

# One of These Things is not Like the Others: The State of Anomaly Detection

*Matthew Tanase* 2002-07-01

## One of These Things is not Like the Others: The State of Anomaly Detection

by [Matthew Tanase](#)

last updated July 1, 2002

---

### Introduction

"To some, our observations can be summarized succinctly as "bugs happen". That certainly is not news. But dismissing our results so cavalierly misses the point. Yes, bugs happen. But bugs can be fixed - if they are detected. The Internet is, as a whole, working remarkably well. Huge software packages (i.e., X11R5) can be distributed electronically. Connections span the globe. But the very success of the Internet makes some bugs invisible." - Steven Bellovin [1]

This excerpt, from the well-known 1993 report [Packets Found on an Internet](#), was written nearly nine years ago. As we all know, times have changed. Today, such "bugs", are likely part of an attempt to breach network security. The investigation of strange packets, the cited paper's topic, is now quite common. We know it as intrusion detection. In the past few years, intrusion detection systems have joined firewalls as the fundamental technologies driving network security. In the near future, a third component will emerge - anomaly detection.

### What is Anomaly Detection?

Anomaly detection can be described as an alarm for strange system behavior. The concept stems from a paper fundamental to the field of security - [An Intrusion Detection Model](#), by Dorothy Denning[2]. In it, she describes building an "activity profile" of normal usage over an interval of time. Once in place, the profile is compared against real time events. Anything that deviates from the baseline, or the norm, is logged as anomalous.

### How Does it Differ from Intrusion Detection?

It's easy to compare an Intrusion Detection System (IDS) with an Anomaly Detection System (ADS). They both look for "bad things" on a system or network, things that may be potential security incidents. Each can work well or produce loads of false alarms. And the final results are

the same - a suspicious event is flagged for an administrator to investigate. However, despite these similarities, these tools work quite differently.

An IDS uses a defined set of rules or filters that have been crafted to catch a specific, malicious event. This is often referred to as "misuse" detection. An ADS, on the other hand, operates only from a baseline of normal activity. As described above, behavior that varies from this standard is noted. See the difference? While an IDS looks for a misuse signature, the ADS looks for a strange event.

With these differences in mind, consider the benefits of an ADS. An IDS can only catch events that it has been told to look for, such as a DNS zone transfer or the latest IIS exploit. Anything outside of this list will not be recognized. An ADS, in turn, has the potential to detect new, unknown, or unlisted events. As you can see, it's a powerful tool that can fill the gaps of an IDS. But how does it determine what is normal? What makes an activity strange? The answer to these questions, and the key to an effective ADS, is the profile.

## **Generating Profiles**

The differentiator of anomaly detection technology resides in the methods for constructing a system profile. It should come as no surprise then, that this area attracts the majority of ADS research and development. To date, the modelling of normal behavior can be based on two approaches: statistics and specifications. Both can be applied to single machines, networks, protocols or even applications. However, the techniques used in each method to generate a profile are quite different.

## **Statistics-Based Anomaly Detection**

The earliest approach, proposed by Denning, employs statistics to construct a point of reference for system behavior. The process begins with the training of an anomaly detection sensor. This is accomplished by observing specific events in the monitoring environment such as system calls, network traffic or application usage, over a designated time period. When the interval expires, one of several mathematical methods is used to generate a quantitative measure for the observed data. The result is a baseline for some variable of a system's behavior. With a point of reference in place, the monitoring process is repeated in a live environment. Recorded data is transformed into the same quantitative metric and compared to the baseline. If the deviation exceeds a specified threshold, the event is flagged as anomalous.

Although the process sounds simple, it's not. The first problem is deciding what method should be used to measure deviation. Multiple techniques have been tested, ranging from simple frequency analysis to complex statistical models (see recommended reading). The next problem deals with multivariate correlation. In the simplified explanation above, we measured only one element of a complex system. In order to produce truly useful data, an understanding of multiple variables and their relationships is required. For instance, log-in times, network utilization, and protocol analysis could be used to discover that an intruder has hijacked a valid account in order to transfer files. That's quite a leap to make based on our measured data. An ADS could likely note three independent anomalies, but it would be up to the reviewer to put the pieces together. It's easy to see that training an ADS is a difficult task. But there's a greater problem for statistic-based anomaly detection - a changing environment. As every system administrator can attest, things change. Users are added, services are removed, new machines are introduced. Each of those factors requires an updated profile. In a large, diverse system, an ADS would require constant training.

### **Specification-Based Anomaly Detection**

The second school of thought depends less on mathematics and more on human observation and expertise. Introduced by Calvin Ko [3], this method uses a logic-based description of expected behavior to construct a profile. It requires a language or standard that can be interpreted by the ADS. Using such a syntax, an administrator could construct a list similar to the rules and signatures used by an IDS. But instead of looking for misuse, these rules would ignore normal usage. However, anything outside of the specified behavior, would be marked as anomalous. It's a concept similar to a "Deny-All" firewall, where rules are constructed to block everything but explicitly permitted traffic. A specification-based ADS extends this idea to multiple system elements - an application, a server, network traffic, et cetera. And instead of blocking an event, it would create an alert.

Like the statistical model, there are drawbacks to specification-based anomaly detection. Given the proper syntax, creating a set of rules for normal behavior wouldn't be difficult. But like an IDS, refining those rules to record a minimal amount of non-threatening events (false positives), while continuing to catch odd occurrences is quite difficult. Additionally, the sheer number of variables required to monitor a relatively simple environment is daunting. Rules ranging from general to specific would need to be written for network, application and user behavior. Unlike statistically generated profiles, there is no data set for training. All of a

system's normal elements - uncountable scenarios, would need to be accounted for by the administrator.

## The Current State of Anomaly Detection

Anomaly detection is beginning to appear in commercial and open-source products. One of the most well known Anomaly Detection projects, and my first exposure to the concept, is the [Statistical Packet Anomaly Detection Engine \(SPADE\)](#). Produced by [Silicon Defense](#), this plug-in, for the open source IDS Snort, inspects recorded data for anomalous behavior based on a computed score. It's a great program for anyone new to ADS, since it's open-source and can be implemented with minimal resources.

[Psionic Technologies](#) has built [Login Anomaly Detection \(LAD\)](#) into their HostSentry program. The software learns user log-in behavior, then reacts when it detects strange activity.

[Lancope](#) uses flow-based anomaly detection in their [Stealthwatch](#) product. This program "characterizes and tracks network activities to differentiate abnormal network behavior from normal".

[Okena's Stormwatch](#) is a commercial package that exemplifies specification-based anomaly detection. It interacts with an application and its OS "to make a real-time allow/deny decision according to the customer's application security policy".

Distributed projects such as [SecurityFocus DeepSight Threat Management System](#) make use of statistical data to detect potential Internet threats. This is a fascinating area of both IDS and ADS, since the security community works together to produce dangerous trend warnings. At the lowest level - the sensors, this is pure intrusion detection. They are reporting actual scans and attacks. Taken as a whole, however, this turns out to be a massive anomaly detection system. In this case, the entire Internet is the system, and the individual incidents are statistical anomalies.

Without a doubt, anomaly detection techniques are also being incorporated into modern intrusion detection systems. While they might not be advertised specifically as an ADS, IDS products of the near future will generate alerts based on deviant system behavior.

## Conclusion: The Future of Anomaly Detection

Anomaly Detection will play a key role in advancing the capabilities of security technology. As the number of threats grows and diversifies, an ADS becomes a required element of system security. Security professionals need technology that can identify both subtle variations and new breeds of threats in the wild, much like modern anti-virus software.

Thankfully, Anomaly Detection is moving from research and development into production and application. It will be used in conjunction with firewalls and intrusion detection systems to create a robust defence for networks and hosts. The lines separating these distinct areas of computer security are starting to blur. Vendors and open source projects are working to correlate the techniques and data of each tool. In time, components of all three will be used to transparently enforce policy, report misuse and protect against unknown threats.

*Matthew Tanase, CISSP is President of [Qaddisin](#), a network security company based in St. Louis. His company provides consulting services for several organizations. Additionally, he produces [The Security Blog](#), a daily weblog dedicated to security.*

## References

- [1] Steven Bellovin. ["Packets Found on an Internet"](#), 1993.
- [2] Dorothy Denning. ["An Intrusion-Detection Model"](#), IEEE Symposium on Security and Privacy, 1986.
- [3] Calvin Ko, Manfred Ruschitzka and Karl Levitt. ["Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-based Approach"](#), IEEE, 1997.

## Recommended Reading

Paul Barford and David Plonka. ["Characteristics of Network Traffic Flow Anomalies"](#), University of Wisconsin, Madison.

Stephanie Forrest, Stevent Hofmeyr, Anil Somajaji and Thomas Longstaff. ["A Sense of Self for Unix Processes"](#), IEEE Symposium on Security and Privacy, 1996.

Nong Ye and Qiang Chen. ["An Anomaly Detection Technique Based on Chi-Square Statistic for Detecting Intrusions Into Information Systems"](#), John Wiley and Sons, Ltd., 2001.

Syed Masum Emran and Nong Ye. ["Robustness of Canberra Metric in Computer Intrusion Detection"](#), IEEE Workshop on Information Assurance and Security, USMA, 2001.

Luca Deri, Stefano Suin and Gaia Maselli. "[Design and Implementation of an Anomaly Detection System: An Empirical Approach](#)".

George Noel, Steven Gustafson and Gregg Gunsch. "[Network-Based Anomaly Detection Using Discriminant Analysis](#)", Graduate School of Engineering and Management, Air Force Institute of Technology.

Christopher Krugel, Thomas Toth and Engin Kirda. "[Service Specific Anomaly Detection for Network Intrusion Detection](#)", ACM, 2002.

Leonard LaPadula, "[State of the Art in Anomaly Detection and Reaction](#)", MITRE Corporation, 1999.

## Relevant Links

[Snort](#)

[Silicon Defense](#)

[Dsheild.org](#)

[DeepSight Analyzer](#)

[Psionic Technologies](#)

[Lancope](#)

[Okena](#)

[Privacy Statement](#)

Copyright 2006, SecurityFocus