Chapter 11

Summary

For NIDES to find wide use in sensitive applications in the federal government and in private companies working in partnership with federal agencies, the system must meet several basic requirements. It must be able to detect a variety of intrusion types with high detection accuracy, including external penetrators and malicious insiders; it must be capable of detecting such intrusions or intrusion attempts within minutes of their having occurred; it must be able to display and interpret current and past results; it must be target-system independent to the extent possible; and it must be customizable for use in new and changing environments.

10.4 The NIDES Test Facility

NIDES must provide a test facility to allow the SO to experiment with candidate NIDES configurations before making them effective. The test facility must allow a candidate NIDES configuration to be tested using test audit data sets, by the SO, from the audit data archive. The test facility must produce reports of test runs so that the SO can compare the results of different candidate configurations for NIDES. NIDES must be capable of running test configurations in parallel with the real-time NIDES instance that is monitoring target systems for suspicious activity.

Chapter 10

Customization

One of the basic requirements for NIDES is to provide the security officer with the capability to customize the system and to tune its components for best performance in each new environment. This requirement is met by the NIDES customization and test facilities.

10.1 Runtime Configuration of Statistical Component

NIDES must allow the SO to turn on or off specific intrusion-detection measures while NIDES is running. NIDES must also allow the SO to change the parameters of the real-time executing NIDES instance. Such parameters include aging factors and thresholds.

10.2 Runtime Configuration of Rulebased Component

NIDES must allow the SO to modify the NIDES rulebase by adding, deleting, or modifying rules. NIDES must be capable of making such modifications to the runtime system. NIDES must also allow the SO to activate and deactivate predefined rules in the real time running system.

10.3 Runtime System Monitoring and Control

NIDES must allow the SO to view and modify the current NIDES configuration. The SO must be able to view the status of the analysis components, the list of systems being monitored, the rate of audit data collection, and the rate of alert reporting. The SO must be able to configure the set of target machines being monitored.

9.5 Dynamic Addition and Deletion of Target Systems

The NIDES realm interface must dynamically recognize target systems that either leave or join the realm. This is necessary because of target system failures (crashes and reboots) as well as SO intervention (changing the configuration of target systems).

9.6 Sources of Audit Data

NIDES must be capable of accepting as input audit data from all relevant sources. Activities can be observed at different levels of a system's software hierarchy. In general, the higher the level of audit data collection, the more insight is provided into the user's activities. Analyzing lower-level audit data requires NIDES to recognize a higher-level activity from low-level information. However, different types of intrusive behavior will be evident by examining audit data collected at different target system levels. In general, when determining which level to audit, there are several tradeoffs to consider, including the types of intrusions that can be detected, the complexity and volume of audit data that is collected, the ability to appeal to intuition when an alarm is raised, the ability to formulate rules that characterize intrusions, the ability to "play back" suspicious activity, and the ability to perform damage assessment or gather evidence. Thus, NIDES must be capable of processing data collected from multiple system levels.

NIDES must be easily adaptable so that it can accept the following sources of audit data:

- Operating system audit trails, some of which may be C2-level or above; typically, these audit trails collect information at the system-call level in the operating system
- Other operating system logs — for example, accounting logs
- Logs kept by various security packages
- Database system logs
- Application logs
- Network-level logs

Chapter 9

Target System Realm Interface

The NIDES realm interface is a logical component of NIDES that interfaces all sources of audit data with NIDES.

9.1 Generic Audit-Record Format

NIDES must use a standardized representation of audit trails to process audit data from various audit data sources and from various target system types. The NIDES realm interface must convert target-specific audit record formats into a generic, standardized format, to provide target-system independence.

9.2 Audit Data Transfer

The NIDES realm interface must provide an audit data transfer mechanism from target systems to NIDES. The NIDES realm interface must transmit audit records to NIDES as soon as possible.

9.3 Filtering

The NIDES realm interface must be modifiable to filter the source audit data stream if necessary, so as to discard irrelevant audit data.

9.4 Robustness

The NIDES realm interface must be capable of dealing with short peaks in the rate of audit data generation. If network bandwidth is limiting or NIDES computational resources are insufficient, then data must be queued at the target until it can be transmitted and processed.

Chapter 8

Data Management Facility and Archiver

The NIDES data management facility supports the archiving of audit data, analysis results, and alert data. It also supports security officer browsing and querying of the archived data.

8.1 Persistent and Efficient Storage of Archived Data

NIDES must maintain a persistent store of audit records, analysis results, and alert reports. The archived files must be labeled using a naming convention that identifies their contents.

8.2 Querying and Retrieval of Archived Data

NIDES must support SO browsing of its archive of audit records, analysis results, and alert reports. NIDES must support SO querying of the archive by user name or user name set, by desired time window, or by a combination of these.

Chapter 7

Resolver

The nature of the audit data is such that large numbers of audit records may be generated for a single action of a monitored user. Thus, once the NIDES statistical component determines that a user has become anomalous (i.e., the user's statistical score has crossed the threshold for the user), it is likely to continue finding the user to be anomalous for each of many audit records in sequence. Clearly, such anomalies do not signify independent events of concern, and in fact are related to the same set of recent user activity. To eliminate the need for the SO to review and acknowledge each of these redundant anomalies, the NIDES resolver receives the alerts reported by the two NIDES analytical components and filters out such redundant alarms. Thus, the NIDES resolver must filter out redundant alerts before they are reported to the user interface.

sequence number of the audit record that caused the alert to be triggered.

In addition, when an alarm is raised, NIDES must indicate to the SO what caused the event to be reported. When an alert is reported because of the triggering of one or more rules in the NIDES rulebase, NIDES must provide the SO with a brief explanation of that rule and the observed values that caused it to be triggered. When an alert is reported because of a detected statistical component anomaly, NIDES must identify to the SO the user's computed statistical score value, the statistical threshold for the user, and which intrusion-detection measures contributed most to the activity's high statistical score.

6.4 Querying for More Information

The NIDES security officer must be able to query NIDES for more information when an alert is reported. NIDES must support SO browsing of its archive of audit records, analysis results, and alert reports. NIDES must support SO querying of the archive by user name or user name set, by desired time window, or by a combination of these.

6.5 Suppressing Redundant Alerts

NIDES must avoid displaying unnecessarily redundant alerts by filtering them out before they are reported to the user interface.

6.6 Alert Reporting

The security officer must be able to configure NIDES so that alerts are reported through a pop-up window, e-mail to a specified list of e-mail recipients, or both.

6.7 Attracting Attention

When an alert is reported via a pop-up window, that window must be prominently displayed in the center of the screen. The SO must be required to acknowledge the alert before the window is removed from the screen.

6.8 Levels of Authority

The NIDES security officer user interface must support two levels of SO authorization. A basic level of authorization is required to perform routine monitoring and querying functions. A higher level of authorization is required to use the NIDES customization and test facilities.

Chapter 6

Security Officer User Interface

The NIDES security officer is a user who watches for target system intrusions, confirms reported problems, and takes necessary action to prevent or curtail a discovered intrusion. The SO whose primary concern is the occurrence of target system intrusions must have the information needed to quickly track down unusual or suspicious system behavior once it has been detected.

NIDES must be capable of supporting an SO who has one or more of the following roles and responsibilities:

- Monitors users of target systems for suspicious or unusual activity
- Determines an explanation for a discovered problem
- Modifies the monitoring parameters; this includes configuring the statistical analysis and updating the rulebase

6.1 Ease of Use

The NIDES security officer user interface must be easy to understand and use. A window- and mouse-based graphical user interface must be provided. Context-sensitive help, explaining all of the SO's options, must be available from every NIDES window.

6.2 Timeliness of Alert Reporting

NIDES must report alerts in a timely manner — that is, within seconds after the NIDES analytical components have detected the event.

6.3 Explaining Reported Alerts

When an alarm is raised, NIDES must indicate to the SO the date and time of the alert, the name of the purported user, the machine on which the reported activity originated, and the

methods, policy elements, activities, and behaviors that the NIDES rulebased component must represent are limited to those that can be described using the filled-in fields of the audit record.

5.3 Customization

The NIDES rulebase must be customizable by the security officer.

5.3.1 Rulebase Modification

A security officer must be able to add, delete, or modify rules in the NIDES rulebase. NIDES must be capable of making such modifications to the runtime system. Some amount of experimentation is necessary to determine that the rulebase results in the desired behavior for NIDES.

5.3.2 Rule Activation and Deactivation

The NIDES rulebased component must allow runtime activation and deactivation of predefined rules. Proper use of this feature by the SO prevents rules from uselessly applying their tests to the knowledge base. This can result in large (an order of magnitude) increases in execution speed.

5.4 Compiled Execution

For fastest performance, the rules in the NIDES rulebase must be translated into C language code and compiled into a stand-alone, noninterpreted instantiation of the rulebase.

5.5 Testability

The NIDES rulebased component must be designed so as to accommodate module testing of the rulebase.

Chapter 5

Rulebased Component

One of the requirements of NIDES is to detect a variety of intrusion types. The NIDES rulebased component is designed to allow the security officer to specify the specific events of concern that must be detected. The NIDES rulebased components must also contain a core set of rules designed to detect suspicious activity in monitored UNIX systems.

The NIDES statistical component is based on the premise that abusive behavior will be abnormal behavior. The statistical component thus has no direct concept of what malicious behavior consists of. The NIDES rulebased component, on the other hand, explicitly embodies notions of malicious behavior. The rulebased approach will detect abuse that the statistical approach may miss, and will detect some kinds of abuse more quickly than the statistical approach.

5.1 Detection Security Violations

The NIDES rulebased component must be capable of detecting violations of security policy beyond normal access controls of the organization or target system. The rulebased component must allow an organization to set a security policy that is not or cannot be enforced by normal access controls, either because such enforcement is technically infeasible, or because it is inconvenient. An example of this might be a policy that forbids users from conducting privileged login sessions over the telephone line. It may be desirable to allow such sessions for use by system administrators for emergency purposes, yet forbid such use on an ordinary basis. The rulebased system must alert the SO when such violations of normal policy occur.

5.2 The Rulebase

The NIDES rulebased component must contain a rulebase that is capable of encoding the known intrusion methods for the target system. The rulebase must be capable of encoding a site-specific or organization-specific security policy. It must also be capable of encoding arbitrary activities and behavior patterns considered by the SO to be suspicious. Those

4.7 Detection Accuracy

The statistical component must have at least a 90% detection accuracy for masqueraders and a 0.1% or lower false alarm rate.

4.2 Detecting Departures from Normal Patterns of Behavior

4.2.1 Anomaly Detection

NIDES must compare each user's current activity and historical behavior profile to determine the degree of normality of that user's recent short-term behavior.

4.2.2 Explaining Detected Anomalies

If a user's recent short-term behavior is abnormal with respect to that user's historical profile, NIDES must determine which aspects of the behavior caused it to be considered abnormal.

4.3 Default Profiles

The NIDES statistical component must exhibit acceptable behavior during profile training periods and for new monitored users. Default profiles must be provided for these occasions.

4.4 Intrusion-Detection Measures

NIDES must observe and measure user behavior based on a set of predefined intrusion detection measures. Measures describe aspects of a user's behavior — for example, CPU usage or files read. The anomaly detection algorithms must transform the observed values for these measures into numerical values. These values are stored in the users' profiles.

4.5 Configurability

The NIDES statistical component must be configurable by the SO. Examples of configurable items are the set of active intrusion-detection measures, profile aging factors, and score thresholds. Some amount of system analysis and experimentation is necessary to determine the most appropriate configuration settings for a given target environment.

4.6 Testability

The statistical component must be designed so as to accommodate module testing of the statistical algorithms.

Chapter 4

Statistical Component

One of the requirements of NIDES is to detect a variety of intrusion types, and one type of intrusion is masquerading. A masquerader is an intruder who assumes the identity of an authorized user of the target system. The NIDES statistical component is designed to detect masqueraders through detecting abnormal behavior. The hypothesis is that masqueraders will exhibit behavior patterns detectably different from the normal behavior of the users they claim to be. Abnormal behavior may also be generally indicative of potential misuse on a target system.

4.1 Learning Normal Subject Behavior Patterns

NIDES must maintain historical statistical behavior profiles for users of the monitored systems. A separate historical profile must be maintained for each monitored user. A user's historical profile must reflect the long-term activity of that user. These behavior profiles must be capable of reflecting multimodal user behavior distributions.

4.1.1 Profile Updating

The historical profiles must be updated daily (or according to a schedule specified by a security officer) based on user behavior observed since the last profile update. Regular profile updating will allow the historical profiles to adapt to changes in user behavior and to remain accurate descriptions of the user's expected behavior.

4.1.2 Profile Aging

When a historical profile is updated, the information in the old profile must be discounted, or "aged," to give greater weight to the new information, and thus to the more recent user behavior. The discount factor, or profile aging factor, must be specifiable by the SO. This will allow the SO to specify the size of the moving time window represented in the historical profile.

set as test data to be run against the trial measures, rules, and parameter settings. Such experiments can be run concurrently on the same machine that is supporting the "live" intrusion-detection system. After performing experiments, if the SO should decide that the new measures, rules, or parameter settings are desired in the running version of the intrusion-detection system, the SO can then so indicate, and the changes will become effective. The test facility produces reports of test runs so that the SO can compare the results of different candidate configurations for NIDES. Because the SO who makes customizations to NIDES must have specialized knowledge and skill, additional authorizations will be necessary for the SO who uses its customization and test facilities.

alert report identifies the top five statistical intrusion-detection measures that contributed to the assessment that the recently observed behavior of the subject is unusual. For alarms raised by the NIDES rulebased component, the alert report contains a brief explanation of the rule that triggered the alert as well as the data that caused the rule to be triggered. To allow the SO to review current and past events, NIDES includes an archiver that stores audit data, analysis results, and alerts. The SO can browse this archived data to investigate reported suspicious events.

3.5 Target System Independence

A *realm* is a group of monitored machines that use the same auditing mechanism to produce the target system's audit trails. The NIDES *realm interface* allows for the communication of this audit data to the NIDES intrusion-detection components. One portion of this realm interface resides within the target system (the *realm client*), and the other portion resides on the NIDES processor (the *realm server*). Because various realm types have their own audit trail formats, a different realm client is required for each realm type. Using a simple data transfer protocol, the realm client and server establish a connection whereby the target system's audit information is converted into a generic NIDES audit record format to be used by the NIDES intrusion-detection components, and then transported from the target system to the intrusion-detection system.

This allows NIDES to be target-system independent. That is, the NIDES intrusion-detection components need not be altered or rewritten in any way when a new target system type is to be monitored. All that is required is the addition of a realm client that converts the native audit data into the NIDES generic format.

The NIDES customization facility also contributes to the target-system independence of NIDES. Because this facility allows the NIDES user to customize the rulebase, select the appropriate statistical intrusion-detection measures, and adjust the statistical parameters, NIDES can be applied to a new target system type with no recoding other than the addition of a new realm client.

3.6 Customization

The NIDES customization facility allows the SO to tune and to make customizations to NIDES. This facility provides the SO with the ability to make selective use of the NIDES intrusion-detection components. It also provides the SO with the capability of selectively turning on and off specific intrusion-detection measures within the statistical component. It also provides the SO the capability of adding, dropping, or modifying rules in the rulebased component and of changing the parameter settings of the statistical component.

The NIDES customization facility allows the SO to experiment with new measures, rules, and parameter settings before making a change effective in the running intrusion-detection system. The SO can select an audit data set from the archived audit data and use that data

for monitored users. The historical profile is a description of a user's normal (i.e., expected) behavior with respect to a set of intrusion-detection measures.

The rulebased component contains a rulebase that describes suspicious behavior. The rules are constructed based on knowledge of past intrusions, known system vulnerabilities, and the installation-specific security policy. The rules describe suspicious behavior that is independent of whether a user is deviating from past behavior patterns. The rules can be used to encode information about known system vulnerabilities and reported attack scenarios, as well as intuition about suspicious behavior. Input audit data are matched against these rules to determine whether the observed behavior is suspicious. The current rulebase is customized for use in a UNIX environment.

3.2 Detection Accuracy

The NIDES statistical component is designed to have a nominal false alarm rate of 0.1%. This nominal rate can be modified by the SO. The NIDES rulebased component contains a rulebase that is customizable by the SO. Its detection accuracy is highly dependent on the particular customizations made. NIDES includes a test facility so that the SO can experiment with candidate customizations of the NIDES statistical component and rulebase to assure that the detection accuracy is sufficiently great, before making the candidate customizations effective.

Detection accuracy is enhanced by the NIDES resolver, which filters redundant alarms before reporting them to the SO.

3.3 Real-time Detection

The capability for NIDES to operate in real time, that is, to detect suspicious events within minutes of their having occurred, is provided in part by several of the NIDES components. This requirement has been addressed by engineering the NIDES software so that throughput is maximized and performance overhead is minimized. To this end, the statistical and rulebased components as well as the supporting NIDES infrastructure have been designed for high performance. For example, the rulebased component allows the rule base to be compiled, with the result that execution is very fast. In addition, the statistical algorithms have been designed so that very little processing time is required for each audit record.

3.4 Display and Interpretation of Results

NIDES includes a graphical user interface that displays alerts on its screen. It also allows the specification of e-mail recipients of alerts. NIDES avoids displaying unnecessarily redundant information through the use of the resolver, which filters out redundant alarms before they are reported to the user interface. The NIDES graphical user interface also provides explanations for the alarms that are reported. For alarms raised by the NIDES statistical component, the

Chapter 3

NIDES Components

We have structured NIDES into components intended to satisfy the basic requirements discussed in the previous chapter:

- Intrusion-detection components
  - Statistical component
  - Rulebased component
- Results display/interpretation
  - Graphical user interface
  - Resolver
  - Archiver
    - for audit data
    - for analysis results
    - for alert data
- Target system realm interface
- Customization and test facility

Below we discuss how these components of NIDES contribute to satisfying the NIDES software requirements discussed in Chapter 2. In the following chapters, we discuss in turn the software requirements for each of the individual NIDES components.

3.1 Detection of Intrusions

NIDES includes two intrusion-detection components: the statistical component and the rule-based component. Both components accept input data from the realm interface (discussed in Section 3.5). Their function is to detect a variety of intrusion types, including external penetrators and malicious insiders. The statistical component maintains *historical profiles*

users use the target system relatively uniformly during the business day and are prohibited access during off hours, measures relating to the time of day may be less useful. Thus, NIDES must provide the capability of customizing its statistical component.

2.6.2 Rulebase Customization

NIDES must provide the capability of customizing its rulebase. The rulebase encodes the known intrusion methods for the target system, and these intrusion methods take advantage of the known target system vulnerabilities. These vulnerabilities vary for different target system types; that is, UNIX vulnerabilities are different from MVS vulnerabilities, and thus intrusion methods for UNIX systems are different from intrusion methods for MVS systems. NIDES includes a standard rulebase for use in UNIX systems. In addition, the rulebase can encode the security policy for a given user community. Thus, even for the same target system type, portions of the rulebase will be unique to the given community, because different security policies are enforced. In addition, an SO may want to add to or modify the rules as NIDES is used over time, based on actual experience with intrusion attempts. Thus, NIDES must allow the SO to customize its rulebase.

2.6.3 Test Facility

Because the customizations to the NIDES rulebase and statistical component can greatly alter NIDES's detection ability in ways perhaps unanticipated by the SO, NIDES must also provide a test facility to allow the SO to experiment with candidate NIDES configurations before making them effective. The facility must allow a candidate configuration to be tested using test audit data sets, defined by the SO from the audit data archive. The test facility must produce reports of test runs so that the SO can compare the results of different candidate configurations for NIDES.

2.6.4 Special Authorization

Because some specialized knowledge and skill will be required of the SO who makes customizations to NIDES, NIDES must require additional authorizations for an SO to use its customization and test facilities.

2.3 Real-time Detection

NIDES operates in real time, and must be capable of detecting an intrusion or intrusion attempt within minutes of its having occurred. This allows a security officer to take timely action, such as terminating an active session or telephoning to verify identity. Timely action both discourages and frustrates the intruder, and prevents unnecessary damage from occurring.

2.4 Display and Interpretation of Results

NIDES must clearly display its findings to the SO. In addition, it must avoid confusing the SO by displaying too much information or unnecessarily redundant information. Thus, NIDES must be capable of filtering its results so that it can screen out redundant alarms before they are reported. NIDES must also provide the SO with an explanation for why it raised an alarm. In addition, NIDES must be capable of allowing review of current and past events.

2.5 Target System Independence

A major goal of NIDES is the monitoring of a wide variety of target system types. Thus, NIDES must be target-system independent. This means that NIDES should require only very localized and relatively small alterations before working with a new target system type. This allows NIDES to be easily assimilated into new environments. It also enables NIDES to simultaneously monitor many different target system types.

2.6 Customization Capability

Because each user community is unique in how the behavior of its users can be characterized, and because each target system type provides different information in its audit data, it must be possible to customize NIDES and to tune its components for best performance in each new environment.

2.6.1 Statistical Component Configuration

Certain measures may not be relevant to certain target system types; for example, measures related to window command usage are not relevant to target systems that do not support windowing. In addition, certain measures may not be relevant because audit data relating to them are not collected; for example, measures related to window command usage are not relevant to most UNIX workstation target system types because information relevant to window command usage is not captured in the audit data. In addition, for some user communities, certain measures may not be relevant; for example, if in a given community

Chapter 2

Basic Requirements

Our overriding goal is for NIDES to find wide use in sensitive applications in the federal government and in private companies working in partnership with federal agencies. To reach this goal, NIDES must meet the following basic requirements:

- Detection, with confidence, of a variety of intrusion types
- Real-time detection (within minutes)
- Display and interpretation of current and past results
- Target-system independence
- Customization for use in new and changing environments

We discuss each of these below.

2.1 Intrusion Types

The goal of NIDES is to detect both successful and attempted intrusions. The more types of intrusion that NIDES is capable of detecting, the more useful it will be, and the more it can be relied on to alert security officers. Thus, NIDES must detect suspicious activity by authorized users as well as penetration attempts by outsiders.

2.2 Detection Accuracy

For NIDES to reliably protect a target system, it must have a very high detection rate of the intrusions and intrusion attempts on that system. It is equally as important that NIDES not claim that events are suspicious when, in fact, they are not. In other words, NIDES must have a relatively low false alarm rate, so that a security officer (SO) always gives due attention to the authentic events of serious concern that NIDES discovers.

Chapter 1

Introduction

This Software Requirements Specification for NIDES was developed under U.S. Government contract number N00039-92-C-0015, and the work it describes is funded by the U.S. Navy SPAWAR. The document is deliverable A001, Software Requirements Specification.

1.1 Scope

The Specification defines the software requirements for the Next-Generation Intrusion Detection Expert System (NIDES). Its intent is to enable the government to assess whether the completed NIDES complies with the requirements stated here.

1.2 Purpose

NIDES is a comprehensive intrusion-detection system that performs real-time monitoring of user activity on one or more target system computers and detects unusual and suspicious user behavior on those target systems.

1.3 Document Overview

This specification is organized as follows. Chapter 2 discusses the basic requirements for NIDES. Chapter 3 describes the components of NIDES that have been introduced to satisfy those requirements. Chapter 4 discusses the software requirements for the NIDES statistical component. Chapter 5 discusses the software requirements for the NIDES expert system component. Chapter 6 describes the user interface requirements. Chapter 7 describes the requirements for the NIDES resolver. Chapter 8 describes the requirements for the NIDES archiver and data management facility. Chapter 9 describes the requirements for NIDES to interface to target (monitored) systems as well as the audit data collection requirements for NIDES. Chapter 10 describes the requirements for the NIDES customization and test facility. Chapter 11 provides a summary.

11 Summary **27**

4.4 Intrusion-Detection Measures 12
4.5 Configurability 12
4.6 Testability 12
4.7 Detection Accuracy 13

5 Rulebased Component **15**
5.1 Detection Security Violations 15
5.2 The Rulebase 15
5.3 Customization 16
5.3.1 Rulebase Modification 16
5.3.2 Rule Activation and Deactivation 16
5.4 Compiled Execution 16
5.5 Testability 16

6 Security Officer User Interface **17**
6.1 Ease of Use 17
6.2 Timeliness of Alert Reporting 17
6.3 Explaining Reported Alerts 17
6.4 Querying for More Information 18
6.5 Suppressing Redundant Alerts 18
6.6 Alert Reporting 18
6.7 Attracting Attention 18
6.8 Levels of Authority 18

7 Resolver **19**

8 Data Management Facility and Archiver **21**
8.1 Persistent and Efficient Storage of Archived Data 21
8.2 Querying and Retrieval of Archived Data 21

9 Target System Realm Interface **23**
9.1 Generic Audit-Record Format 23
9.2 Audit Data Transfer 23
9.3 Filtering 23
9.4 Robustness 23
9.5 Dynamic Addition and Deletion of Target Systems 24
9.6 Sources of Audit Data 24

10 Customization **25**
10.1 Runtime Configuration of Statistical Component 25
10.2 Runtime Configuration of Rulebased Component 25
10.3 Runtime System Monitoring and Control 25
10.4 The NIDES Test Facility 26

Contents

1 Introduction **1**
- 1.1 Scope . . . 1
- 1.2 Purpose . . . 1
- 1.3 Document Overview . . . 1

2 Basic Requirements **3**
- 2.1 Intrusion Types . . . 3
- 2.2 Detection Accuracy . . . 3
- 2.3 Real-time Detection . . . 4
- 2.4 Display and Interpretation of Results . . . 4
- 2.5 Target System Independence . . . 4
- 2.6 Customization Capability . . . 4
  - 2.6.1 Statistical Component Configuration . . . 4
  - 2.6.2 Rulebase Customization . . . 5
  - 2.6.3 Test Facility . . . 5
  - 2.6.4 Special Authorization . . . 5

3 NIDES Components **7**
- 3.1 Detection of Intrusions . . . 7
- 3.2 Detection Accuracy . . . 8
- 3.3 Real-time Detection . . . 8
- 3.4 Display and Interpretation of Results . . . 8
- 3.5 Target System Independence . . . 9
- 3.6 Customization . . . 9

4 Statistical Component **11**
- 4.1 Learning Normal Subject Behavior Patterns . . . 11
  - 4.1.1 Profile Updating . . . 11
  - 4.1.2 Profile Aging . . . 11
- 4.2 Detecting Departures from Normal Patterns of Behavior . . . 12
  - 4.2.1 Anomaly Detection . . . 12
  - 4.2.2 Explaining Detected Anomalies . . . 12
- 4.3 Default Profiles . . . 12

Software Requirements Specification: Next-Generation Intrusion Detection Expert System

March, 1993

Teresa F. Lunt and Debra Anderson
Computer Science Laboratory
SRI International
Menlo Park, California 94025