

*The Definitive Guide to Understanding,  
Selecting, and Deploying Intrusion Detection in the Enterprise!*

# THE PRACTICAL Intrusion Detection HANDBOOK

- Product selection, planning, and operations
- Filled with real-life cases and stories of intrusion detection systems in action
- Covers host-based and network-based intrusion detection



**PAUL E. PROCTOR**

Foreword by **Dorothy Denning**, author of *Cryptography and Data Security* and *Information Warfare and Security*

Technical Editor: **Ira Winkler**, author of *Corporate Espionage*

# Practical Intrusion Detection Handbook

*Paul E. Proctor*



*Prentice Hall PTR*  
*Upper Saddle River, New Jersey 07458*  
*www.phptr.com*



Library of Congress Cataloging-in-Publication Data Available

Acquisitions Editor: *Tim Moore*  
Production Editor: *Rose Kernan*  
Cover Design: *Talar Agasyan*  
Cover Design Director: *Jerry Votta*  
Manufacturing Manager: *Alexis R. Heydt*  
Internal Graphics: *Glen VanHouten*  
Technical Copy Editor: *Sue Heim*



© 2001 Prentice Hall PTR  
Prentice-Hall, Inc.  
Upper Saddle River, New Jersey 07458

Prentice Hall books are widely used by corporations and government agencies for training, marketing, and resale.

The publisher offers discounts on this book when ordered in bulk quantities. For more information, contact Corporate Sales Department, phone: 800-382-3419; fax: 201-236-7141; e-mail: [corpsales@prenhall.com](mailto:corpsales@prenhall.com) or write: Prentice Hall PTR, Corporate Sales Department, One Lake Street, Upper Saddle River, NJ 07458.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

All product names mentioned herein are the trademarks or registered trademarks of their respective owners

Printed in the United States of America  
10 9 8 7 6 5 4 3

ISBN 0-13-025960-8

Prentice-Hall International (UK) Limited, **London**  
Prentice-Hall of Australia Pty. Limited, **Sydney**  
Prentice-Hall Canada Inc., **Toronto**  
Prentice-Hall Hispanoamericana, S.A., **Mexico**  
Prentice-Hall of India Private Limited, **New Delhi**  
Prentice-Hall of Japan, Inc., **Tokyo**  
Pearson Education Asia Pte. Ltd.  
Editora Prentice-Hall do Brasil, Ltda., **Rio de Janeiro**

**Malformed Packets.** Malformed packets come in a variety of shapes and sizes with the intent of causing a protocol stack to crash. Network protocols are complicated pieces of code, and it's difficult to handle all the different types of error conditions that can arise. In most cases, programmers do not attempt to handle impossible situations such as null arguments in critical fields. Hackers take advantage of this by creating these very situations, causing the protocol to fail. Results range from hung networking to machines that crash. There are a number of DOS tools that use this technique available on the Internet and go by names like *land*, *bonk*, and *bink*.

**Packet Flooding.** Packet flooding is a simple DOS technique that involves sending as many packets as you can at a single network device until it either crashes because it can't handle the load or becomes so slow that legitimate user requests can't get through. This is not a very sophisticated attack, and it is fairly easy to detect and defend against by denying access to the source computer sending the packets. However, if the attacker is spoofing the source address, then it may be very hard to find out where the packets are originating.

**Distributed Denial of Service.** A special case of packet flooding is the distributed denial of service attack in which a number of computers are used to attack all at once. If the source IP addresses are spoofed, then it can be difficult to shun and defend against distributed DOS. Many times the attacking computers are unwitting drones created by Internet viruses. Network intrusion detection is not a panacea against this attack, but it is a vital tool in both detection and response.

#### ARCHITECTURE

Network-based intrusion detection systems consist of sensors deployed throughout a network that report to a central console. Sensors are usually self-contained detection engines that obtain network packets, search for patterns of misuse, and then report alarms to a central command console. There are two types of architectures: network node and traditional sensor architectures. Traditional sensor-based architectures are also known as promiscuous-mode network intrusion detection systems, or network taps.

Network-node systems place an agent on each computer in the network to monitor traffic bound only for the individual target. Sensor-based systems monitor whole network segments. Sensor-based systems are not widely distributed because there are relatively few segments to monitor, while network-node systems are widely distributed onto every mission-critical target.

### Traditional Sensor-Based Architecture

Figure 3-1 shows traditional sensor-based intrusion detection architecture. A sensor, usually an Ethernet chip set to promiscuous mode, is used to “sniff” packets off the network where they are fed into a detection engine, typically on the sensor machine itself. Taps are distributed to the various mission-critical segments of the network, usually one per segment. A central console is used to correlate alarms from multiple sensors. We can follow the lifecycle of a network packet through the system (Figure 3-1).

1. A network packet is born. This takes place when one computer communicates with another computer.
2. The packet is read in real-time off the network through a sensor that lies on a network segment somewhere between the two communicating computers. The sensor is usually a stand-alone machine or network device.
3. A sensor-resident detection engine is used to identify predefined patterns of misuse. When a pattern is detected an alert is generated and forwarded to the central console.
4. The security officer is notified. This can be done through audible, visual, pager, e-mail, SNMP, or any other number of different methods.

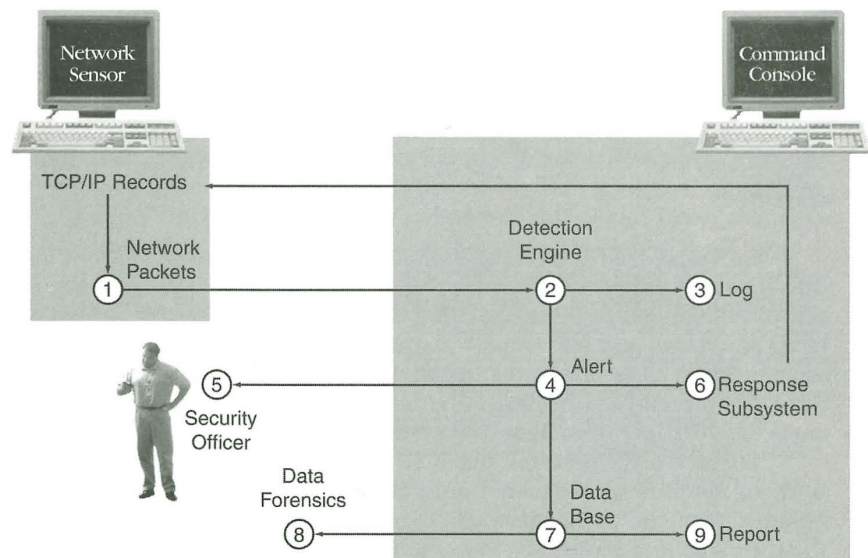


Figure 3-1. A Standard Network Intrusion Detection Architecture

5. A response is generated. The response subsystem matches alerts to predefined responses or can take direction from the security officer to execute a response. Responses include actions such as reconfiguring the router or firewall to refuse traffic from a particular source address.
6. The alert is stored for later review and correlation.
7. Reports are generated summarizing alert activity.
8. Data forensics is used to look for long-term trends. Some systems allow archiving of the original traffic to replay sessions.

#### DISTRIBUTED NETWORK-NODE ARCHITECTURE

In early 1999, all commercial network intrusion detection systems used promiscuous-mode sensors. However, these technologies were subject to packet loss on high-speed networks. A new distributed architecture for network intrusion detection surfaced in mid-1999 that deals with the performance problem on high-speed networks by distributing sensors to every computer on the network. In network-node intrusion detection each sensor is concerned only with packets directed at the target on which it resides. The sensors then communicate with each other and the main console to aggregate and correlate alarms.



#### KEY POINT

The difference between network- and host-based intrusion detection is the source of the data, not the location of the sensor or mode of operation. Network intrusion detection systems process TCP/IP packets. Host-based systems process event logs from operating systems and applications.

This network-node architecture has added to the confusion over the difference between host- and network-based intrusion detection. A network sensor running on a host does not make it a host-based sensor. Network packets directed to a host and sniffed at a host are still network intrusion detection.

Figure 3-2 shows network-node intrusion detection architecture. An agent is used to read packets off the TCP/IP stack at a layer where the packets have been reassembled. The packet is fed into a detection engine located on the target machine. Network-node agents communicate with each other on the network to correlate alarms at the console.

1. A network packet is born.
2. The packet is read in real-time off the network through a sensor resident on the destination machine.

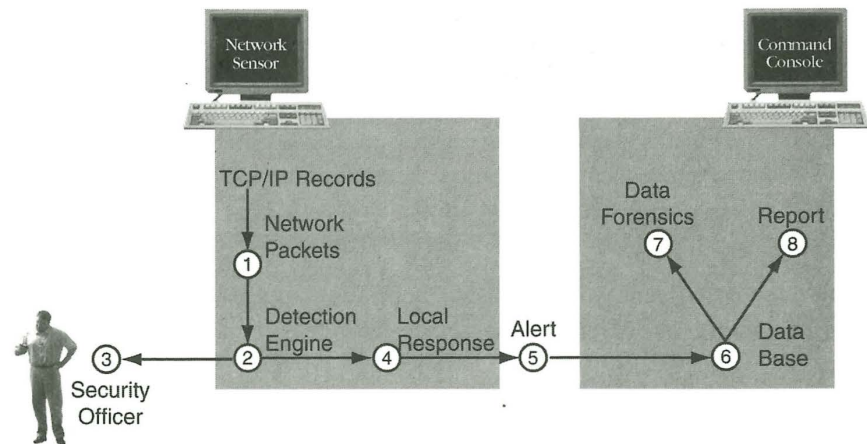


Figure 3-2. A Distributed Network-Based/Host Resident Intrusion Detection Architecture

3. A detection engine is used to identify predefined patterns of misuse. When a pattern is detected an alert is generated and forwarded to a central console or to other sensors in the network.
4. The security officer is notified.
5. A response is generated.
6. The alert is stored for later review and correlation.
7. Reports are generated summarizing alert activity.
8. Data forensics is used to look for long-term trends.



#### REAL-TIME INTRUSION DETECTION?

There is a common misconception in the market that network intrusion detection is the definition of intrusion detection. Most commercial network intrusion detection systems run in real-time because of the relatively low number of passive sensors. Host-based systems run in a combination of real-time and batch, or scheduled processing. Many people have the misconception that the differentiator between network- and host-based intrusion detection is real-time versus non-real-time. This results in many conversations such as the following:

"I need intrusion detection."

"Are you interested in host-based or network intrusion detection?"

"Oh, I need real-time intrusion detection."

"Great. On the network or the host?"

"What?"