

Sound the Alarm

A Survey of Modern Intrusion Detection Methodologies

Erin Jamroz University of Puget Sound
1500 N. Alder St.
Tacoma, WA
ejamroz@ups.edu

Jacob Fuhrman University of Puget Sound
1500 N. Alder St.
Tacoma, WA
jfuhrman@ups.edu

ABSTRACT

1. INTRODUCTION

Intrusion detection is the process of detecting unauthorized use of a system and alerting the proper authorities of such misuse. Intrusion detection systems (IDS) are the systems used to detect these misuses and aid in their defence.

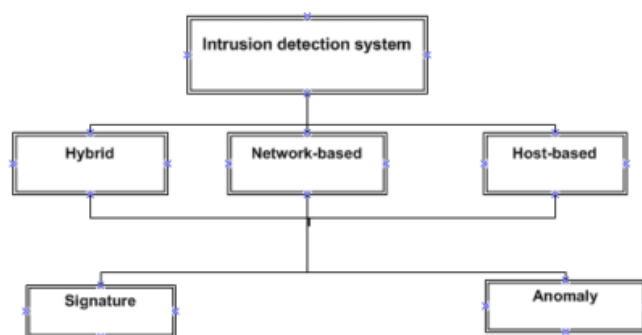


Figure 1: A general breakdown of IDSs by Audit System and Event Analysis. Image taken from [2]

1.1 Audit Systems

This section will breakdown IDSs by their audit system, which describes the source of the data that they analyse for intrusions. There are three main types of audit systems: *Host-Based*, *Network-Based*, and *Application-Based*.

1.1.1 Host-Based

Host-Based IDSs (HIDS) are characterised by having detection sensors on each individual machine within a network. These types of systems monitor an individual computer's system logs to detect attacks. HIDSs are able to detect attacks with much more specificity than network-based systems because they are able to monitor individual processes on an particular machine for malicious activity.

There are many advantages to using a HIDS. First, and fore most, they are able to detect attacks that network-based sensors cannot, because they are running on the host machine. Next, unlike network-based sensors, they are fully functional within both switched, and encrypted networks because their audit source is independent of network traffic, and they are able to use host resources to decrypt traffic before analysis.

1.1.2 Network-Based

Network-based IDSs (NIDS) are characterised by having detection sensors placed at network hubs, such as routers, rather than on each individual machine in a network. These types of systems work by monitoring network traffic looking for patterns in flow or analysing packet headers for trends. These are the most commonly used types of systems today.

NIDSs have many advantages over a host-based system, mainly that fewer sensors are required to cover the entire network, which also means less installation time and maintenance. These types of systems also have little to no impact on network performance, because they are sensors that traffic is simply routed through. As they are their own independent system residing on the network, they can be set up in such a way that they are invisible to outsiders. This hidden nature also makes them very insulated from attack themselves, which is a large advantage for network administrators.

However, these systems are not without their drawbacks. It has been shown that in periods of high network load, these systems performance drops due to an inability to process all incoming packets. As such, they could fail to detect an attack that was launched during one of these periods. The effectiveness of these types of systems also drops in switched networks, because switches subdivide network traffic such that a NIDS can no longer monitor all traffic on the network because much of it is hidden behind a switches. Finally, NIDSs are unable to analyse encrypted traffic. This incapability is a major drawback, especially in this day in age, as more and more traffic is being moved to encrypted channels of communication.

1.1.3 Application-Based

1.2 Event Analysis

This section will further subdivide and discuss IDSs by event analysis, which describes the method of detection used, given

a particular audit source of data. These detection methods fall into two major categories: *Signature Detection* and *Anomaly Detection*. Signature based methods are much more common in practice, but anomaly detection is an area of much research and promise. A summary of both strategies can be found in 2.

1.2.1 Signature Detection

1.2.2 Anomaly Detection

Anomaly detecting systems work by constructing a system profile of "normal" behavior, and use that profile to then detect abnormal behavior. The idea is that attacks are a subset of abnormal behavior, which will allow these types of systems to detect them. Most of these systems use some complex statistical analysis to determine when activity differs from the system profile. The major draw to these types of systems is that, unlike signature detection methods, these systems can detect not only variations on known attacks, but completely novel attacks as well. The problem with this functionality is that it is predicated on having an effective system profile, which is by no means trivial to produce. They require extensive training sets to learn from, of which few exist. In practice, these types of systems produce large numbers of false positives that require manual inspection to effectively classify as an attack or not. For this reason, few existing systems use this type of detection. Instead IDSs using this type of detection mechanism are more an open area of research than production level systems.

Method	Detection time	Reliability	Detect new attacks	False Positive	Requirements
Signature	Fast	Yes	No	Very low	Well-known signature
Anomaly	vary	Yes	Yes	High	Trained data

Figure 2: Summary of Event Analysis systems. Image taken from [2]

1.3 Host-Based Audit Systems

Host based audit systems are audit systems that rest on the host that they monitor, and thus analyze the activity on only one computer. They are primarily used due to the extremely high granularity of analysis they provide, as they can observe and monitor every aspect of a running system, from running processes and services, to who launched them and when. In the case of many attacks on an individual system, this finely detailed analysis is essential, as it allows host-based systems to detect attacks that cannot be observed from anywhere but the host itself (such as network traffic in a network-based system, which will be explained in the next section). Host-based audit systems are also compatible with systems that converse using encrypted traffic, if configured properly, as the traffic needs to be decrypted at some point before reaching the user, at which point the host-based system can intercept it. This way the traffic can remain secure within a layer of encryption, and the host can remain secure and analyzable by the audit system, creating a solid series of defense mechanisms that do not conflict with one another. Similarly, host-based audit systems can operate without handicaps of any kind in switched networks, as

the physical layout of the network does not affect the host system, a point that will also be made more relevant once network-based audit systems are explained.

However, host-based audit systems do lend themselves to several key flaws. The first flaw, and arguably the most regrettable considering the impressive list of strengths listed above, is the fact that host-based systems are vulnerable to attack. Simply, the host-based system resides on the system it monitors, and thus a clever attacker, if they can locate where the software sits on the system, can disable or perhaps even destroy a host-based audit system if they gain access to the computer it rests on. The second flaw is that host-based systems, by design, have only the resources of the host system they reside on and monitor available to perform their analysis with, and cannot be supplemented in this task. Thus in the perhaps rare scenario in which one possess an important system that needs to be protected at the host-level, but the system is composed of low-performance hardware, a host-based solution is simply not feasible. Moreover, on systems with reasonably powerful hardware, one could imagine the scenario in which the system, if it is acting as a webserver or providing services to a large number of clients, may experience a large enough request for said services that its resources could become tied up in managing server-client interactions to the extent that a host-based system is unable to perform its analysis. This, of course, is a critical flaw in the system's design, as the behavior of the increasingly prevalent Denial of Service attack exploits just this sort of resource-binding potential. Thus the efficacy of host-based systems is predicated by the amount of resources a system has to work with, making such systems incredibly vulnerable to Denial of Service attacks.

1.4 Network-Based Audit Systems

1.5 Application-Based Audit Systems

1.6 Signature Detection

1.7 Anomaly Detection

2. RELATED WORK

2.1 Probes

This section is will discuss various approaches to detecting Probe attacks. Probes have been classified as any passive information gathering intrusion. The most common form of probe attack, one which will be discussed at length, is the port scan. The motivation for detecting these types of scans is that any intelligent attacker wishing to launch a successful attack will gather background information about the victims system before launching his or her attack. Their primary means of gathering information is by first launching a port scan.

There are 65,536 standardly defined ports on a given machine. These are broken down into three large categories: *a.* Well-known ports(0-1,023), *b.* Registered ports (1,024-49,151), *c.* Dynamic and/or private ports (49,152-65,536) [Matiti, P Lecture notes from Surveying Port Scans Methodologies]. Essentially, a port scan consists of sending a message to each of these ports and analyzing the response message to gain information about what services the victim machine is running. For this reason, TCP ports are the most often scanned because TCP is a connection-oriented protocol, meaning that its response messages are more useful.

Added to this, it is easy to block/detect UDP traffic at the firewall level [5].

2.2 Privilege Escalation

Privilege Escalation attacks are described in two general categories in this paper: Remote to User Attacks and User to Root Attacks. In the broadest sense, Remote to User Attacks are when an attacker seeks to gain local user access to a machine that they have network access to. A User to Root Attack, as one may infer from its name, is when an attacker who already possesses local user access seeks to escalate their privileges to those of a root user, IE gain root user access. In the following sections, both types of attacks are explained in detail.

2.2.1 Remote to User

A Remote to User attack is when an attacker who has network access to a system but does not have an account on that machine exploits a vulnerability on the system to gain unauthorized local access as a user of the target system. These types of attacks come in many forms, and can be as simple as getting valid user authentication information through guessing a user's password with a dictionary attack. However, just as common are attacks that exploit a vulnerability in a common and innocuous system service, such as FTP to gain local user access. These exploits can change system settings to allow an attacker remote access, but they can also trigger buffer overflows, which in many circumstances allow an attacker to execute arbitrary code on the remote host (which is often used to gain root access, as we will discuss later).

2.2.2 User to Root

User to root attacks

2.3 Denial of Service

3. METHODS

4. EVALUATION

5. DISCUSSION

5.1 Training Sets

5.2 Effective Defense

5.3 Open Problems

6. CONCLUSIONS

7. REFERENCES

- [1] 802.11 Denial-of-Service Attacks: Vulnerabilities and Practical solutions.
- [2] M. Alenezi and M. Reed. Methodologies for detecting DoS/DDoS attacks against network servers. *ICSNC 2012: The Seventh International Conference on Systems and Networks Communications*, (c):92–98, 2012.
- [3] S. Alexander. Trust engineering: rejecting the tyranny of the weakest link. *Proceedings of the 28th Annual Computer Security ...*, pages 145–148, 2012.
- [4] M. Bernaschi, E. Gabrielli, and L. Mancini. Operating system enhancements to prevent the misuse of system calls. *... of the 7th ACM conference on ...*, 2000.
- [5] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita. Surveying Port Scans and Their Detection Methodologies. *The Computer Journal*, 54(10):1565–1581, Apr. 2011.
- [6] S. Brugger and J. Chow. An assessment of the DARPA IDS Evaluation Dataset using Snort. *UCDAVIS department of Computer Science*, pages 1–19, 2007.
- [7] N. Carlini, A. Felt, and D. Wagner. An evaluation of the google chrome extension security architecture. *... the 21st USENIX Conference on Security*, 2012.
- [8] M. Dalton, C. Kozyrakis, and N. Zeldovich. Nemesis: preventing authentication & access control vulnerabilities in web applications. *... of the 18th conference on USENIX ...*, 2009.
- [9] Y. Djemaiel and N. Boudriga. A global marking scheme for tracing cyber attacks. *Proceedings of the 2007 ACM symposium on Applied computing - SAC '07*, page 170, 2007.
- [10] W. Eddy. TCP SYN flooding attacks and common mitigations. 2007.
- [11] M. Gandhi and S. Srivatsa. Detecting and preventing attacks using network intrusion detection systems. *International Journal of Computer Science and ...*, (2):49–60, 2008.
- [12] L. A. Gordon, M. P. Loeb, W. Lucyshyn, and R. Richardson. COMPUTER CRIME 2004 CSI / FBI Computer Crime and Security Survey. 2004.
- [13] S. Hansman and R. Hunt. A taxonomy of network and computer attacks. *Computers & Security*, 24(1):31–43, Feb. 2005.
- [14] F.-h. Hsu and T.-c. Chiueh. Scalable Network-based Buffer Overflow Attack. pages 163–171, 2006.
- [15] A. Hussain, J. Heidemann, and C. Papadopoulos. A framework for classifying denial of service attacks. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '03*, page 99, New York, New York, USA, 2003. ACM Press.
- [16] S. Jana, Suman; Kaseram. On Fast and Accurate Detection of Unauthorized Wireless Access Points Using Clock Skews and Categories and Subject Descriptors. pages 104–115.
- [17] M. Kirkpatrick and S. Kerr. Enforcing physically restricted access control for remote data. *Proceedings of the first ACM conference on Data ...*, pages 203–212, 2011.
- [18] B. Kuperman and C. Brodley. Detection and prevention of stack buffer overflow attacks. *Communications of the ...*, 48(11), 2005.
- [19] K. Labib and V. Vemuri. Detecting and visualizing denial-of-service and network probe attacks using principal component analysis. *Proc. SAR*, 2004.
- [20] U. Larson and D. Nilsson. Securing vehicles against cyber attacks. *... of the 4th annual workshop on Cyber security ...*, page 1, 2008.
- [21] V. Livshits and M. Lam. Finding security vulnerabilities in Java applications with static analysis. *... of the 14th conference on USENIX Security ...*, 2005.
- [22] M. McDowell and This. Security Tip (ST04 015) Understanding Denial of Service Attacks. page

2013, 2013.

- [23] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage. Inferring Internet denial-of-service activity. *ACM Transactions on Computer Systems*, 24(2):115–139, May 2006.
- [24] R. Newman. Cybercrime, identity theft, and fraud: practicing safe internet-network security threats and vulnerabilities. . . . of the 3rd annual conference on *Information security* . . . , 2006.
- [25] D. K. Nilsson and U. E. Larson. Conducting Forensic Investigations of Cyber Attacks on Automobile In-Vehicle Networks. *International Journal of Digital Crime and Forensics*, 1(2):28–41, 2009.
- [26] R. Richardson and C. Director. CSI computer crime and security survey. *Computer Security Institute*, 2008.
- [27] R. Sailer, T. Jaeger, X. Zhang, and L. van Doorn. Attestation-based policy enforcement for remote access. *Proceedings of the 11th ACM conference on Computer and communications security - CCS '04*, page 308, 2004.
- [28] E. Silenok, Elena; Roedel, Chris; Silenok. Detection and Characterization of Port Scan Attacks Cynthia Bailey Lee Chris Roedel Classification Methodology.
- [29] S. Specht and R. Lee. Distributed denial of service: Taxonomies of attacks, tools, and countermeasures. . . . on *Parallel and Distributed* . . . , (September):543–550, 2004.
- [30] S. Staniford, J. Hoagland, and J. McAlerney. Practical automated detection of stealthy portscans. *Journal of Computer Security*, 10:105–136, 2002.
- [31] A. Stock, J. Williams, and D. Wichers. OWASP top 10. *OWASP Foundation*, July, 2013.
- [32] F. Sun, L. Xu, and Z. Su. Static detection of access control vulnerabilities in web applications. *Proceedings of the 20th USENIX conference on* . . . , 2011.
- [33] P. Taylor and P. Mell. UNDERSTANDING INTRUSION DETECTION SYSTEMS NEWSLETTER. Technical Report April 2013, The EDP Audit, Control, and Security, 2006.
- [34] G. Vigna and R. Kemmerer. NetSTAT: a network-based intrusion detection approach. *Proceedings 14th Annual Computer Security Applications Conference (Cat. No.98EX217)*, pages 25–34, 1998.
- [35] D. Wagner and R. Dean. Intrusion detection via static analysis. *Security and Privacy, 2001. S&P 2001. . . .*, 2001.