector tries to detect whether this request is an attack or not by comparing the behavior profile of the request with the available behavior profiles in the Behavior Profile Information.
f) If a match is detected a Response is created.

Alternate Flows:

1) The Behavior Profile Information may not be able to detect an attack (a false negative).
2) The Behavior Profile Information can match and may indicate an attack when no attack is present (a false positive).
3) If no attack is detected, the request for service is forwarded to the host.

Postcondition: If an attack is detected while happening, suitable preventive measures can be adopted.

Example Resolved

We added an Intrusion Detection System to our network. Now all traffic is checked against a normal behavior profile to see whether the access request is an anomaly and hence a possible attack. We are now able to detect many new attacks which do not have a known signature and prevent them.

Implementation

Examples of techniques used for anomaly detection in practice are:

- Genetic Algorithm: In this approach, applications are modeled in terms of different system calls for different conditions such as normal behavior, error conditions and attack conditions. A typical genetic algorithm involves two steps. The first step involves coding the vectors with a string of bits, which forms the input population of the algorithm. The second step is finding a fitness function to test each individual of the population against some evaluation criteria. In the learning process each event sequence of node behavior forms a gene. Fitness is calculated for a collection of genes. If genes with required fitness cannot be found in the current generation, new sets of genes are evolved through crossover and mutation. The process of evolution continues until genes with the required fitness are found. The detection process involves defining vectors for event data and methods of testing whether the vector indicates an intrusion or not [Kis10].
- Protocol Verification: The basis for this approach is the fact that most intruders use irregular or unusual protocol fields, which are not handled properly by application systems [Ver02].
- Statistical Models: These can be either multivariate models or models based on available statistics such as threshold measures, mean and standard deviations of the profile. Clustering analysis where clusters represent similar activities or user patterns is also sometimes used [Ver02].

Known Uses

- Cisco IPS 4200 Series utilizes detection techniques including stateful pattern recognition, protocol parsing, heuristic detection, and anomaly detection [Cis].
- AirTight's wireless IPS automatically detects, classifies, blocks and locates wireless threats using behavior analysis. They use a genetic algorithm to establish normal behaviors [Air].

Some other uses of Anomaly-based IDS are given in Table 1[Gar09].

.

Name	**Manufacturer**	**Hybrid**	**Response**	**Anomaly-related techniques**
AirDefense Guard	AirDefense, Inc.	Y(es)	Y	Detection, correlation and multi-dimensional detection
Barbedwire IDS Softblade	BarbedWire Technologies	Y	Y	Protocol analysis, pattern matching
BreachGate WebDefend™	Breach security	Y		Behaviour-based analysis, statistical analysis, Using correlation functions.
Bro	Lawrence Berkeley National Laboratory	Y	Y	Application level semantics, event analysis, pattern matching, protocol analysis

In the table the "*Hybrid*" column indicates hybrid detection, and the "Response" column indicates that some kind of response mechanism is also available

Table 1. Network-based IDS platforms with anomaly detection functionalities, according to the manufacturer's information [Gar09].

Consequences

This pattern has the following advantages:

- *New Attacks:* Detection can be effective against new attacks which could cause abnormal behavior in the network traffic. For example, we can identify an attack with a behavior such as, when a usually passive web server tries to connect to a large number of addresses it could be the result of a worm attack.
- *Real-Time:* This kind of IDS works well with network traffic that exhibits a normal behavior where it will be easier to detect an abnormal behavior for the network.
- *Increased Vulnerability:* This kind of IDS is usually good in wireless networks that are more vulnerable due to their mobile nature.

This pattern has the following liabilities.

- Lots of false positives. Many anomalies detected are not attacks but could be just unusual behaviors of previous users.
- Cannot be implemented in networks that do not have a predictable traffic pattern.
- The technology adopted for one network is not easily portable to another system and can be different from system to system in a network, as normal behavior for one system is usually not the normal behavior for another system.
- If the attacker does an attack mimicking regular traffic or normal behavior, the attack may go undetected.

Related Patterns

- This pattern is used in conjunction with the Signature-Based IDS pattern.
- Firewalls are usually used along with the IDS in a network. Hence the patterns for firewalls [Sch06] complement this pattern.
- The response class could be implemented as a Strategy pattern [Gam94].

Conclusion

We have created a pattern for the abstract IDS and also patterns for the Signature-Based IDS and the Behavior-Based IDS. These patterns are defined by their class diagrams and the sequence diagrams. These patterns can be catalogued and used as reference for someone or an establishment shopping for a new Intrusion Detection System to complement their existing firewall for enhanced security.

Acknowledgements

We sincerely express our deepest gratitude to our shepherd Joseph, W. Yoder (joe@joeyoder.com) for shepherding this paper and providing valuable insight and feedback for the paper.

AUTHORIZATION AND DISCLAIMER
Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.

References

[Air] Airtight Networks: Solutions > Wireless Intrusion Prevention:

DOI= http://www.airtightnetworks.com/home/solutions/wireless-intrusion-prevention.html

[Bie01] E. Biermann, E. Cloete and L. M. Venter. "A comparison of Intrusion Detection systems", *Computers & Security,* Volume 20, Issue 8, 1 December

2001, Pages676-683

[Che] Checkpoint Software Technologies Ltd:. Checkpoint IPS-1:
DOI = http://www.checkpoint.com/products/ips-1/index.html

[Cis] Cisco Systems: Products and Technologies > Cisco Intrusion Detection:
DOI= www.cisco.com/warp/public/cc/pd/sqsw/sqidsz/

[Fer01] E.B.Fernandez, and R.Pan "A pattern language for security models", *Procs. of the Pattern Languages of Programs Conference (PLoP) 2001,* http://hillside.net/plop/plop2001/

[Fer08] E. B. Fernandez, H. Washizaki, and N. Yoshioka, "Abstract security patterns", *Position paper in Procs. of the 2nd Workshop on Software Patterns and Quality (SPAQu'08), in conjuction with the 15th Conf. on Pattern Languages of Programs (PLoP 2008),* October 18-20, Nashville, TN.
http://patterns-wg.fuka.info.waseda.ac.jp/SPAQU/index.html
Or http://hillside.net/plop/2008/papers/ACMVersions/spaqu/fernandez.pdf

[Gam94] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software", *Addison-Wesley Professional*, 1994

[Gar09] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández and E. Vázquez **"**Anomaly-based network intrusion detection: Techniques, systems and Challenges**",** *Computers & Security* Volume 28, Issues 1-2, February–March 2009, Pages 18–28

[Gra00] Grace, Clive., IT Journalist PC Network Advisor - Tutorial: "Understanding Intrusion Detection Systems":
DOI = http://www.techsupportalert.com/pdf/t1523.pdf

[Kis10] P C Kishore Raja , Dr.M.Suganthi.and M, R.Sunder "Wireless node behavior based intrusion detection using genetic algorithm, " *Ubiquitous Computing and Communication Journal.*
DOI= www.ubicc.org/files/pdf/PCKISHORERAJA_88.pdf

[Lid] Linux Intrusion Detection System:
DOI = http://www.lids.org/

[Mid02] P. Midian. "Getting the most out of Intrusion Detection Systems"**,** *Network Security,* Volume 2002, Issue 7, 1 July 2002, Pages 5-7.

[Nok01] Nokia Inc copyright, White Paper: "Combining Network Intrusion Detection with Firewalls for Maximum Perimeter Protection"**,** April 2001
DOI =
http://www.itu.dk/courses/DSK/F2003/Combining_IDS_with_Firewall.pdf

[Ozg05] Ozgur Depren,, Murat Topallar, Emin Anarim and M. Kemal Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks", *Expert Systems with Applications,* Volume 29, Issue 4, November 2005, Pages 713–722

[Pie08] Roberto Di Pietro and Luigi V Mancini, "Intrusion Detection Systems", 2008 *Springer Science+Business* Media LLC.

[Rs] IBM: Real Secure Intrusion Detection Systems by IBM.
DOI = http://publib.boulder.ibm.com/infocenter/sprotect/v2r8m0/index.jsp

[Sch06] M. Schumacher, E. B.Fernandez, D. Hybertson, F. Buschmann, and P. Sommerlad, *Security Patterns: Integrating security and systems engineering",* Wiley 2006.

[Ver02] T. Verwoerd and R. Hunt **.** Intrusion detection techniques and approaches, Computer Communications, Volume 25, Issue 15, 15 September 2002, Pages 1356-1365.