

SRI International

NIDES Training Course • August 1994

Next Generation Intrusion Detection Expert System (NIDES)

Training Course — Beta Release

Presented by:

Debra Anderson, Computer Science Laboratory

Thane J. Frivold, System Technology Division

Alfonso Valdes, Applied Electromagnetics and Optics Laboratory

Contents

Day 1 — Viewgraphs

Day 2 — Viewgraphs

Day 3 — Viewgraphs

Day 4 — Viewgraphs

Worksheets

Glossary

Day 1 Viewgraphs

Day 1 — Agenda

| Course overview

- Intrusion Detection
- NIDES history and overview
- Real-time NIDES operation (discussion)
- Nonanalysis configuration options
- Real-time NIDES operation (hands-on)

Course Overview

Course Overview — Day 1 (Morning)

- Intrusion Detection
- NIDES history and system overview
- General terms and concepts used in NIDES
- NIDES processes and data flow
- NIDES processing modes (real-time and batch)
- Overview of analysis components
- Audit data sources
- NIDES system configuration
(audit data, target hosts, NIDES host)
- NIDES utility programs

Course Overview — Day 1 (Afternoon)

- NIDES real-time processing (discussion)
- Nonanalysis configuration (discussion)
- Real-time NIDES operation (hands-on)
 - Analysis activation
 - Alert configuration and filters
 - Target host activation
 - Archiver
 - Receiving alerts
 - Status reporting
 - Browsing result and audit data
 - Nonanalysis configuration

Course Overview — Day 2 (Morning)

- Overview of configuration options
 - Statistics
 - Rulebase
- Configuration application

Course Overview — Day 2 (Afternoon)

- Rulebase configuration (discussion)
 - Rulebase terms and concepts
 - Rulebase execution
 - Rule Syntax
 - rb_config file
 - Default rulebase
 - Writing and installing rules
 - Rulebase design
 - Design rb_config file (exercise)
 - Design and write rules (exercise)

Course Overview — Day 3 (Morning)

- Rulebase configuration (hands-on)
 - Configuration of rb_config file defaults
 - rb_config file **GENERIC_CONFIG** section
 - Rule writing
 - Rule compiling and installation
 - Rule activation/deactivation
- Statistics configuration (discussion)
 - Statistics configuration options
 - Statistics configuration application

Course Overview — Day 3 (Afternoon)

- Statistics configuration (hands-on)
 - Measures
 - Parameters
 - Classes
 - Profile updating
 - Performance considerations
- NIDES test facility (discussion)

Course Overview — Day 4 (Morning)

- NIDES test facility (hands-on)
 - Audit data sets
 - Instance management
 - Test configuration
 - Test initiation
 - Test status reporting
 - Test result viewing
 - Profile viewing

Course Overview — Day 4 (Afternoon)

- NIDES utility programs (hands-on)
- NIDES upcoming events
- Questions and answers

Intrusion Detection

The Threat to Computer Security

- External penetrators can invade privacy or cause damage
- Unscrupulous insiders can invade privacy or cause damage
- Flawed access controls and other holes can result in accidental disclosure of sensitive information or damage to valuable information assets
- Even secure systems can be violated if procedural safeguards are not observed (e.g., if users write down their passwords)

Security Goals

- Protect privacy of users
- Protect security of confidential information
- Protect integrity of important data and assets

Why Audit?

- User accountability
- Deterrent value
- Detect security problems
- Gather evidence to build a case

The Need for Audit Trail Analysis

- Large volume of data
- Relevant data may not be collected
- Much irrelevant data is collected
- Records must be examined in context
- Analysis tools are needed

Types of Audit Trail Analysis

- Offline, after-the-fact, analysis of audit data
- Real-time testing of audit data to allow an immediate response
- Subsequent analysis of audit data for damage assessment

Intrusion-Detection System Goals

- Detect a wide variety of intrusion types
- High believability in findings
- Real-time detection (within minutes)
- Display and interpretation of current and past results
- Ease of use
- Easy adaptation to diverse computing environments

Types of Threats

- External penetrators
- Internal penetrators, including
 - Masqueraders
 - Clandestine users (who evade auditing and access controls)
- Misfeasors (who misuse their privileges)

Possibilities

- External penetrators: failed logins
- Internal penetrators: failed access attempts
- Masqueraders: departures from established patterns of use

Possibilities Continued

- Misfeasors:
 - *A priori* rules for socially acceptable behavior
 - Comparison with norm established for the class of user
- Clandestine users:
 - Disabling of auditing
 - Departures from established system-wide norms for the facility

Statistical Approach

- Establish a historical behavior profile for each user
- Compare current behavior with the profiles
- Detect departures from established norms
- Update profiles to adapt to changes in user behavior

Example: NIDES Statistical Component

- Identifies anomalous behavior
- Collects statistics on about 50 intrusion-detection measures
 - *Continuous measures, e.g., CPU usage*
 - *Categorical measures, e.g., files used*
- A *historical profile* contains statistics relevant to the measures for each user's observed historical behavior
- A *short-term profile* contains statistics relevant to the measures for each user's recently observed behavior

Example: NIDES Statistical Component Continued

- Continuously evaluates current activity against the profiles
- Raises an alarm when current activity deviates significantly from the profiles
- Updates historical profiles daily
- Ages older data during profile update

Difficulties with the Statistical Approach

- Some users have erratic behavior — masqueraders can go undetected
- For misfeasers, abuse is “normal”
- Vulnerable to defeat by
 - Slowly moving to a new norm
 - Slowly increasing “normal” range
- **Possible Solutions**
 - Default profiles
 - Group profiles
 - Trend tests
 - Rulebased prohibitions

Rule-Based Approach

- Develop a rulebase to encode
 - Known intrusion methods
 - Known system vulnerabilities
 - Suspected “bad” actions
 - Security policy
- Example: > 3 login failures in one second
- Limitation: can detect only known vulnerabilities and attacks
- Variation: define “acceptable” behavior

Separate Machine for Analysis

- Least performance impact on monitored system
- More tamper-resistant
- Monitors several machines at once
- Can be system-independent

Privacy Issues

- Potential for invasion of privacy
- Potential for abuse of personal data
- Obtain informed consent of users

What Level of Data to Audit?

- OS system calls
- OS command line
- DBMS operation invoked
- DBMS data affected
- Within applications
- All keystrokes

Audit Data Considerations

Tradeoffs in:

- Types of intrusions that can be detected
- Complexity and volume of data
- Ability to appeal to intuition when anomaly is detected
- Ability to formulate rules that characterize intrusions
- Ability to “play back” an anomalous session
- Ability to perform damage assessment or gather evidence

Need to combine different types of audit data

Examples of Events Monitored

- Login
- Logout
- Program execution
- Commands used
- System calls
- Directory modification
- Password-protected directory access
- Session location change
- Network activity

Typical Audit Record Fields

- Subject
identifies user, session, and location
- Action
the action attempted
- Object
*what the subject acted upon;
subfields depend on type of action*
- Errorcode
- Resource info
CPU, memory, I/O
- Timestamp

NIDES

- Statistical anomaly detection
- Flexible rule-based detection
- Resolver to filter redundant alarms
- Generic audit record format that facilitates use in new environments
- Graphical user interface
- User-modifiable rulebase
- User-configurable rulebase and statistics
- User-specifiable alert reporting

NIDES Continued

- Context-sensitive online help facility
- Simultaneous monitoring of numerous (possibly heterogeneous) machines
- Real-time operation to detect unusual activity as it occurs
- System monitoring facility
 - Information on target hosts
 - Audit data archiver status
 - Hourly summary of system throughputs
 - Hourly summary of alert generation

NIDES Continued

- Archive of audit records, analysis results, and alerts
- Browsing of audit data and analysis results
- Test facility
 - Flexible creation of test data sets from the audit record archive
 - Configuration of candidate rulebase and statistical parameters
 - Tests can run concurrently with the live NIDES
 - Test result archival for comparison

NIDES History & Overview

History

- IDES Prototypes
 - Initial studies performed at SRI in 1980s
 - Several prototypes developed late 1980s to present
- NIDES Prototypes
 - 1992 IDES prototype re-engineered becomes NIDES
 - NIDES Alpha version released 2/93
 - NIDES Alpha-patch version released 9/93
 - NIDES Beta version released 5/94 (available for evaluation)
 - Beta version update 4th QTR 1994

New Features In Beta Release

- Customization of rulebase and statistics (real-time and batch)
- Performance tuning functions
- Privileged user functions
- Analysis result archive
- Audit data archive
- Enhanced system monitoring
- Subject profile review
- Alert filters

Components

- Audit data generation (agend, agen)
- Audit data collection (arpool)
- Analysis (rulebased and statistical)
- Resolver
- User Interface
- Persistent storage facility
- RPC infrastructure (agents, nameserver)

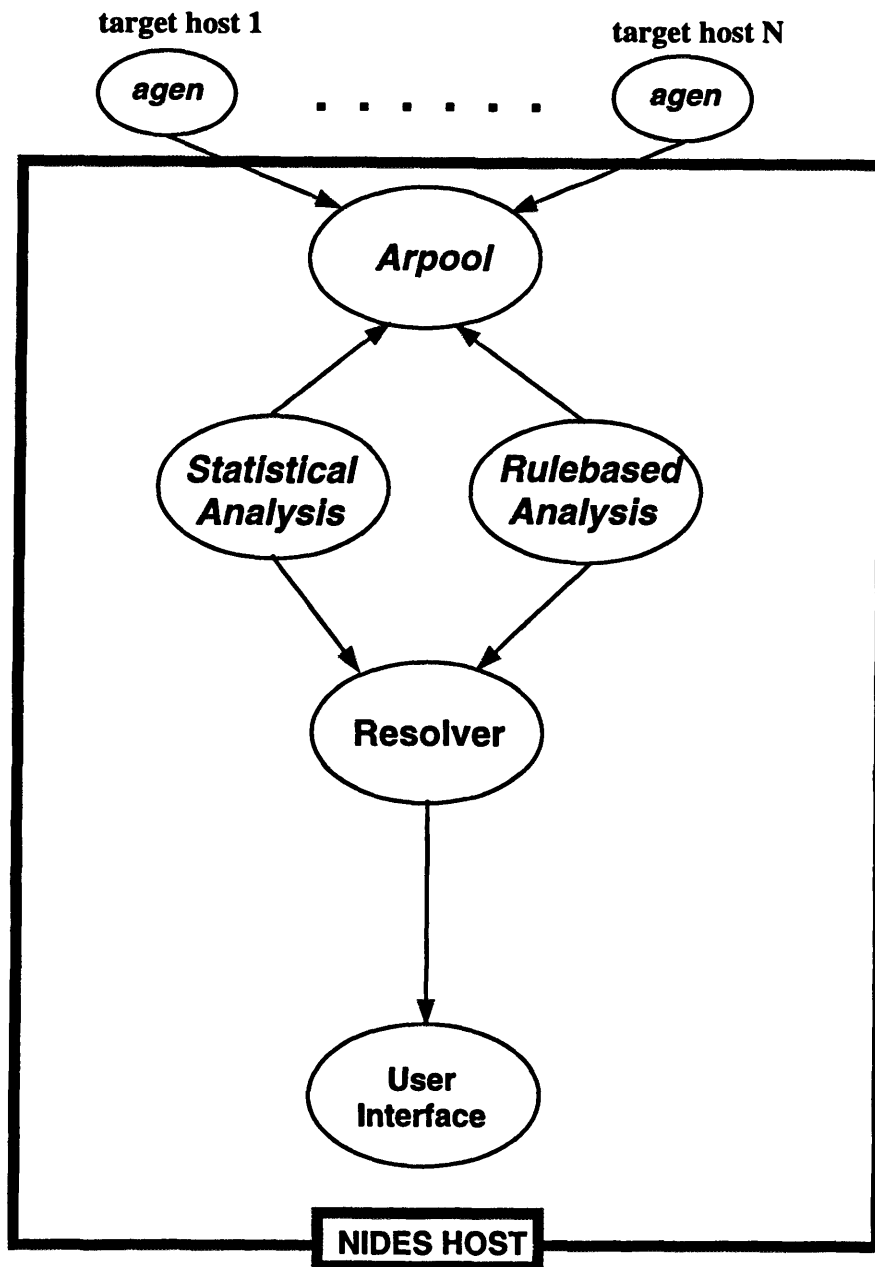
General Terms & Concepts

- Target host
- NIDES host
- Native format audit record
- NIDES audit record
- Alert (anomaly)
- Real-time analysis
- Batch analysis
- Instance
- Profile
- Glossary contains more terms

Processing Mode (Real-time)

- As audit data is generated on a target host it is converted from the target host's native format to NIDES format and transferred to the NIDES host
- Audit data from multiple target hosts is coalesced into a single stream of NIDES audit records and provided to the NIDES analysis components
- NIDES analysis components results are resolved and provided to the user via the NIDES user interface
- NIDES real-time processing is performed by many NIDES processes

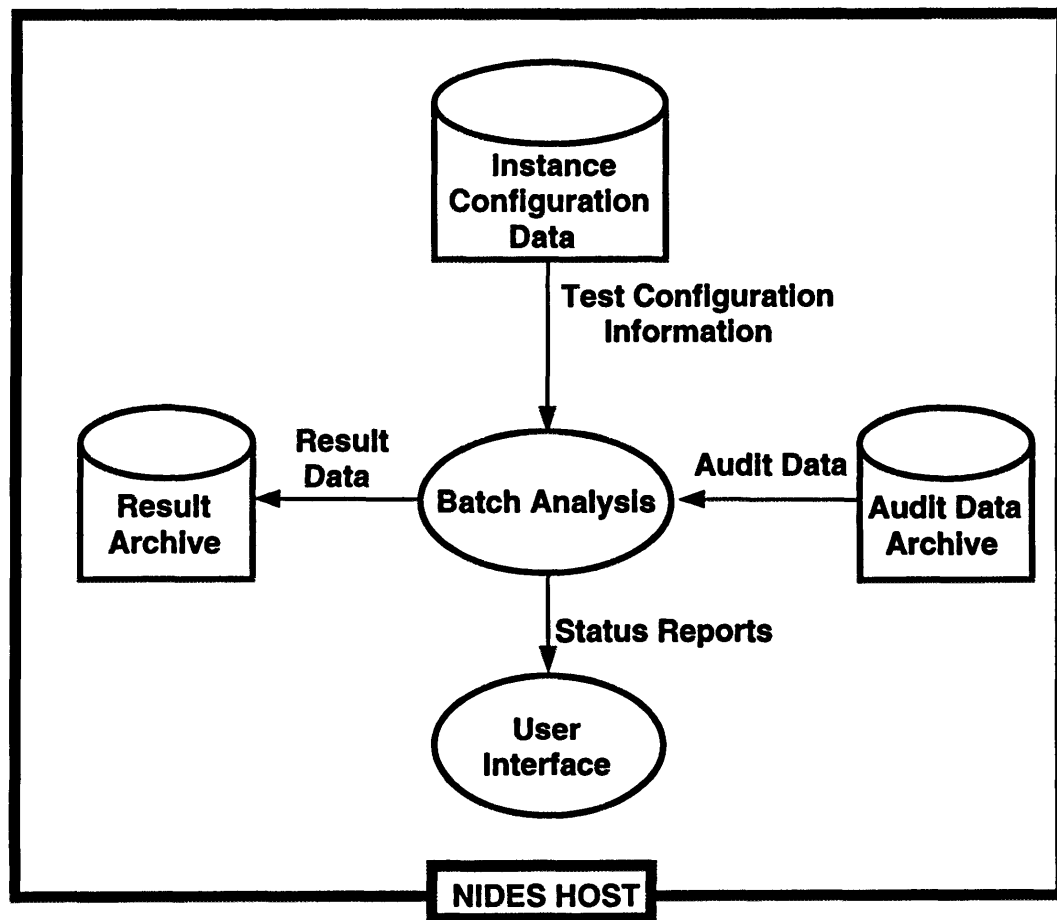
Processes & Data Flow (Real-time)



Processing Mode (Batch)

- A test instance is created and configured
- A NIDES audit data set is constructed from NIDES audit record files
- A batch NIDES run is started
- Results are archived for user review when the batch run completes
- NIDES batch processing is performed by a single monolithic NIDES process

Processes & Data Flow (Batch)



Rulebased Analysis Component

- Rules
 - Two parts (tests and actions)
 - Information used by rules stored in a factbase
 - Rulebase written using SRI-developed rule language
 - Priorities can be assigned to rules
 - Multiple rules can perform as a group

Rulebased Analysis Component Continued

- Rulebase Configuration File (rb_config)
 - Supports rule configuration at runtime
 - rb_config file processed at startup
 - Users customize rulebased analysis without modifying the rulebase
 - Supports user-defined configurations
- Factbase
 - Facts may be asserted or deleted by any NIDES rule
 - Allows rules to store information for later analysis or use
 - Supports rulebase case building

Rulebase Audit Record Processing

- Audit record fact asserted into factbase
- Audit record analyzed by rulebase
- Audit record fact deleted from factbase
- Result reported to resolver
- Process repeated with next audit record

Statistical Analysis

- Compares subject's short-term behavior and long-term behavior
 - | Reports an alarm if difference exceeds a threshold
 - | Subject behavior represented with measures
- Generates and maintains profiles of long and short-term behavior for each subject represented in the audit trail
- Subject's long-term behavior is learned in three training phases

Statistical Analysis Terms

- Subject (traditionally a computer user)
- Profile
(short-term/current and long-term/historical)
- Half-life (short-term and long-term)
- Measure (categorical and continuous)
- Category (general and class list)
(each measure has a category distribution)
- False-positive
- True-positive
- Detection rate

Statistical Analysis Terms Continued

- Cross-profiling
- Red/critical threshold
- Yellow/warning threshold
- Q (each measure has a Q distribution)
- S (each measure has an S distribution)
- T2 (each profile has a T2 distribution)
- Training
- Minimum effective-N

Statistical Analysis Audit Record Processing

- Read an audit record
- Construct an activity vector
- Adjust category counts
- Calculate score
- Compare score to thresholds to determine level (Safe, Warning, or Critical)
- Report result to resolver

Statistical Analysis Score Calculation

- Category determination
- Q calculation
- S calculation
- T2 calculation

Statistical Analysis Profile Building

- Each profile goes through C, Q and T2 training phases (length of each phase determined by training period)
- C training calculates measure category distributions
- Q training calculates a measure of deviation of short-term behavior about the distribution of long-term behavior
- T2 training calculates Q distributions and subject thresholds

Statistical Analysis Profile Building Continued

- Each profile is updated daily or when user requests
- During updating category counts are updated and rarely seen categories are dropped
- Anomalies are NOT reported until at least one measure is trained

Audit Data Sources

- SunOS C2
- SunOS BSM
- UNIX accounting
- Prior NIDES client customizations
 - IBM mainframe database application logs
 - Trusted Xenix
- Straightforward adaptation to other data sources

Audit Data Descriptions — C2

- Older Sun auditing package
 - 18 audit flags
(dr, da, dc, dw, lo, ad, p0, p1)
- Audit flags have four states:
 - OFF (record NO events)
 - ON (record all events)
 - Record failed events ONLY
 - Record successful events ONLY

Audit Data Descriptions — BSM

| BSM stands for 'Basic Security Module'

- Current SunOS audit package
- Versions for SunOS 4.X and Solaris
- NIDES currently supports BSM version 1 (SunOS 4.X)

Audit Data Descriptions — BSM Continued

- 12 Audit flags (BSM version 1)
(dr, da, dc, dw, lo, ad, p0, p1, ex, nt, io, other)
- Audit flags have four states:
 - OFF (record NO events)
 - ON (record all events)
 - Record failed events ONLY
 - Record successful events ONLY

Audit Data Descriptions

UNIX Accounting

- Standard UNIX accounting log
- Initially developed to track user's resource usage for billing purposes
- Higher-level data than C2 or BSM (less verbose)
- Records resource utilization values for each program execution

System Configuration Audit Data

- Minimal configuration uses UNIX accounting
- SunOS C2 or BSM recommended
- Configure all C2/BSM flags ON except data reads (dr)
- All target hosts do NOT need to run the same auditing system

System Configuration

NIDES Host

- Installation of NIDES software
- Creation of “ides” account and group
- Set NIDES environment variables
IDES-ROOT and IPC_NAMESERVER
- Execution of ipc-nameserver as
continuous background process
- X11R5 and “twm” window manager
recommended for NIDES interface
- Initialization of privileged user list

System Configuration

Target Hosts

- Installation of agend and agen programs
- agend runs continuously as a daemon process
- Include startup of agend in each target host's rc.local file

Utility Programs

- acc2ia
 - Converts UNIX accounting files to NIDES audit data files
- audit2ia
 - Converts SunOS C2 or BSM audit files to NIDES audit data files
- adset_index
 - Creates an index file for a NIDES audit data file
 - Audit data files processed by adset_index become audit data sets
 - NIDES tests use audit data sets

Utility Programs Continued

- agen
 - Collects target host native audit data
 - Converts the native audit data to NIDES audit data
 - Transfers the NIDES audit data to the arpool process
 - Beta version handles SunOS C2 or BSM version 1 data and UNIX accounting data
 - Started by agend process through NIDES UI request

Utility Programs Continued

- agend
 - Daemon process that should run continuously on all potential NIDES target hosts
 - Activates and deactivates agen processes
 - Requests to activate or deactivate agen are generated by the NIDES UI
 - Include startup of agend in rc.local file of every potential NIDES target host
- apstat
 - Prints statistics on arpool data flow

Utility Programs Continued

- archiver
 - Converts NIDES audit data into a NIDES audit data archive
 - Runs in two modes: real-time and batch
 - NIDES audit data browse functions use audit data archives
 - Audit data sets are created from NIDES audit data archives

Utility Programs Continued

- arpool (Audit Record Pool)
 - Collects audit data from all active agents
 - Provides audit data to all audit data consumers (analysis and archiver)
 - Started via the NIDES UI
- batch-analysis
 - Runs NIDES analysis using NIDES audit data sets and test instances
- iamerge
 - Merges two NIDES audit data files into one file

Utility Programs Continued

- iapr
 - Prints an ASCII representation of NIDES audit data
 - Runs in two modes: real-time and batch
 - Can be used to monitor record flow through arpool
- init_priv_user_list
 - Configures the NIDES privileged user list
- init_stat_config
 - Creates a binary format statistics configuration file from an ASCII text file

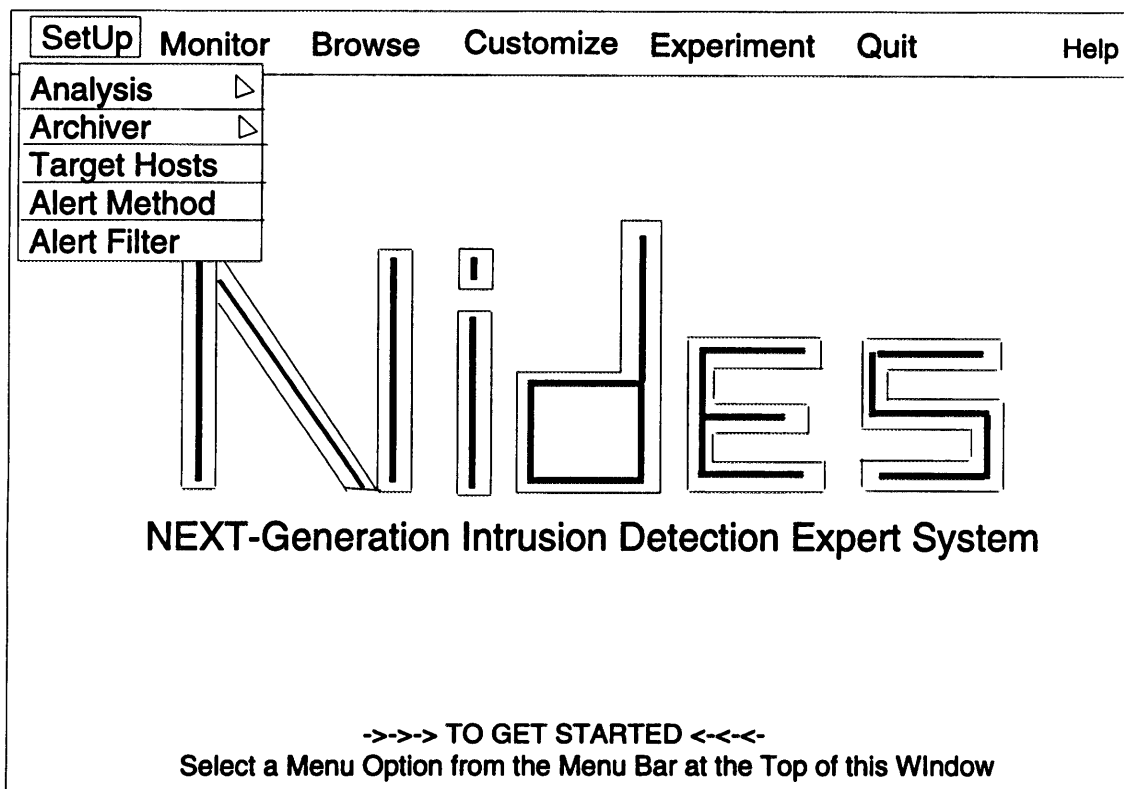
Utility Programs Continued

- ipc_nameserver
 - Provides RPC client/server lookup services for all NIDES host processes
 - Must be running for NIDES to work

NIDES Real-time Operation (Discussion)

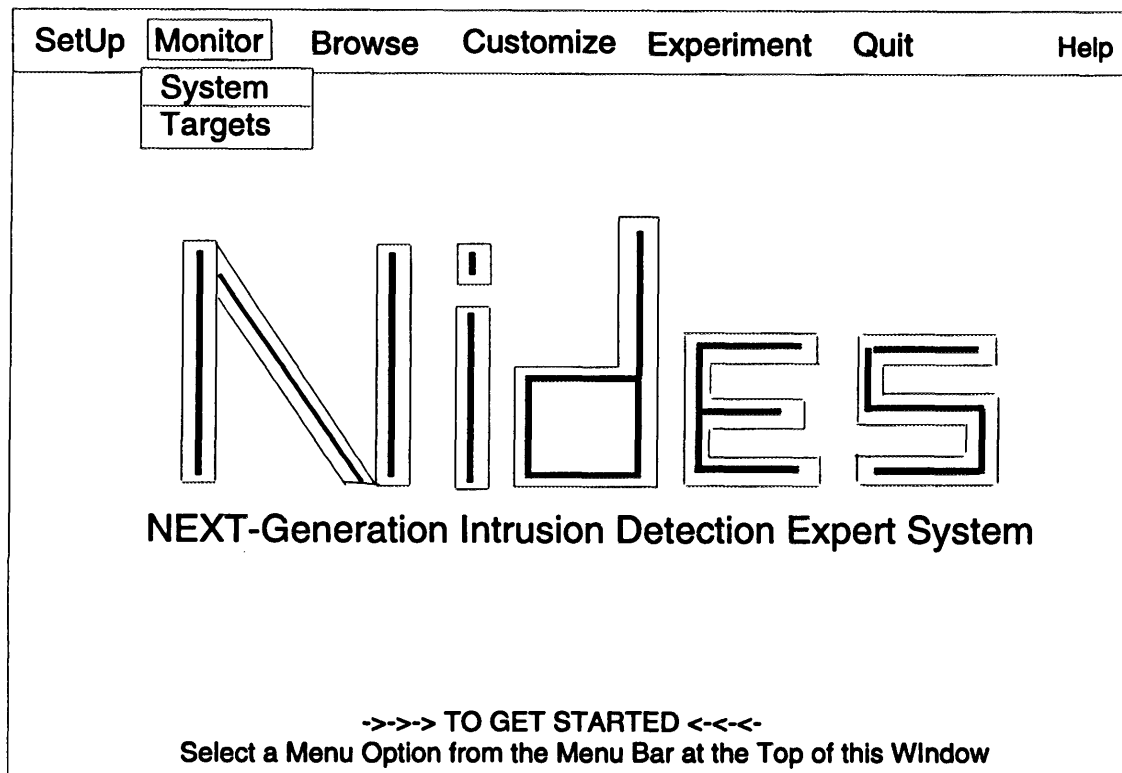
Real-Time Processing Setup Menu

- Supports basic real-time processing functions



Real-Time Processing Monitor Menu

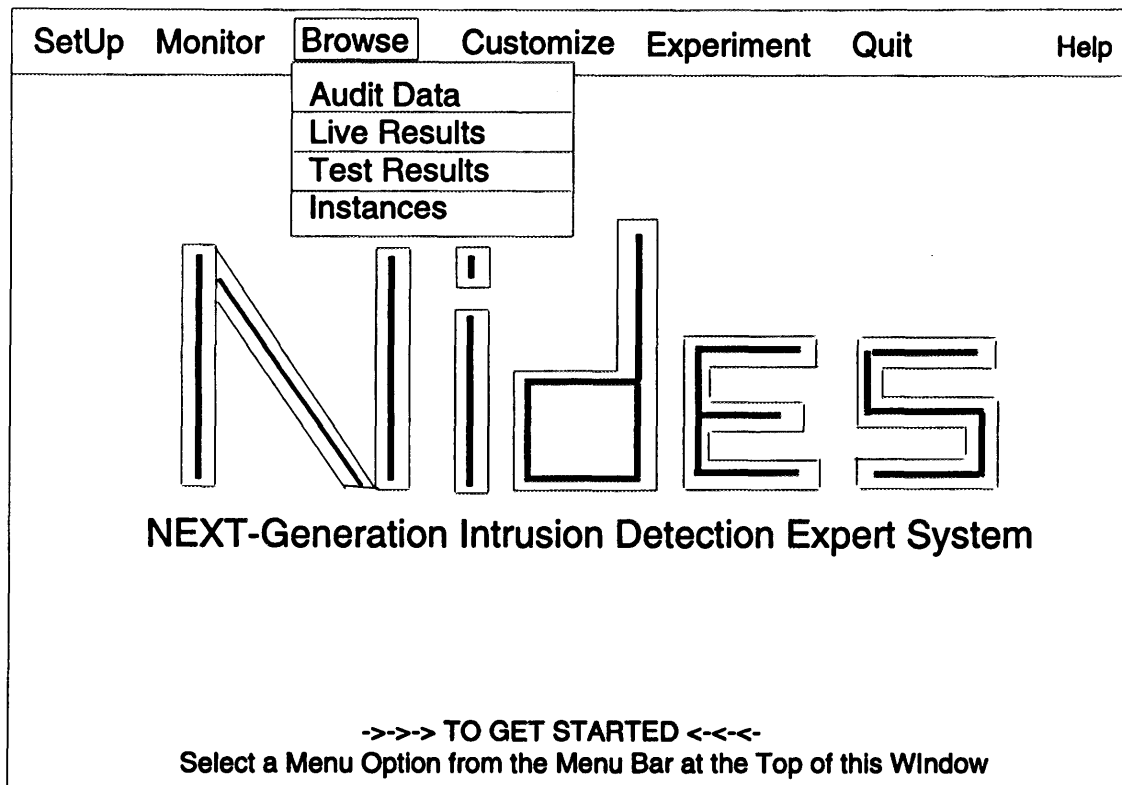
- Provides status of real-time processing



Real-Time Processing Browse Menu

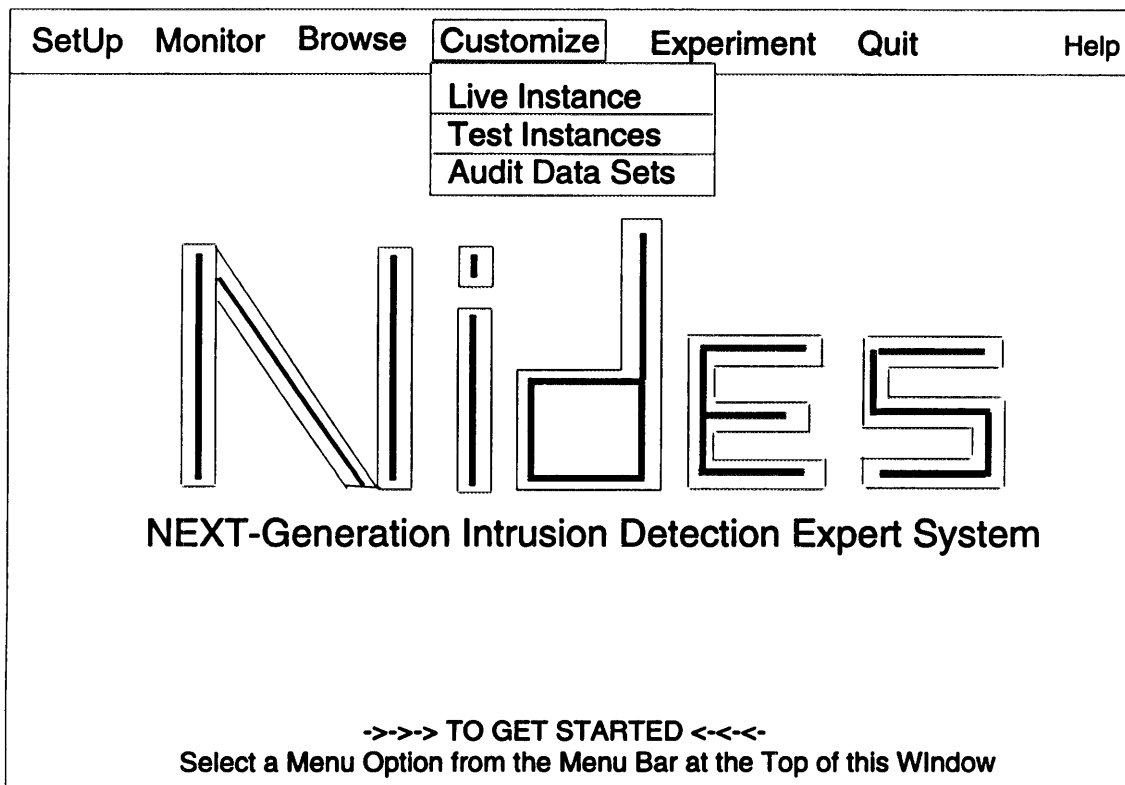
- Supports review of
 - Audit data archives
 - Analysis result data (real-time and batch)
 - Instances

Browse Menu



Real-Time Processing Customize Menu

- Supports analysis configuration functions (real-time and batch)



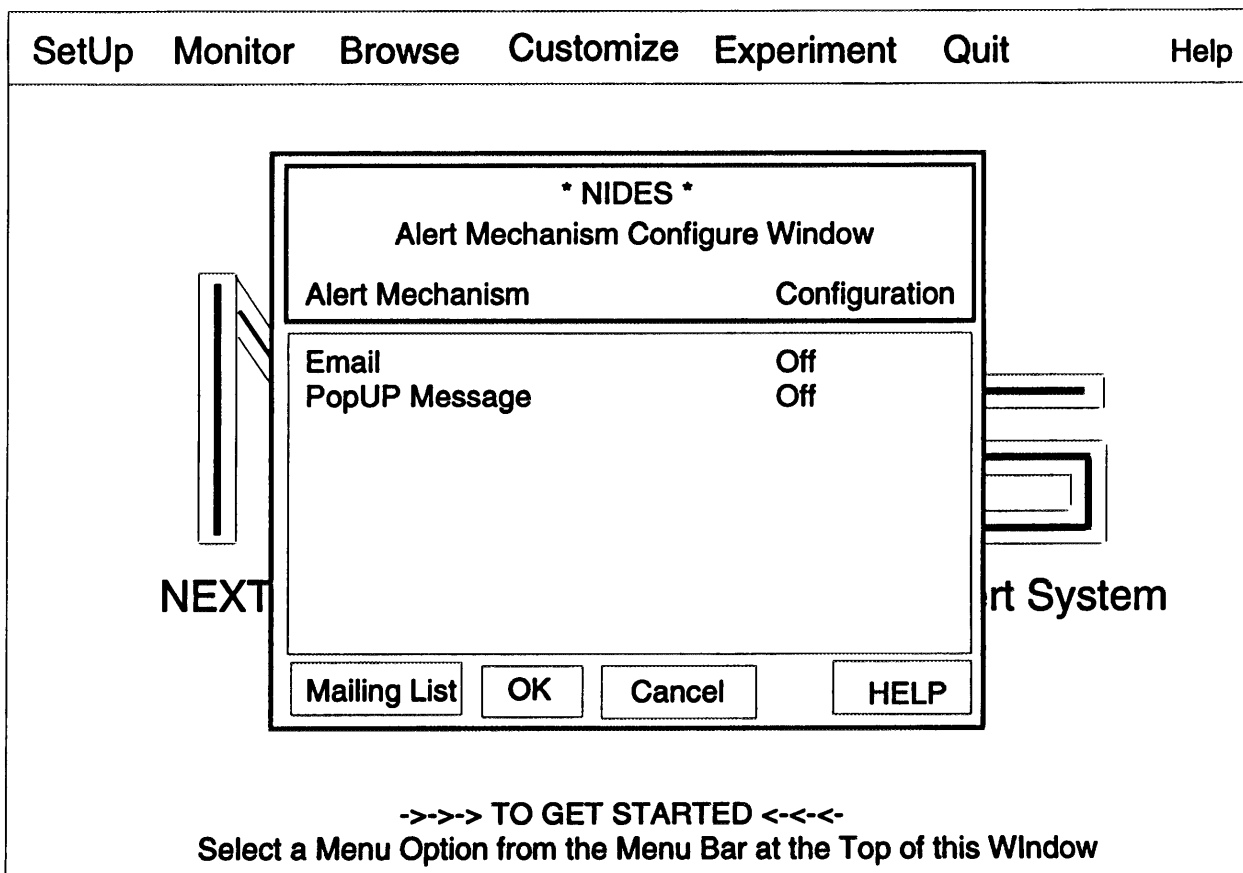
Real-Time Functions System Configuration

- **NIDES** host
 - IDES_ROOT, IPC_NAMESERVER environment variables
 - ipc_nameserver process
 - Privileged user list
- Target hosts
 - Installation of agend and agen
 - Run agend
 - Target hosts can also be configured while NIDES is running

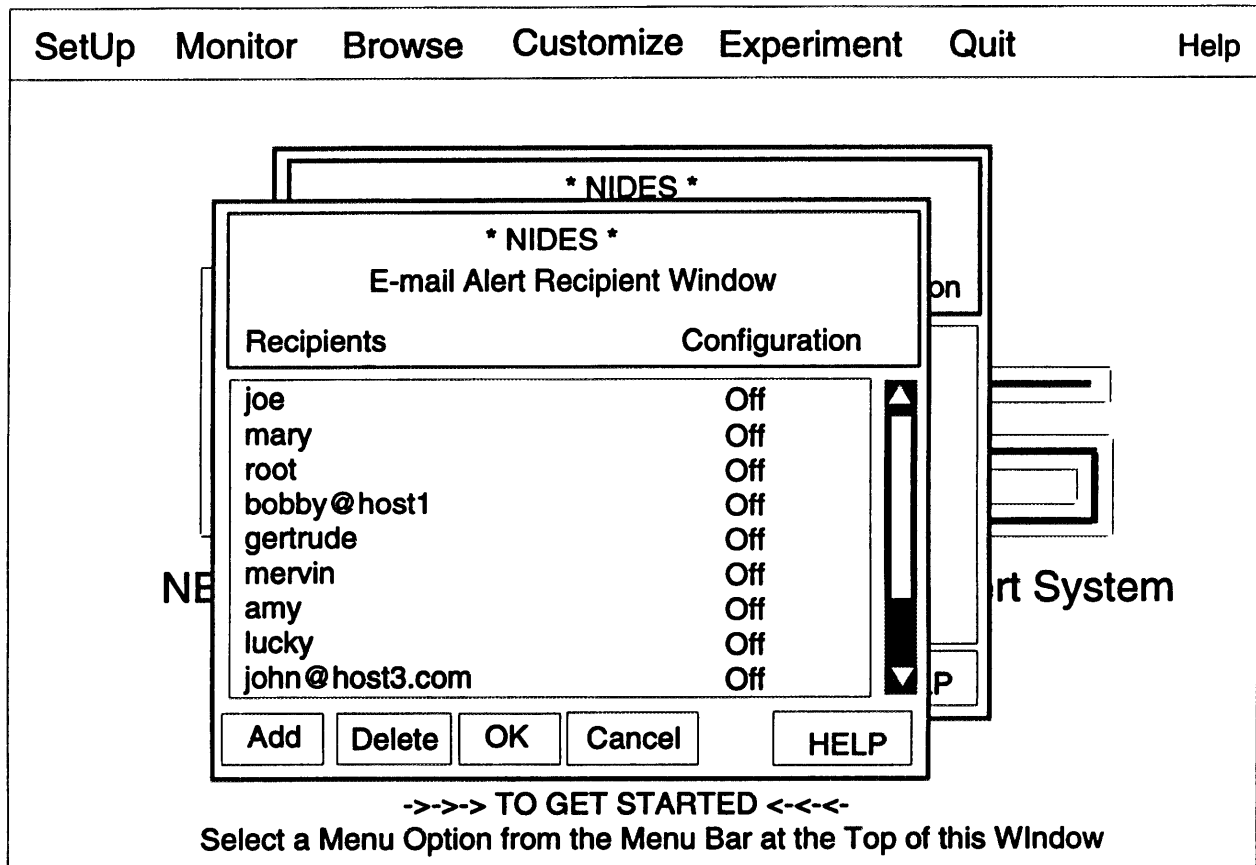
Initiating Real-Time Operation

- Start NIDES analysis
(Setup Menu – Analysis option)
- Configure alert mechanisms
(Setup Menu – Alert Method option)
 - E-mail and/or popup window
 - Both alert methods can be OFF
NIDES will archive all alerts
automatically
 - If e-mail is ON, list of recipients should
be configured

Alert Configuration Window



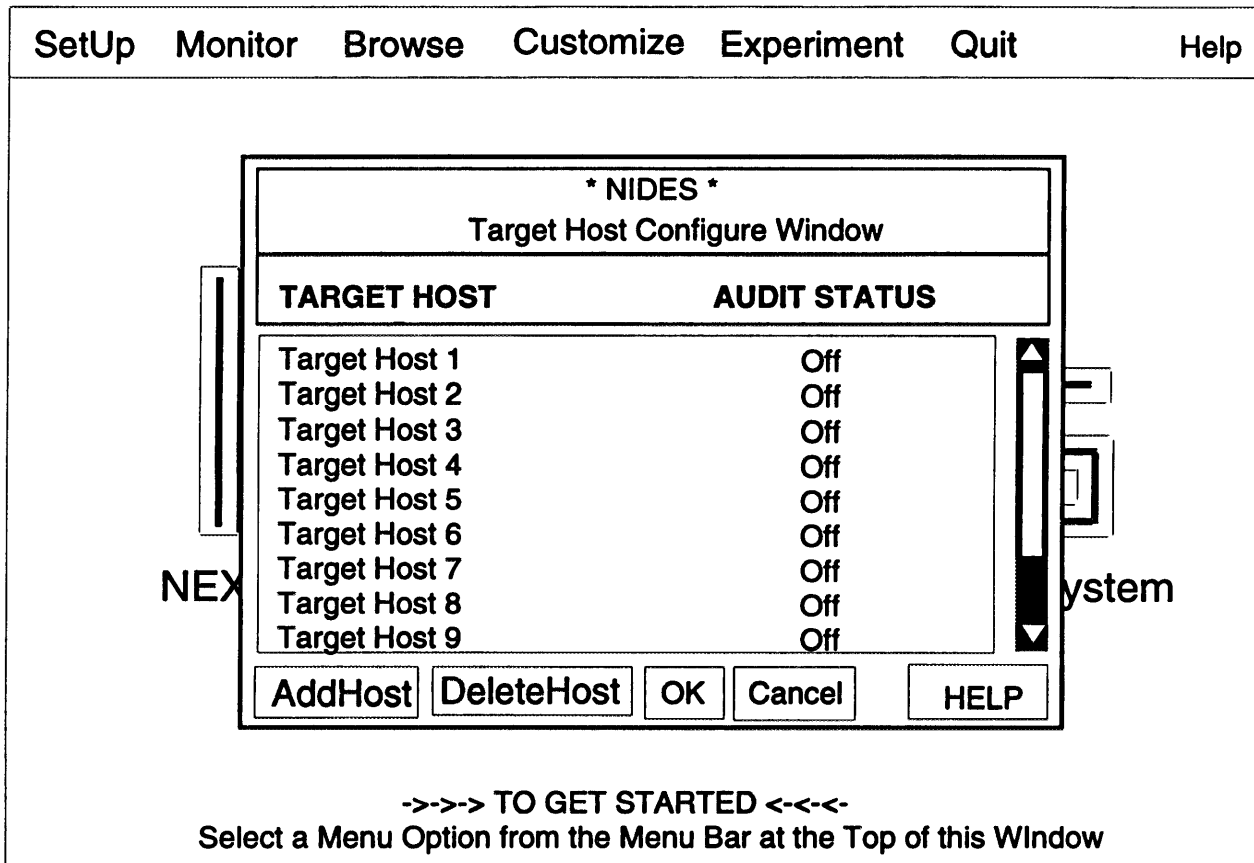
E-mail Recipients Window



Target Host Activation

- Configure targets hosts
(Setup Menu – Target Host option)
- NIDES target host list starts empty
- Each target host must be entered before it can be activated (initial configuration OFF)
- Target hosts are verified when entered
 - Format (alphanumerics, “_”, “.”, “-”)
 - Host tables

Target Host Window



Alert Filter Configuration

- Configure alert filters
(Setup Menu – Alert Filter option)
- Filters suppress real-time alert reporting
(alerts are still archived)
- Configured per subject
- Three filter configurations
 - Rulebased alerts filtered
 - Statistical alerts filtered
 - All alerts filtered

Alert Filter Configuration Window

SetUp Monitor Browse Customize Experiment Quit Help

* NIDES *

Alert Filter Configure Window

Subject	RB Alert	Stat Alert
tamaru		OFF
gilham	OFF	
lunt		OFF
debra	OFF	OFF

AddFilter DeleteFilter OK Cancel HELP

->->-> TO GET STARTED <-<-<-
Select a Menu Option from the Menu Bar at the Top of this Window

Result Filter Configuration

- Configure result filter via Customize Menu Live Instance option (Result Filter option)
- Specifies level of results archived
- One result record is generated for each audit record processed
- Each result record is assigned one of three levels: Safe, Warning, or Critical
- Three possible configurations
 - Critical level results archived
 - Critical and warning level results archived
 - All results archived

Result Filter Configuration Continued

- Minimum configuration archives Critical results only
- Default filter value is “Warning and Above” (Critical and Warning level results)
- Set filter to highest level possible to save disk space and speed up processing

Result Filter Configuration Window

* NIDES * Result Filter Window INSTANCE: test 3		
Result Filter Switch:	<input type="text" value="Critical Results Only"/>	
<input type="button" value="OK"/>	<input type="button" value="Cancel"/>	<input type="button" value="HELP"/>

Archiver Functions

- Optional process activated via Setup Menu Archiver option
- Can be started only after analysis has been initiated
- Archiver places each each audit record processed in the NIDES real-time audit data archive
- Archiver process obtains audit data from the arpool process

Archiver Functions Continued

- NIDES audit data archives stored in compressed format (via freeze) to conserve disk space
- Use archiver judiciously --- archived data consumes disk space
- If native format audit data is archived, archiver should not be activated
- Archiver is switched OFF by default

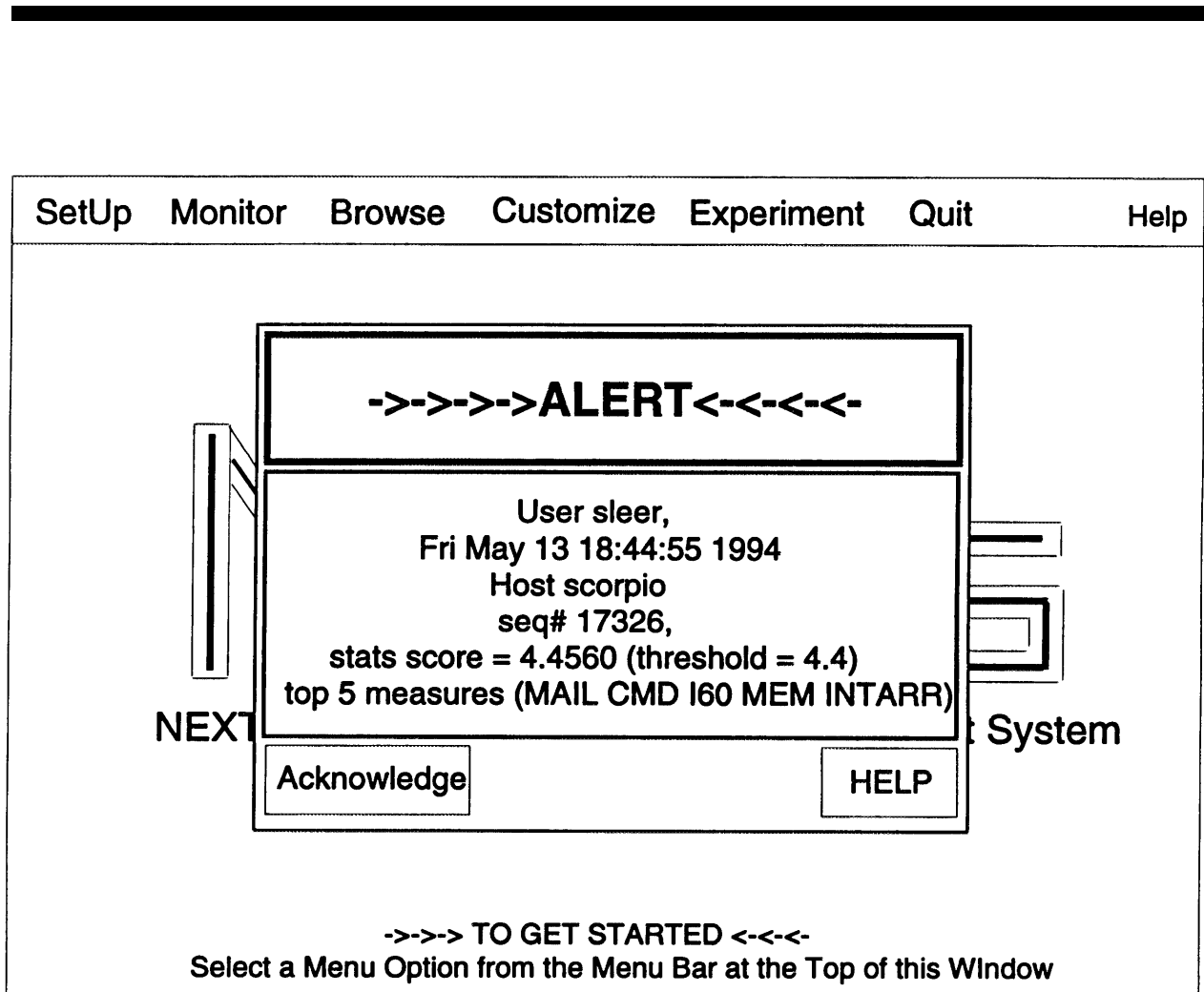
Receiving Real-Time Alerts

- When either real-time alert reporting mechanism is activated, NIDES will report the alert immediately after the resolver determines an audit record produced a ‘Critical-level’ result that is an alert
- E-mail alert reporting
 - Alert message e-mailed to all activated recipients immediately after resolver reports the alert to the UI
 - Recommended alert reporting method when NIDES host console is unattended
 - E-mail alert messages make a useful log

Receiving Real-Time Alerts Continued

- Popup window alert reporting
 - Alert window pops up and a bell sounds immediately after the resolver reports the alert to the UI
 - Displayed alert windows must be acknowledged before any NIDES functions can be accessed
 - Use popup method judiciously

Alert Window



Monitoring Real-Time Status

- Monitor Menu provides two options that provide status information on NIDES real-time operation (System and Targets)
- Monitor windows can remain displayed while other NIDES functions are accessed

Monitor Menu – System Option

- Provides ON/OFF state of the real-time analysis, arpool, and archiver processes
- Shows time each process was last started or stopped
- Provides counts of audit records processed and alerts generated since analysis was started and during the past hour
- Audit record counts are provided by arpool
- Alert counts are provided by the resolver

System Monitor Window

* NIDES *			
NIDES System Status Window			
NIDES PROCESSES	STATUS	TIME STARTED/STOPPED	
Analysis	ON	03/29/94 15:42:24	
Arpool	ON	03/29/94 15:42:24	
Archiver	Off	00:00:00	
		SINCE START-UP	PAST HOUR
Audit Records Processes		0	0
Alerts Received		0	0
<input type="button" value="DONE"/>		<input type="button" value="HELP"/>	

Monitor Menu – Targets Option

- Lists all target hosts known to NIDES
- Shows audit configuration ON or OFF for each target host
- Shows the state of each target host UP or DOWN: UP indicates arpool has received audit data from the target host
- Displays audit records received since the target host was turned ON and during the past hour

Monitor Menu – Targets Option Continued

- | Displays alerts generated by each target host since activation and during the past hour
- Target hosts may be listed as ON and DOWN if they are inactive when first turned on (this is not an error)

Target Host Monitor Window

* NIDES *						
Target Host Status Window						
HOST	AUDIT	STATE	AUDIT RECORDS		ALERTS	
			Total	Past Hour	Total	Past Hour
alpha.beta.com	ON	down	0	0	0	0
callie.zen.com	off	down	0	0	0	0
davros.skaro.com	off	down	0	0	0	0
ensor.orac.com	off	down	0	0	0	0
gandalf.middle.com	ON	UP	5098	5098	5	5
vila.zen.com	off	down	0	0	0	0

DONE HELP

Browsing Real-Time Data

- | Browse Menu provides three options that support review of
 - Audit data
 - Results
 - Instances
- Result and audit data displayed can be seconds to minutes behind actual real-time processing

Browse Menu – Audit Data Option

- | Supports review of audit data contained in any NIDES audit data archive
- Real-time audit data archive is called “real-time”
- | Four retrieval parameters are used (archive, subjects, time, and data view)
- An archive must be selected before other retrieval parameters can be entered
- One or more subjects must be selected as part of search key
- Start and end timestamps are used as part of the search key

Browse Menu – Audit Data Option Continued

- Default start/end timestamps encompass the entire archive date range
 - Seven data view options determine which fields in each audit data record are presented — an eighth option displays all fields
 - Selection of a view option initiates the retrieval — a status window is displayed during the retrieval process
 - A single retrieval is limited to 5,000 records
- Retrieved records can be saved to an ASCII text file

Audit Data Browse Window

-- NIDES -- Audit Data Browse Window																		
ARCHIVE SELECTION	SUBJECT SELECTION	TIME RANGE SELECTION																
archive_1 archive_2 archive_3 archive_4 archive_5 archive_6	<table border="1"><thead><tr><th>Available Subjects</th><th>Subjects to display</th></tr></thead><tbody><tr><td>root</td><td>ides</td></tr><tr><td>user_1</td><td>user_3</td></tr><tr><td>user_5</td><td></td></tr><tr><td>sys_admin</td><td></td></tr><tr><td>tmp_user</td><td></td></tr><tr><td>admin_user</td><td></td></tr><tr><td>user_16</td><td></td></tr></tbody></table>	Available Subjects	Subjects to display	root	ides	user_1	user_3	user_5		sys_admin		tmp_user		admin_user		user_16		From 06/28/93 00:05:02 To 07/31/93 23:58:41
Available Subjects	Subjects to display																	
root	ides																	
user_1	user_3																	
user_5																		
sys_admin																		
tmp_user																		
admin_user																		
user_16																		
Current Selection: archive_3	Subject Options: <input type="button" value="Clear"/> <input type="button" value="All"/>																	
Number of Records: 568790	RETRIEVED RECORD COUNT:																	
< Data Area >																		
View Options: <input type="button" value="Basic"/> <input type="button" value="System"/> <input type="button" value="Host"/> <input type="button" value="User"/> <input type="button" value="Resource"/> <input type="button" value="File"/> <input type="button" value="Misc"/> <input type="button" value="All"/>																		
<input type="button" value="Done"/> <input type="button" value="SaveToFile"/>	<input type="button" value="HELP"/>																	

Browse Menu – Live Results Option

- Supports review of real-time analysis result data
- Three retrieval parameters are used (subjects, time range, and result type)
 - One or more subjects must be selected
- Start and end timestamps are used as part of the search key
- Default start/end timestamps encompass the entire result archive date range
- Timestamps can be modified to narrow search

Browse Menu Live Results Option Continued

- Four result-type options further determine which records are retrieved (StatAlerts, RBAlerts, AllAlerts, or AllResults)
- Two sets of record counts are presented for the result archive (processed and archived)
- Counts are presented for alerts, critical-level, warning-level, and safe-level results, and totals
- Critical result records encompass alert records

Browse Menu – Live Results Option Continued

- Archived records are a subset of processed records
- Differences between processed counts and archived counts are due to the configuration of the result filter
- Selection of one of the four view options initiates the retrieval process — a status window is displayed during the retrieval
- A single retrieval is limited to 5,000 records
- Retrieved records can be saved to an ASCII text file

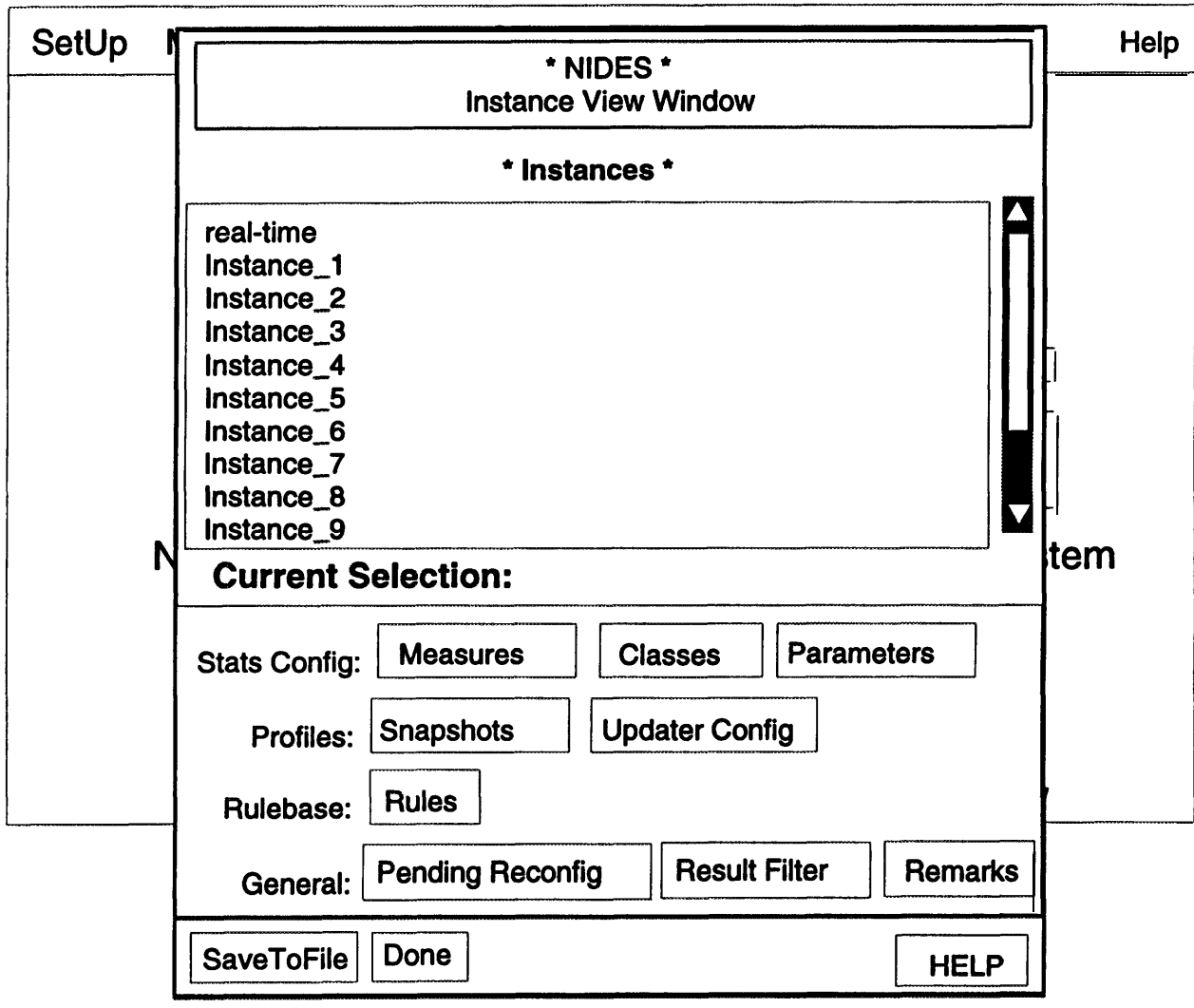
Result Data Browse Window

-- NIDES --														
Analysis Results View Window														
Test Instance Selection	Subject Selection	Time Range Selection												
july_inst july_root_only_adset_index real-time-old test2 test3	<table border="1"> <thead> <tr> <th>Avail Subjects</th> <th>Subjects to display</th> </tr> </thead> <tbody> <tr> <td></td> <td>caveh</td> </tr> <tr> <td></td> <td>debra</td> </tr> <tr> <td></td> <td>hogan</td> </tr> <tr> <td></td> <td>root</td> </tr> <tr> <td></td> <td>teo</td> </tr> </tbody> </table>	Avail Subjects	Subjects to display		caveh		debra		hogan		root		teo	From 06/28/92 00:15:02 to 07/31/92 23:58:41
Avail Subjects	Subjects to display													
	caveh													
	debra													
	hogan													
	root													
	teo													
Current test: test2	Subject options: <input type="button" value="Clear"/> <input type="button" value="All"/>													
TEST INSTANCE NAME: test 2		TIME STARTED: 04/21/94 15:53:42												
AUDIT DATA SET: jul_adset_test.Z		TIME FINISHED: 04/21/94 17:11:30												
- RECORD COUNTS -														
	ALERTS	CRITICAL	WARNING	SAFE	TOTAL									
Processed:	80	101	122	200895	201118									
Archived	80	101	122	0	223									
NUM. OF RECORDS: 223/223			NUM. OF ALERTS: 80											
<pre> root @ qslax 07/31/92 04:30:54 203738 (C) 4.0485 (3.9999) INT600 MEM INT60 COMMD IO (3. Alert: User root, Fri Jul 31 04:30:54 1992, Host qslax seq# 203738 stats score = 4.0485 (threshold = 3.99986), root @ qslax 07/31/92 04:30:54 203739 (C) 4.1370 (3.9999) INT600 MEM COMMD INT60 IO (3. root @ qslax 07/31/92 04:30:54 203740 (C) 4.1370 (3.9999) INT600 MEM COMMD INT60 IO (3. root @ qslax 07/31/92 04:30:55 203741 (C) 4:1370 (3.9999) INT600 MEM COMMD INT60 IO (3. root @ qslax 07/31/92 04:30:55 203742 (C) 4:1370 (3.9999) INT600 MEM COMMD INT60 IO (3. root @ qslax 07/31/92 04:30:55 203743 (C) 4:0480 (3.9999) INT600 MEM INT60 COMMD IO (3. root @ qslax 07/31/92 04:30:55 203744 (C) 4.1482 (3.9999) INT60 INT600 MEM COMMD IO (3. Alert: User root, Fri Jul 31 04:30:55 1992, Host qslax seq# 20374 stats score = 4.1482 (threshold = 3.99986) root @ qslax 07/31/92 04:30:55 203745 (W) 3.9596 (3.9999) INT60 INT600 MEM COMMD IO (3. root @ qslax 07/31/92 15:35:38 208737 (C) 0.2794 (3.9999) MEM HOUR IO CPU INT600 (1. Alert: User root, Fri Jul 31 15:35:38 1992, Host qslax seq# 208737 rulebase rule BadLoginAnomaly: Bad login by root reported by qslax current total without success 24 </pre>														
View options: <input type="button" value="StatAlerts"/> <input type="button" value="RBAAlerts"/> <input type="button" value="AllAlerts"/> <input type="button" value="AllResults"/>														
<input type="button" value="Done"/>	<input type="button" value="SaveToFile"/>	<input type="button" value="HELP"/>												

Real-Time Instance Viewing

- Real-time instance configuration review via Browse Menu Instances Option
- Items available for review are
 - Measures
 - Classes
 - Parameters
 - Snapshots
 - Updater Config
 - Rules
 - Pending Reconfig
 - Result Filter
 - Remarks

Instance View Window



Real-time NIDES Operation (Hands-On)

Real-time (Hands-On) Exercise

- Activate real-time analysis
- Configure alert methods
- Configure result filter
- Configure target hosts
- Activate archiver
- ┆ Generate and receive alerts
- Configure alert filters
- Browse result and audit data

Day 2 Viewgraphs

Day 2 — Agenda

- Overview of configuration options
- Configuration application
- Rulebase configuration

Overview Of Configuration Options

NIDES Analysis Configuration

- NIDES Beta version provides functions to configure statistical and rulebased analysis for real-time and batch modes
- Customize Menu provides configuration interface
(Live Instance and Test Instances options)
- Real-time analysis configuration changes can be made while analysis is running
- Batch analysis configurations are made prior to execution of a batch run
- Some configuration changes are applied immediately; others are deferred until the next profile update

Statistics Configuration Options

- Measures
 - ON/OFF state
 - QMAX
 - Scalar
 - Short-term half-life
 - Minimum effective-N
- Classes
 - Measure category classes (editors, compilers, shells, window commands, mailers)
 - Tmp file filter class

Statistics Configuration Options Continued

- Parameters
 - Training period
 - Long-term half-life
 - Red (critical) threshold
 - Yellow (warning) threshold
 - Maximum sum for rare category probability
 - Profile cache size

Statistics Configuration Options Continued

- Profile Management
 - Profile update schedule (real-time only)
 - Profile update flags — ON/OFF per subject (real-time only)
 - Profile update flag — ON/OFF globally (test instances only)
 - Profile deletion, replacement, and copying
 - Initiate nonscheduled profile update per subject (real-time only)

Rulebase Configuration

- Rules turned ON/OFF
- New rules can be compiled and are available to NIDES immediately
- rb_config file
 - 25 sections specify various configuration lists used by the NIDES rulebase
 - rb_config file read when analysis started, contents asserted into rulebase factbase
 - rb_config file allows for straightforward customization of NIDES default rulebase

Configuration Application

Analysis Configuration Application

- Immediate application method applies configuration changes as soon as reconfiguration message is received by the analysis components
- Configuration changes applied immediately
 - Turning rules ON or OFF
 - Profile cache size
 - Profile options
 - Turning measures ON or OFF

Analysis Configuration Application Continued

- Deferred application method applies configuration changes at next profile update (scheduled or user initiated)
- Configuration changes applied at next profile update
 - Measure QMAX, scalar, short-term half-life, and minimum effective-N
 - Class list changes
 - Statistics parameters options except profile cache

Rulebase Configuration

Rulebase Configuration Process

- Review rb_config file and default rules to see if they can address your problem
- Determine scenario new rules need to address if default rulebase cannot be configured to meet your needs
- Review audit trail to locate relevant data
- Write prototype rule(s)
- Collect sample audit data containing one or more versions of scenario
- Test new rule(s) using NIDES test facility
- If results are satisfactory, introduce new rule(s) into real-time operation

Rule Concepts

- Facts, factbase and factbase maintenance
- Marks
- Priorities
- Sets
- Ptypes
- Rule inference groups
- Rulebased analysis execution

Rule Concepts Continued

- Rule syntax
- rb_config file
- Default rulebase
- Rule installation
- Rulebase security

Facts & the Factbase

- Transitory rulebase information is stored in facts
- Factbase is the rulebase's repository of facts
- Ptypes are the templates that define fact structures

Factbase Maintenance

- Factbase size should be kept to minimum
- Facts should be deleted as soon as possible for three reasons:
 - Prevents the same rule from firing repeatedly
 - Reduces factbase search times
 - Prevents unbounded growth of rulebase process
- Rules that delete facts must ensure that all rules interested in the fact have already examined it

Rule Marks

- Marks applied to facts so rules process the fact once only
- Marks can be any letters
- Marks can be applied and removed
- Rules in the same group usually use the same mark
- Marks help control execution flow
- Facts can be tested for a mark
- Antecedent clauses test for marks
- Consequent clauses apply or remove marks

Rulebase Priorities

- Rules can be assigned priorities
- Rules are tested in order of priority from high to low
- Priorities must be from -96 to 99
- Default priority is 0

Rulebase Priorities — Examples

```
rule[BadRoot(#50;*):  
    [antecedent clause]  
==>  
    [consequent clause]  
]
```

```
rule[BadLogin(#-20):  
    [antecedent clause]  
==>  
    [consequent clause]  
]
```

Rulebase Sets

- Sets are analogous to C' enumerated types
- A set maps an identifier to an integer
- Rulebase sets
 - Audit action (ia)
 - Audit record source (src)
 - Result codes (m)

Rulebase “ia” Set Members

- Action types assigned by agen, acc2ia, or audit2ia

```
set[ia: VOID,          DISCON,
        ACCESS,       OPEN,
        WRITE,        READ,
        DELETE,       CREATE,
        RMDIR,        CHMOD,
        EXEC,         CHOWN,
        LINK,         CHDIR,
        RENAME,       MKDIR,
        MOUNT,        UNMOUNT,
        LOGIN,        BAD_LOGIN,
        SU,           BAD_SU,
        EXIT,         LOGOUT,
        UNCAT,        RSH,
        BAD_RSH,      PASSWD,
        RMOUNT,       BAD_RMOUNT,
        PASSWD_AUTH,  BAD_PASSWD_AUTH
]
```

Rulebase “m” and “src” Set Members

- Possible rulebase result levels

```
set[m: SAFE, WARNING, CRITICAL]
```

- Audit data source codes assigned by
agen, acc2ia, or audit2ia

```
set[src: IA_SRC_VOID, IA_SRC_C2,  
        IA_SRC_PACCT, IA_SRC_APPLICATION,  
        IA_SRC_LINK, IA_SRC_BSMV1,  
        IA_SRC_BSMV2]
```

Rulebase Ptypes

- Ptypes are templates for facts
(similar to C' structure declarations)
- Users may not define new ptypes
- Frequently used ptypes are
 - event
 - generic
 - generic_config

Rulebase Ptype — event

- Used to store audit data
- event facts should not be asserted/deleted
- Rules may apply, delete, or test for marks on events facts

event Ptype

```
ptype[event
    targid:string,
    real_userid:string,
    current_userid:string,
    otheruser:string,
    file:string,
    action:ia,
    response:int,
    rhost:string,
    term:string,
    process_id:int,
    cmd:string,
    cputime:float,
    audit_src:src,
    hi_sequence:int,
    lo_sequence:int,
    timerec:ptime,
    timegen:ptime
]
```

Rulebase Ptype — generic

- Available for user-defined rules
- Not used by any NIDES default rules
- Rules may assert or delete generic facts
- Rules may apply, delete, or test for marks on generic facts

generic Ptype

```
ptype[generic
    id:string,
    s1:string,
    s2:string,
    s3:string,
    s4:string,
    i1:int,
    i2:int,
    i3:int,
    i4:int
]
```

Rulebase Ptype — generic_config

- Supports rb_config file configuration of user developed rules
- generic_config facts are initialized with rb_config file GENERIC_CONFIG section contents
- generic_config facts should NOT be asserted or deleted
- Rules should NOT apply any marks to generic_config facts

generic_config Ptype

```
ptype[generic_config
      id:string,
      sval:string,
      ival:int
]
```

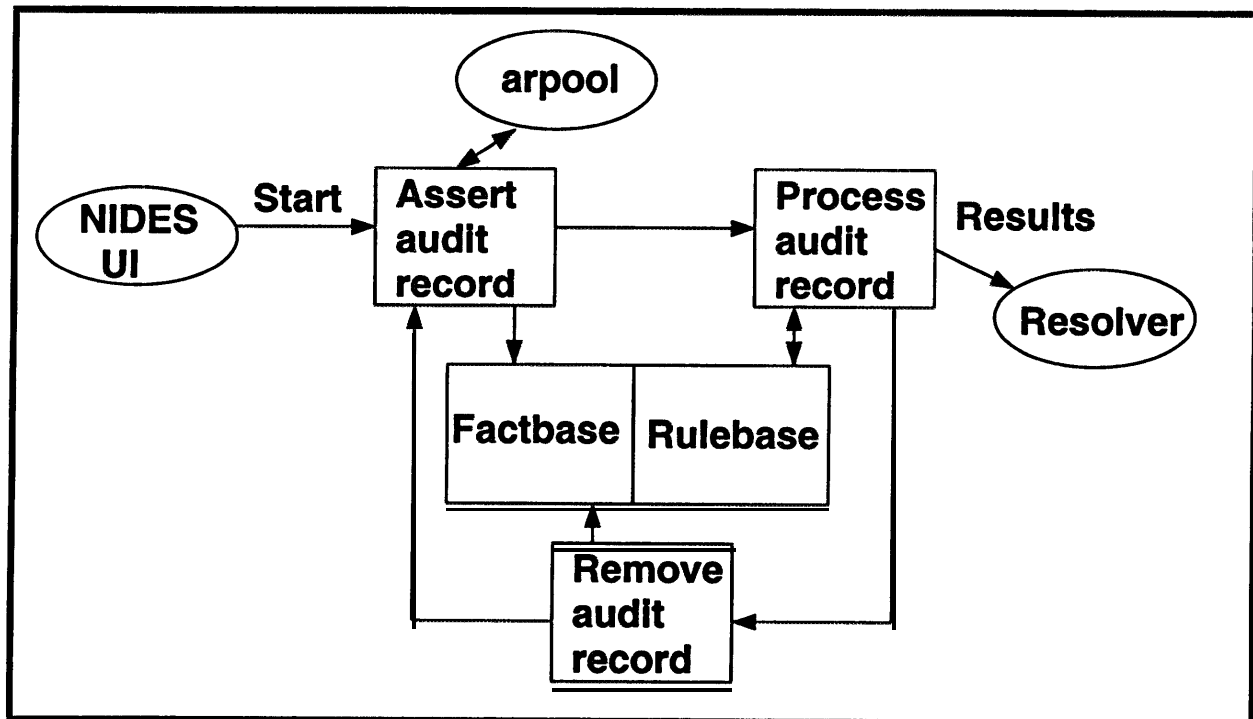
Rule Inference Groups

- A grouping of rules that performs a particular inference
- Using multiple rules allows more complex tests over multiple audit records
- Rules are generally mutually exclusive
- | Rules in same group generally apply the same mark to facts and test for the absence of that mark

Rulebased Analysis Execution

- Assert audit record (“event” fact) into factbase
- Process audit record
- Remove audit record from factbase
- Assert next audit record

Rulebased Analysis Execution Flow Diagram



Rule Syntax

- Rules contain two parts: head and body
- Rule head
 - Rule name
 - Optional rule priority
 - Optional rule operating modes
- Rule body
 - Antecedent: tests performed by the rule
 - Consequent: actions performed by the rule if antecedent tests satisfied

Basic Rule Structure

```
rule[Rulename(#Priority;Mode):  
    [Antecedent Clause]  
    [Antecedent Clause]  
    ...  
==>  
    [Consequent Clause]  
    [Consequent Clause]  
    ...  
]
```

Rule Body

- Rule antecedent
 - Test factbase for existence or nonexistence of facts
 - Compare facts
 - Alias facts
 - Examine facts for marks
- Rule consequent
 - Delete facts
 - Assert facts
 - Apply marks to facts
 - Remove marks from facts
 - Generate an alert report

Rule Antecedent Syntax

Factbase Tests

- If multiple facts satisfy antecedent test most recently asserted or modified fact is returned
- Test for existence of a fact

```
[+event|action == ia#LOGIN]
```

- Test for fact and alias fact

```
[+ev:event|action == ia#DELETE,  
  audit_src == src#IA_SRC_BSMVI,  
  real_userid != 'root']
```


Rule Antecedent Syntax

Factbase Tests Continued

- Test for absence of a fact

```
[-session|userid == 'root']
```

```
[-ev:event|action == ia#EXEC]
```

Rule Antecedent Syntax

Compare Fact Values

- Generally facts are aliased before tests are performed

```
[?|ev.real_userid == se.userid]
```

```
[?|ev.response > 0]
```

```
[?|ev.action != ia#BAD_RSH  
|| ev.action != ia#BAD_SU]
```

Rule Antecedent Syntax Mark Tests

- Test for existence of marks

```
[+ev:event$BADROOT]
```

```
[+ev:event$SEENMARK|targid == 'server']
```

- Test for absence of marks

```
[-ev:event^LOG|real_userid != "root"]
```

```
[-se:session^COUNTD|count > 0]
```

Rule Consequent Syntax Marks

! All facts accessed in a rule consequent must first be aliased in the rule's antecedent

- Apply mark to fact
(fact aliased in antecedent)

`[$|ev:BADLOG]`

- Remove mark from fact
(fact aliased in antecedent)

`[^|ev:BADLOG]`

Rule Consequent Syntax Factbase Modification

- Assert fact (all fields must be initialized)

```
[+generic|id = "security violation",  
      s1 = "vila",  
      s2 = "zen.dept1.com",  
      s3 = "lapsed clearance",  
      s4 = "secret",  
      i1 = 10,  
      i2 = 15,  
      i3 = 0,  
      i4 = 0]
```

- Remove fact (fact must first be aliased)

```
[-tr]
```

Rule Consequent Syntax

Fact Modification

Modify fact

- Fact must be aliased in antecedent
- Fact has same precedence as if it was removed and a new fact asserted

```
[/gen|s4 = 'top secret',  
  i1 += 1,  
  i2 = gen.i2 * 15]
```

Rule Consequent Syntax

Generate Alert Report

- Create alert message string

```
[!]sprintf(prstr,  
    'user %s breaks root on host %s!! \n',  
    ev.real_userid, ev.targid)]
```

- Call rulebase 'inform' function

```
[!]inform(m#CRITICAL, ev.real_userid,  
    ev.timegen, ev.hi_sequence,  
    ev.lo_sequence, prstr, 'RuleName')]
```

Inference Group Example

```
rule[BadPassword1(#50;*):
  [+ev:event^BP|action == ia#BAD_PASSWORD_AUTH]
  [-bad_password|userid == ev.otheruser]
==>
  [$|ev:BP]
  [+bad_password|userid = ev.otheruser,
    count =1]
]

rule[BadPassword2(#40;*):
  [+ev:event^BP|action == ia#BAD_PASSWORD_AUTH]
  [+bp:bad_password|userid == ev.otheruser,
    count < ATTACK_THRESH - 1]
= = >
  [$|ev:BP]
  [/bp|count += 1]
]
```


Inference Group Example Continued

```
rule[BadPasswordAnomaly(#30;*):  
  [+ev:event^BP|action == ia#BAD_PASSWD_AUTH]  
  [+bp:bad_password|count >= ATTACK_THRESH - 1,  
    userid == ev.otheruser]  
  == >  
  :  
  [$|ev:BP]  
  [/bp|count += 1]  
  [!|sprintf(prstr,  
    'Sad password by %s reported by %s,  
    current total without success %d',  
    ev.otheruser, ev.targid, bp.count)]  
  [!|inform(m#CRITICAL, ev.real_userid, ev.timegen,  
    ev.hi_sequence, ev.lo_sequence,  
    prstr, 'BadPasswordAnomaly')]  
]
```

rb_config File

- 25 sections
- File located in \$IDES_ROOT/etc
- Read each time analysis (real-time or batch) invoked
- Contents of rb_config asserted into factbase

rb_config File Rule Dependencies

Section	Rule
DOMAIN	MultiLogin1 LocalLogin RemoteRootBadLogin RemoteRootBadPassword
GENERIC_CONFIG	None
HOME_DIR	ChmodOtherUser AccessPrivateFile1
KNOWN_LOGIN	KnownLogin1
LOG_DIR	TruncateLog
LOGIN_CONFIG	ChangeLoginFile
NOEXEC	BadUserExec
PARANOID_PROG	ParanoidUser1 ParanoidUser2 ParanoidUser3 ParanoidUser4
PRIVATE_DEVICE	AccessPrivateDevice
PRIVATE_FILE	AccessPrivateFile1
PROGLOCATION	TrojanHorse ModSystemExec LinkSystemExec ReadSystemExec ChmodSystemFile
PROGRAM	TrojanHorse

rb_config File Rule Dependencies Continued

Section	Rule
RAREEXEC	RunsRareExec SuspiciousUser
REMOTE_FILE_NO_ACCESS	RemoteFile2
REMOTE_FILE_NO_MODIFY	RemoteFile1
REMOTE_NO_EXEC	RemoteExec
REMOTE_NOT_OK	NoRemote
ROOT_OK	BadRoot
SPECIAL_FILE	AccessSpecialFile
SPECIAL_PROGRAM	SpecUserProgram
SPECIAL_USER	SpecUserExec
SYSTEM_SCRIPTS	ReadSystemExec
TMP_DIRNAME	DotFile
TMP_FILE	DotFile
USER_TYPE	AccessSpecialFile

rb_config File Syntax

- Each section begins with the section name
- Each section ends with the keywords
“NO-MORE”
- Syntax of contents varies with each section

rb_config File Syntax — Examples

DOMAIN

dept1.net.com

dept2.net.com

NO_MORE

HOME_DIR

jones /homes/a/jones

smith /homes/b/smith

flagg /homes/a/flagg

NO_MORE

SPECIAL_FILE

/etc/exports 1

/etc/netgroup 1

/etc/inetd.conf 1

NO_MORE

rb_config File Sections

- **DOMAIN**
Defines local network domains
- **GENERIC_CONFIG**
User-defined configurations
- **HOME_DIR**
Users and their home directories
- **KNOWN_LOGIN**
Commonly unprotected accounts
- **LOG_DIR**
Locations of log/audit files

rb_config File Sections Continued

- **LOGIN_CONFIG**

Scripts automatically executed at login/shell execution

- **NOEXEC**

Programs only "root" should execute

- **PARANOID_PROG**

Programs paranoid users execute frequently

- **PRIVATE-DEVICE**

Devices abusers can use to eavesdrop or spoof others

- **PRIVATE-FILE**

Files in users' home directory that should be accessed only by that user

rb_config File Sections Continued

- **PROGLOCATION**
Directories where system files reside
- **PROGRAM**
System programs that should be executed only from system directories as listed with a code "1" in the PROGLOCATION section
- **RAREEXEC**
Programs users don't ordinarily run
- **REMOTE-FILE-NO-ACCESS**
Files remote users should not access
- **REMOTE_FILE_NO_MODIFY**
Files remote users should not modify

rb_config File Sections Continued

- **REMOTE_NO-EXEC**
Programs remote users should not execute
- **REMOTE_NOT_OK**
Users not authorized to log in remotely
- **ROOT_OK**
Users authorized to become "root"
- **SPECIAL_FILE**
Lists files that only selected users should access

rb_config File Sections Continued

- **SPECIAL-PROGRAM**

Programs only specific users should execute; selected users are listed in **USER-TYPE** section

- **SPECIAL-USERS**

Users who should execute only specific programs; each entry lists user/program pair

- **SYSTEM-SCRIPTS**

Shell scripts that reside in system directories listed in **PROGLOCATION**

rb_config File Sections Continued

- **TMP_DIRNAME**
Temporary directories
- **TMP_FILE**
‘Dot’ files that may be written into temporary directories listed in **TMP_DIRNAME**
- **USER_TYPE**
Users allowed to access files listed in **SPECIAL_FILE**

Default Rulebase Overview

- 70 rules; 39 generate alerts
- Some rules function as a group and must be turned ON or OFF together
- Four rule groups are
 - Password/login
 - Session
 - Paranoid user
 - TFTP
- 42 marks used by default rules should not be used by new rules

Rulebase Default Rule Groups

Rule Group	Description
Password/Login BadPassword1 BadPassword2 BadPasswordAnomaly GoodPassword1 GoodPassword2 BadLogin1 BadLogin2 BadLoginAnomaly BadLoginBadPassword GoodLogin1 GoodLogin2 GoodSU1 GoodSU2	Maintains password and login information. This group counts bad password/login entries for a user, and reports an alert if a threshold is exceeded. Some of these rules update or remove bad password/login counts.
Paranoid User ParanoidUser1 ParanoidUser2 ParanoidUser3 ParanoidUser4 ParanoidUserAnom ClearParanoidUser	Maintains information about paranoid user activity.
TFTP TFTPUse TFTPAnomaly	Records tftp usage.

Rulebase Default Rule Groups Continued

Rule Group	Description
Session	
MultLogin1	Maintains information about a user's current session.
MultLogin2	
FlagRSH	Includes session type, counts of various activities, and removal of session facts when the session is terminated or remains inactive for a period of time. While none of the Session rule group rules generate an alert, many other NIDES rules rely on this groups information to function.
ConsoleLogin	
DialInLogin	
LocalLogin	
RemoteLogin	
Logout1	
Logout2	
Su1	
Exec	
ClearSession	
TouchSession	
	<i>Recommend leaving all Session group rules ON.</i>

Rulebase Default Rule Marks

Default Rulebase Marks			
APD	CSF	NR	RRBP
APF	DF	PFA	RRE
ASF	EX	PU	RSE
BAR	FA	PUA	RSH
BE	ID	RE	SSU
BLOG	KL	RF1	SU
BP	LF	RF2	SUE
BR	LO	RF3	TH
BT	LOG	RM	TL
CLF	LSE	RRBL	TU
COU	MSE		

Default Rulebase Descriptions

- Housekeeping rules (4)
Maintain session information, remove event facts, and maintain timestamp information
- Bad password rules (6)
Count and report bad password entries
- Bad login rules (5)
Count and report bad logins
- Login rules (12)
Record login events and determine the type of login (e.g., local, remote, rsh)

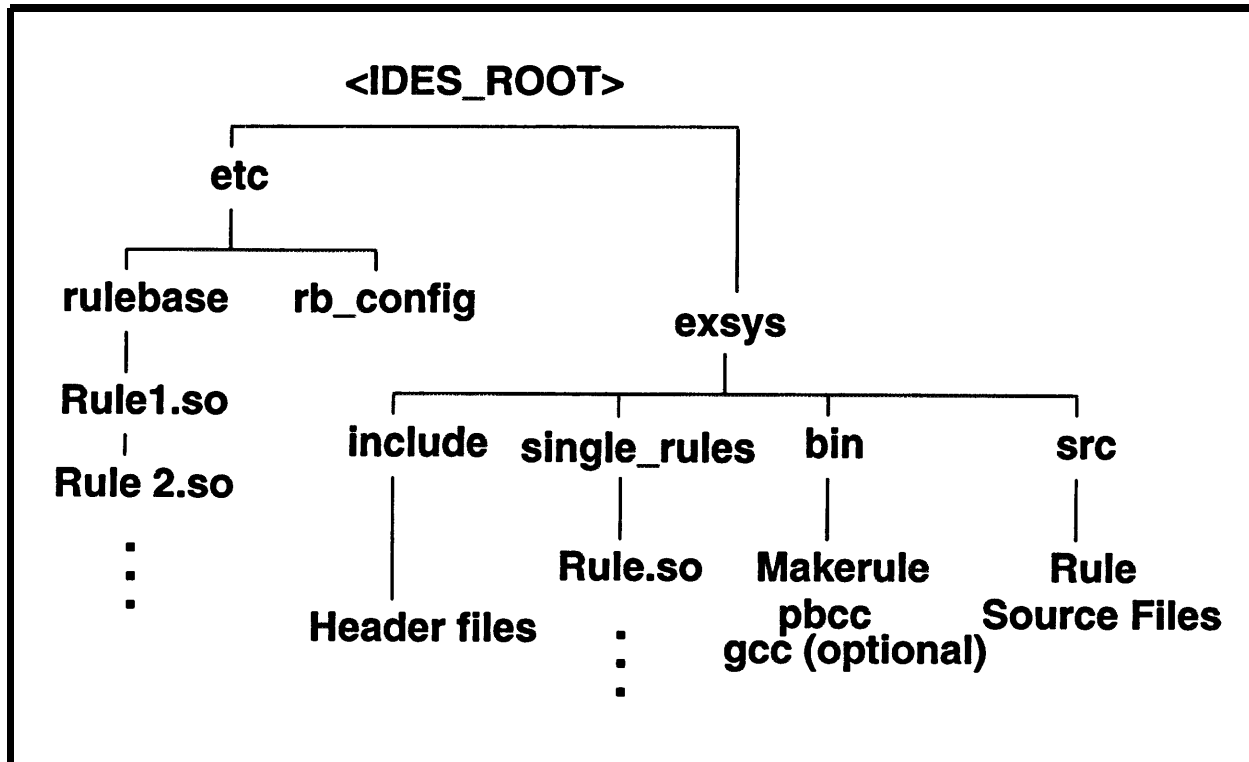
Default Rulebase Descriptions Continued

- Trojan horse rules (2)
Detect Trojan horse execution
- File and device access rules (11)
Detect access to sensitive files and devices
- Remote user rules (9)
Monitor and detect suspicious remote user activity
- User ID rules (6)
Monitor changes in user identity reporting suspicious changes primarily involving the “root” account

Default Rulebase Descriptions Continued

- FTP rules (3)
 - Monitor and detect unauthorized ftp/tftp usage
- Suspicious behavior rules (12)
 - Monitor and report suspicious user behavior, grouped into three categories
 - Hiding tracks
 - Paranoia
 - Aggregate suspicious behavior

Rulebase Directory Structure



Rule Installation

- Create rule source file 'Rulename.pb'
and place file in \$IDES_ROOT/exsys/src
- Compile rule using makerule script
\$IDES_ROOT/exsys/bin/makerule rulename
- Makerule script places rulename.so file in
\$IDES_ROOT/etc/rulebase directory
- Compile all rules that function as a group
before using them in NIDES
- Test rules using NIDES test facility before
real-time use

Rulebase Security

- Encrypt files located in `$IDES_ROOT/exsys` when not in use
- Remove default rulebase source code file `$IDES_ROOT/exsys/rulebase.src` from system
- Encrypt files located in `$IDES_ROOT/etc/rulebase` when NIDES not running (rule object files and `rb_config` file)
- Set rulebase file permissions to limit access (read and write) to authorized NIDES users

Day 3 — Viewgraphs

Day 3 — Agenda

- Rulebase configuration (hands-on)
- Statistics configuration (discussion)
- Statistics configuration (hands-on)
- NIDES test facility (discussion)

Rulebase Configuration (Hands-On)

Rulebase Configuration Exercises

- Configure rb_config file
- Write simple rule
- Write group of rules that manipulate facts
- Write rule utilizing GENERIC_CONFIG
- Compile/install new rules
- Activate/deactivate rules

rb_config File Customization

Section configuration

- DOMAIN
- GENERIC_CONFIG
- HOME_DIR
- LOG_DIR
- PROGLOCATION (review)
- PROGRAM (review)
- ROOT_OK

Rules That Manipulate Facts

- “generic” facts are the only facts that new rules should assert/delete/modify (marks can be used with most facts)
- generic fact format

```
ptype[generic
    id:string,
    s1:string,
    s2:string,
    s3:string,
    s4:string,
    i1:int,
    i2:int,
    i3:int,
    i4:int
```

```
]
```

Rules That Manipulate Facts Continued

- Assertion of generic fact in consequent

```
[+generic|id = "security alert",  
  s1 = "slocomb",  
  s2 = "baby.lab2.com",  
  s3 = "changed password",  
  s4 = "",  
  i1 = 1,  
  i2 = 0,  
  i3 = 0,  
  i4 = 0]
```

Rules That Manipulate Facts Continued

- Deletion of a generic fact
 - Alias fact in antecedent

```
[+gen:generic] id=="security alert",  
                s1==ev.real_userid]
```

- Delete fact in consequent

Rules That Manipulate Facts Continued

- Modification of a generic fact
 - Alias fact in antecedent

```
[+gen:generic| id == "security alert"  
                s1 == ev.real_userid,  
                s2 == ev.file
```

- Modify fact in consequent

```
[/gen| i1 += 1]
```


Using rb_config File GENERIC_CONFIG Section

- GENERIC_CONFIG section useful for runtime configuration of new rules
- generic_config fact format

```
ptype[generic_config
      id:string,
      sval:string,
      ival:int
]
```

- generic_config facts should not be modified by any rules (assert, delete, modify, or marks)

Using rb_config File

GENERIC_CONFIG Section

Continued

- Use of generic_config in rule antecedents

```
[+generic_config] id == 'limited_host_user',  
                  sval == ev.real_userid]
```

```
[-generic_config] id == 'limited_host',  
                  sval == ev.targid]
```

Using rb_config File

GENERIC_CONFIG Section Continued

- Corresponding rb_config file entries

```
GENERIC_CONFIG
#Begin limited host user list
limited_host_user sleer 0
limited_host_user orion 0
# Begin limited host list
limited_host carbon 0
limited_host      zinc      0
NO_MORE
```

Rule Compilation/Installation & Activation/Deactivation

- “makerule” script compiles and installs rules (reads IDES_ROOT environment variable)
- Real-time rulebased analysis configuration
 - Customize menu “live instance” option
 - Live instance “rulebase” option
 - Rules can be turned on/off only when real-time analysis is activated

Rule Compilation/Installation & Activation/Deactivation Continued

Batch analysis rulebased configuration

- Customize menu ‘test instances’ option
- Test instance management ‘MODIFY’ option
- Test instance customization ‘Rulebase’ option

Rule Compilation/Installation & Activation/Deactivation Continued

- rb_config file use
 - real-time analysis reads rb_config when analysis started
(`$SIDES_ROOT/etc/rb_config`)
 - batch analysis reads rb_config when test starts
(`$SIDES_ROOT/etc/rb_config`)

Rulebase Configuration Window

NIDES
Rulebase Configuration
INSTANCE: real-time

RULENAME	STATUS
ClearParanoidUser	ON
ClearSession	ON
ConsoleLogin	ON
CuriousUser	Off
DialInLogin	ON
DisCon	Off
DatFile	ON
Exec	ON
FTPAnomaly	ON
FlagRSH	ON

OK

CANCEL

HELP

Statistics Configuration (Discussion)

Statistics Configuration Options

- Measures
- Classes
- Parameters
- Profile management
- Updater configuration (real-time only)
- Updater mode (batch only)
- Manual update (real-time only)

Measure Configuration Status

- Measure may be configured ON or OFF
- Measures turned ON contribute to score calculation once trained
- All measures are trained regardless of ON/OFF status
- Intensity measures (I60,I600,I3600) and the audit record distribution measure (ARECDIST) should be configured ON

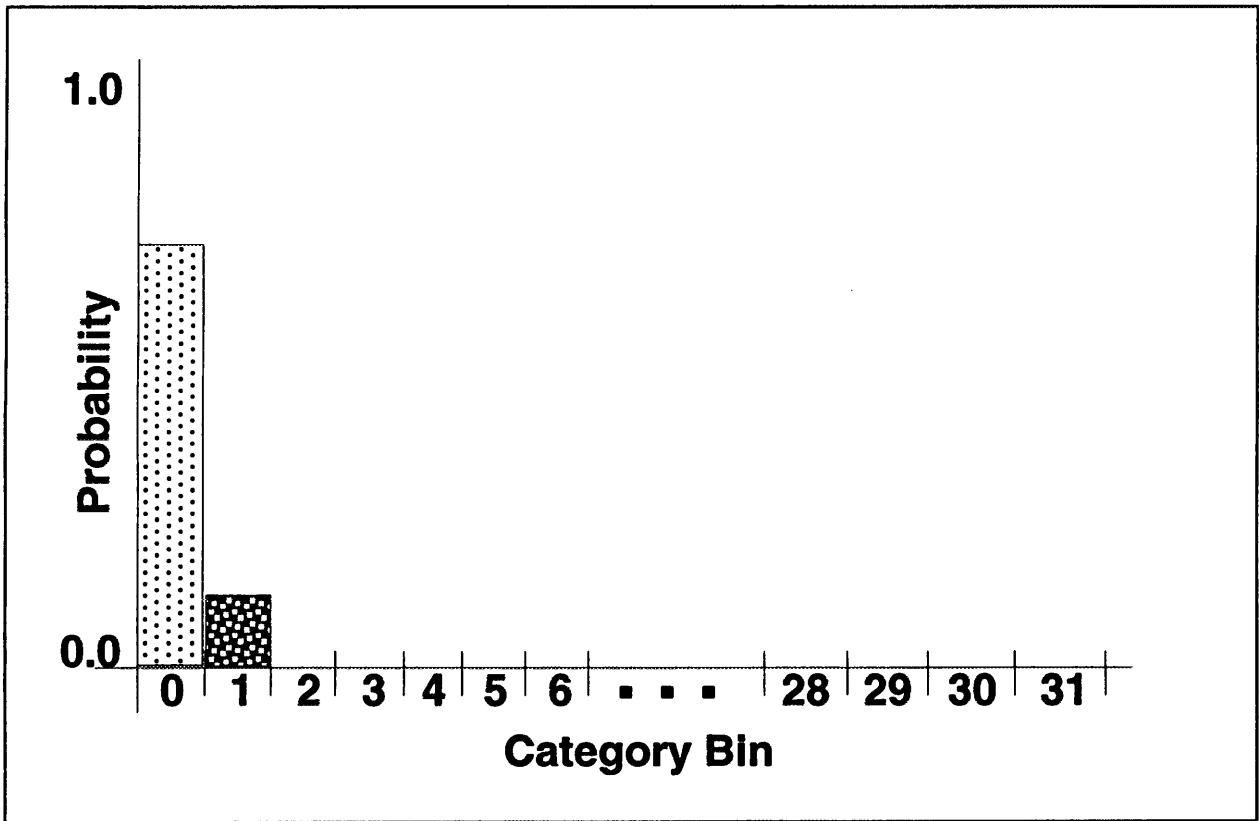
Measure Configuration Status Continued

- Measures likely to aid in differentiating users should be activated
- Measures likely to be similar across many/most users can be deactivated
- False alarms triggered consistently by the same measure/measures may indicate the measure/measures should be deactivated
- Changes to measure status are applied immediately

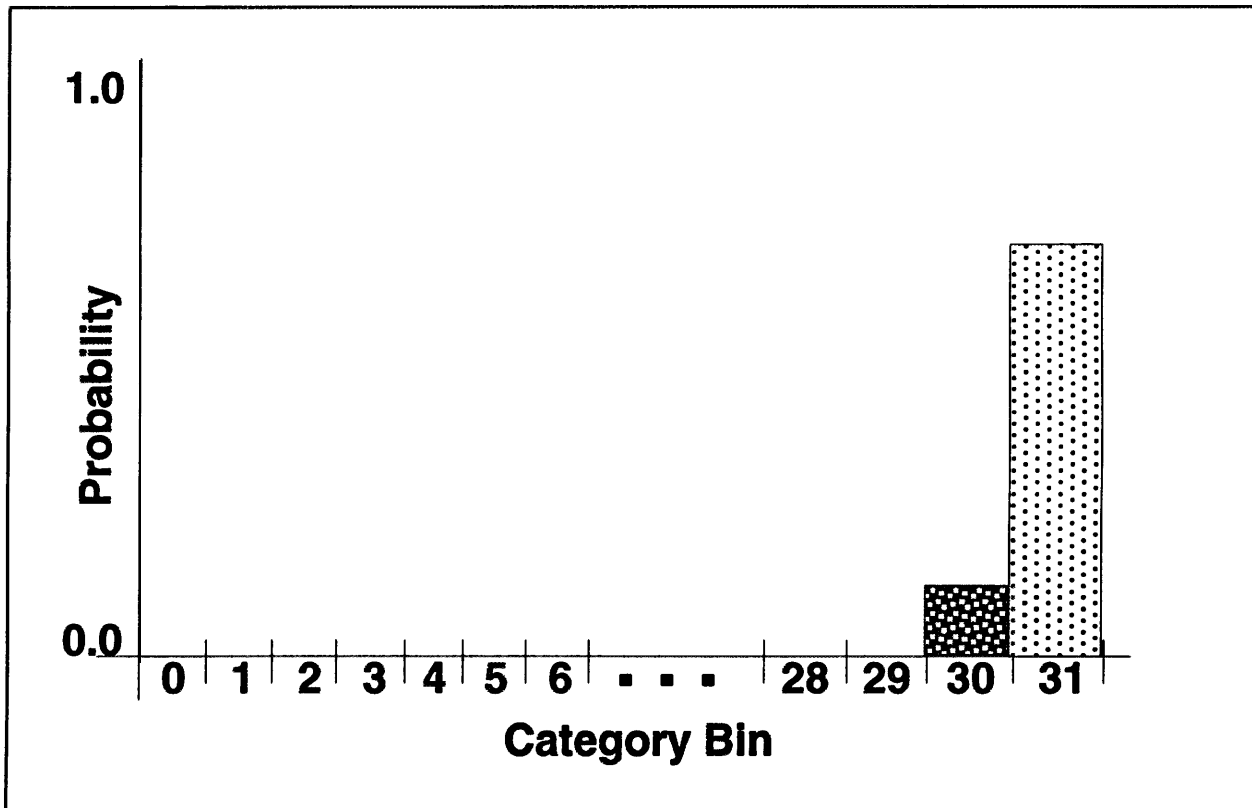
Measure Configuration QMAX

- Valid values 10 to 1000
- Determines binning ranges for Q distribution
- Changes to QMAX are seldom needed
- Q probabilities clustered at one end of the bin ranges for most subjects indicates QMAX may need adjustment
- Changes to QMAX require measure to go off-line for Q and T2 training phases
- Review Q probabilities only after Q training completed

Measure QMAX High



Measure QMAX Low



Measure Configuration Scalar

- Valid values 0 to 100,000,000
- Valid for continuous measures only
- Determines binning ranges for categories
- Should be set larger than highest value ever likely to be observed for the measure (even factors of 10 are acceptable)
- Changes to Scalar are rarely needed

Measure Configuration Scalar Continued

- Category probabilities clustered at one end of the bin ranges for most subjects indicates the Scalar may need adjustment
- Full measure retraining (C,Q,T2) is needed when Scalar changed
- Review category probabilities only after C training completed

Scalar High

***** NIDES *****
Profile View Window
INSTANCE: demo SUBJECT: root

Last Profile Update: Fri Jul 31 01:00:00 1992 Number of Profile Updates: 34
Last Audit Record Timestamp: Fri Jul 31 01:00:00 1992

Profile Item	Categories					
Measure Status Measure Misc Info Categories Q & S values Q distribution table Tails of Q dist'n table Daily Q bin counts T2 distribution table T2 counts (daily) Misc profile data	PROB (COUNT)	Type	AGECNT	PREVOBSCNT	CATID	CA RA
	.0000 (0)		0.3655	0.0000	10	
	.0000 (0)		0.0000	0.0000	9	
	.0000 (0)		0.0000	0.0000	8	
	.0000 (0)		0.0000	0.0000	7	
	.0000 (0)		0.0000	0.0000	6	
	.0000 (0)		0.0000	0.0000	5	
	.0000 (0)		0.0000	0.0000	4	
	.0000 (0)		0.0000	0.0000	3	
	.0000 (0)		0.0000	0.0000	2	
	.0001 (0)		0.0000	0.0000	15	
	.0100 (0)		0.0000	0.0000	1	
	.8900 (0)		0.0000	0.0000	0	

Scalar Low

--- NIDES ---
Profile View Window
INSTANCE: demo SUBJECT: root

Last Profile Update: Fri Jul 31 01:00:00 1992 Number of Profile Updates: 34
Last Audit Record Timestamp: Fri Jul 31 01:00:00 1992

Profile Item	Categories					
Measure Status	PROB (COUNT)	Type	AGECNT	PREVOBSCNT	CATID	CA
Measure Misc Info	.0000 (0)		0.3655	0.0000	20	RA
Categories	.0000 (0)		0.0000	0.0000	21	
Q & S values	.0000 (0)		0.0000	0.0000	22	
Q distribution table	.0000 (0)		0.0000	0.0000	23	
Tails of Q dist'n table	.0000 (0)		0.0000	0.0000	24	
Daily Q bin counts	.0000 (0)		0.0000	0.0000	25	
T2 distribution table	.0000 (0)		0.0000	0.0000	26	
T2 counts (daily)	.0000 (0)		0.0000	0.0000	28	
Misc profile data	.0009 (0)		0.0000	0.0000	11	
	.0016 (0)		0.0000	0.0000	7	
	.0500 (0)		0.0000	0.0000	30	
	.8700 (0)		0.0000	0.0000	31	

SaveToFile Done HELP

Measure Configuration Minimum Effective-N

- Valid values 0 to 100,000
- Represents minimum number of observations needed (modified by aging factors) before measure contributes to score calculation
- Effective N =
$$\sum_i \gamma^i \text{DailyCount}_i$$

γ = long-term aging factor
 i indexes the day
- Can be set high to prevent rarely observed measures from contributing to the score too soon

Measure Configuration

Minimum Effective-N

Continued

- No measure retraining needed
- Current training stage not affected by changes to minimum effective-N
- Measures in training will complete the next stage of training when one third of the current minimum effective-N aged observations are made
- Increasing minimum effective-N during training will lengthen the training period for remaining training stages
- Decreasing minimum effective-N during training may shorten the training period

Measure Configuration

Short-term Half-life

- Valid values 0 to 100,000
- Should be approximately 5% of typical user's daily audit record activity for the measure
- Low values shorten time range represented in short-term profile, possibly generating more false alarms
- High values lengthen time range represented in short-term profile, possibly reducing detection sensitivity
- Changes to short-term half-life require Q and T2 training

Measure Configuration Window

-**- NIDES -**-
Statistics Measures Configuration
INSTANCE: real-time

MEASURE	DESCRIPTION	TYPE	STATUS
U_CPU	User_CPU_Usage	CONT	ON
U_IO	User_I/O_Usage	CONT	ON
U_MEM	User_Memory_Usage	CONT	ON
U_LOC	User_Physical_Location_of_Use	CAT	Off
U_MAIL	User_Mailer_Usage	CAT	ON
U_EDIT	User_Editor_Usage	CAT	ON
U_COMPILER	User_Compiler_Usage	CAT	Off
U_SHELL	User_Shell_Usage	CAT	Off
U_WINDOW	User_Window_Command_Usage	CAT	Off
U_COMMD	User_General_Command_Usage	CAT	ON

Current selection: U_CPU
Measure status: ON 16/49

Qmax value: Minimum Effective-N:
Scalar value: Short-Term Halflife:

OK Cancel HELP

Class Configuration

- Class lists define categories for some measures
- New class members should be added prior to observation
- No retraining is required with class list modification
- Class list changes are applied at the next profile update

Class Configuration Continued

- Class members may be added or deleted
- All class lists should be reviewed during installation
- LOCALHOSTS and TMPDIRS classes should be configured during installation
- Valid values — alpha-numeric and /

Statistics Classes

- COMPILER
Used by U_COMPILER measure
- EDITOR
Used by U_EDIT
- MAILER
Used by U_MAIL
- SHELL
Used by U_SHELL
- WINDOW
Used by U_WINDOW

Statistics Classes Continued

- NETWORK

Used by U_RNETTYP

- LOCALHOSTS

Used to determine if a host is local or remote, affects U_RNET and U_LNET measures

- e TMPDIRS

Used to filter out temporary files and directories from the U_FILE and U_DIR measures

TMPDIRS Class

- Has direct effect on performance
- Temporary files are filtered out, thus reducing process and profile file size
- TMPDIR classes list directory prefixes
- All files under a TMPDIR directory are filtered
- Example — /tmp
files /tmp/joe and /tmp893 would be filtered
- Example — /tmp/
file /tmp/joe filtered and /tmp893 not filtered

Class Configuration Window

--- NIDES ---
Statistics Classes Configuration
INSTANCE: real-time

CLASSES	Class items for TEMPORARY FILES
COMPILERS EDITORS MAILERS SHELL ENVIRONMENTS WINDOW COMMANDS NETWORK COMMANDS LOCAL HOSTS	/tmp /var/tmp
:TEMPORARY:FILES:	

Current class item selected: TEMPORARY FILES

Parameters Configuration

- Default values generally acceptable in most cases
- Profile cache size configuration useful for performance tuning
- All changes applied at next profile update except cache size changes, which are applied immediately

Parameters Configuration

Long-term Half-life

- Time period, measured in profile updates, after which data is downweighted by one half
 - Larger half-lives mean older data takes longer to be aged out of profiles
 - Smaller half-lives mean the long-term profile reflects more recent activity and older data is more quickly forgotten

Parameters Configuration

Training Period

- Interval of time, measures in profile updates, required before statistical anomalies will be reported
 - Shorter training periods may increase false-alarm rates
 - Longer training periods mean more stable profiles, but also mean a longer time must elapse before statistical anomaly detection is on-line

Parameters Configuration Thresholds

- Red/Critical & Yellow/Warning Threshold
Represent percentage used to determine red and yellow threshold values
 - Smaller values cause fewer audit records to be flagged at the red and yellow levels because the thresholds will be set higher
 - Larger values may cause an excessive number of records to be flagged as suspicious

Parameters Configuration

Max Sum of Rare Category

- Maximum probability sum for categories grouped into the 'RARE' class
 - Changes rarely warranted

Parameters Configuration

Profile Cache Size

- Most recently needed/used profiles are kept in the profile cache, others are checkpointed to disk
 - Smaller cache sizes keep process size small but may slow processing if profiles are swapped frequently — useful if NIDES host has limited memory
 - Larger cache can speed processing if process growth is not too great

Parameters Configuration Window

***- NIDES *-**
Statistics Parameters Configuration
INSTANCE: real-time

Long-term profile half-life:	<input type="text" value="20.00"/>	Updates
Training Period:	<input type="text" value="20.00"/>	Updates
Red/Critical threshold:	<input type="text" value="0.1000"/>	%
Yellow/Warning threshold:	<input type="text" value="1.0000"/>	%
Max Sum of Rare Cat Probs:	<input type="text" value="0.01"/>	
Profile Cache Size:	<input type="text" value="5"/>	

Parameters Default & Valid Values

Parameter	Default	Valid Values
Long-term Half-life	20	1-365 days
Training Period	20	1-365 days
Red/Critical Threshold	0.1%	0.001%-100.0%
Yellow/Warning Threhsold	1.0%	0.001%-100.0%
Max Sum Rare Prob.	0.01	0.0001-0.25
Profile Cache	5	1-100

Profile Management Copying

- Copies one subject's profile into new non-existent subject's profile
- Useful to quickly provide trained profile for a new subject
- Subjects should be very similar if copying is used to initialize a profile
- Initial false alarm rates may be high for new subject

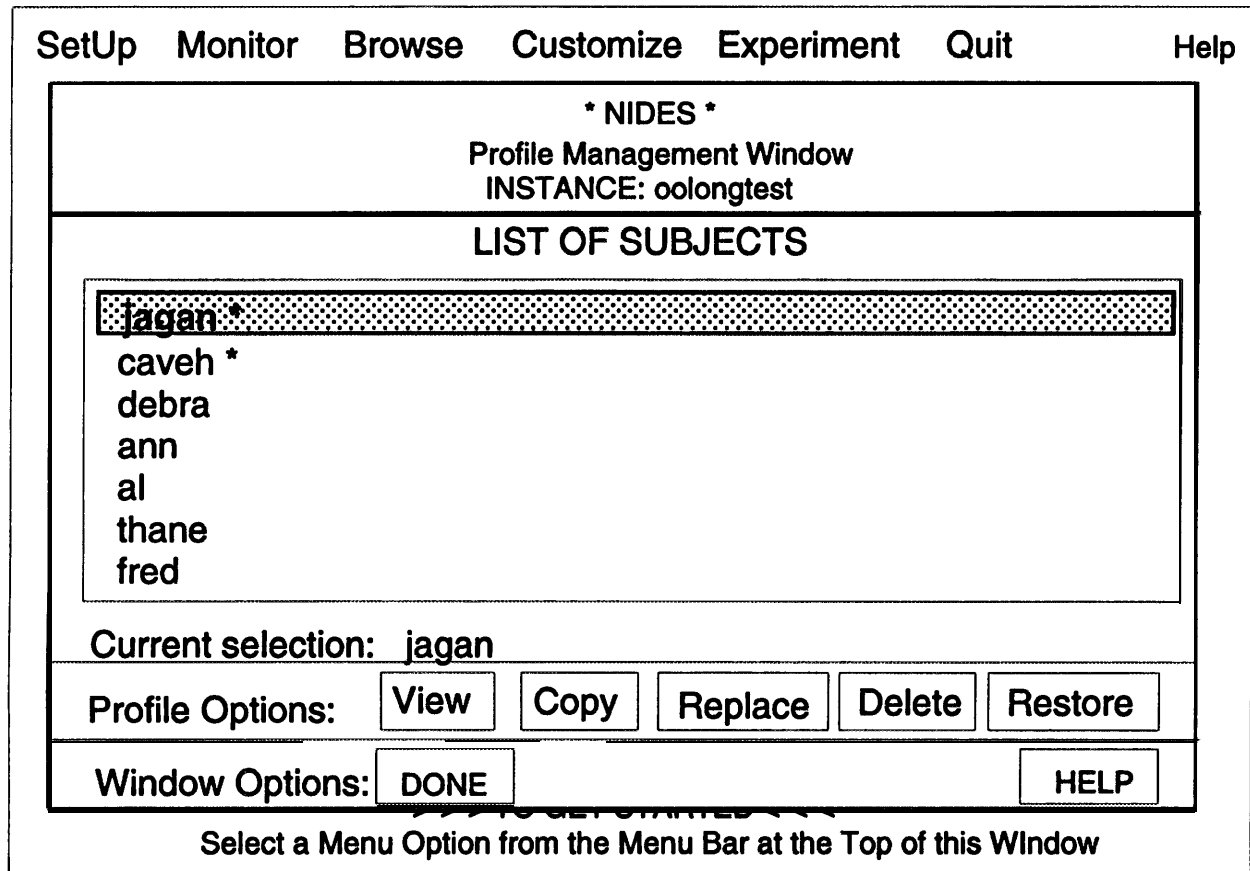
Profile Management Replacement

- Replaces a subject's profile with the profile of another subject
- Useful for cross-profiling experiments
- Replaced profile can be copied to a temporary profile first to checkpoint it

Profile Management Deletion

- Deletes a subject's profile
- Profiles for users no longer on the system can be deleted to save space
- A user's profile can be retrained from scratch by deleting the existing profile — a new profile is generated as soon as data is seen for the subject
- For experiments, profile deletion can help control which data is used to train a profile
- Deletion should be used with caution

Profile Management Window



Profile Update Configuration (Real-time)

- Updater Schedule
 - Time daily updates occur
 - Default is 00:00:00
- Update Method
(Audit Record Timestamp or System Clock)
 - System Clock
Based on system's clock
 - Audit Record Timestamp
Based on timestamps of audit records processed

Profile Update Configuration (Real-time) Continued

- Subject Update Flag (ON or OFF)
 - Specifies which subjects' profiles are updated
 - May be turned OFF when a subject's behavior is expected to deviate from normal activity in an acceptable manner

Profile Update Window (Real-time)

--- NIDES ---			
Profile Update Configuration Window			
INSTANCE: real-time			
(view-only mode)			
PROFILE UPDATE STATUS	PROFILE UPDATE SCHEDULE		
<table><tr><td>PROFILE UPDATING ON</td><td>PROFILE UPDATING OFF</td></tr></table> <div style="border: 1px solid black; padding: 5px;"><p>AUpwdauthd caveh debra dodd donovan guest jagan neumann root tamaru</p></div> <p>Profile Update Options: <input type="button" value="ALL ON"/> <input type="button" value="ALL OFF"/></p>	PROFILE UPDATING ON	PROFILE UPDATING OFF	<p>Profile updating for all subjects will occur daily at:</p> <div style="border: 1px solid black; padding: 2px;">00:00:00</div> <p>PROFILE UPDATE METHOD</p> <div style="border: 1px solid black; padding: 2px;">Audit Record Timestamp</div>
PROFILE UPDATING ON	PROFILE UPDATING OFF		
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	<input type="button" value="HELP"/>		

Manual Profile Updating (Real-time Only)

- Performs an instantaneous profile update on selected subjects
- Forces application of pending configuration changes
- Only selected subjects with activity since last update will be updated
- Should not be used to accelerate profile training

Manual Update Window

--- NIDES ---
Trigger Profile Updater Window
(real-time only)

Available Profiles	Profiles to Update
<div style="border: 1px solid black; padding: 2px;">ATUpwdauthd</div> <ul style="list-style-type: none">auditcavehdebradodddonovanguestjaganluntneumann	<div style="border: 1px solid black; height: 150px; width: 100%;"></div>

Subject List Options:

Updater Configuration (Batch)

- Profile update flag can be set ON or OFF
- ON flag required when training profiles
- OFF flag useful when testing detection rates

Profile Update Window (Batch)

* NIDES * Profile Update Mode Window INSTANCE: july-test		
Profile Updater Switch:	<input type="text" value="ON"/>	
<input type="button" value="OK"/>	<input type="button" value="Cancel"/>	<input type="button" value="HELP"/>

Statistics Configuration (Hands-On)

Statistics Configuration Exercises

- Configure/install ascii stat_config file
- Real-time configuration
 - Profile cache
 - Measures
 - Classes
 - Updater configuration
 - Manual update
 - Profile management

ASCII stat_config File Configuration

- Backup default stat_config file located in \$IDES_ROOT/etc
- Modify two class lists
 - TMPDIRS
 - LOCALHOSTS
- Modify measure ON/OFF configuration
- Modify statistics parameters
- Build/install configuration file using init_stat_config command

ASCII stat_config File Format Class Lists

- Begins with 'BEGINCOMMANDCLASSES'
- Ends with 'ENDCOMMANDCLASSES'
- Each class must be listed on one line followed by carriage return
- Each line's format is
 'Class Name' = "member-1, member-2, ... "
- Maximum line length 1000 characters

ASCII stat_config File Format Class Lists Example

```
BEGINCOMMANDCLASSES
COMPILER=gcc,cc,g++
EDITOR=emacs,vi,ed,edit
MAILER=mm,mail,mh,send,comp
SHELL=csh,sh
WINDOW=X,xinit,suntools,xcalc,cmdtool
NETWORK=rsh,ftp,kermit,rcp,rdist
MISC=
LOCALHOSTS=zen,orac,slave,orion,holly
TMPDIRS=/tmp,/var/tmp
ENDCOMMANDCLASSES
```

ASCII stat_config File Format Measures

- Begins with 'BEGINMEASURES'
- Ends with 'ENDMEASURES'
- All measures must be listed
- Each measure has seven fields

ASCII stat_config File Format

Measure Fields

- Measure ID
Should NOT be changed
- Status
Valid values are ON and OFF
- Measure Type
Should NOT be changed — possible values are CAT, CONT, BINCONT
- QMAX
Floating Point Number between 10 and 1000

ASCII stat_config File Format Measure Fields Continued

- Weight
Not used in this NIDES release
- Scalar
CONT measures only, floating point
number between 0 and 1,000,000,000
- Measure description
Should NOT be changed

ASCII stat_config File Format Measures Example

```
BEGINMEASURES
U_CPU      ON    CONT 100.0  0.0  1000.0    Description
U_IO       ON    CONT 100.0  0.0  10000000.0 Description
U_MEM      ON    CONT 100.0  0.0  10000000.0 Description
U_LOC      OFF   CAT  100.0  0.0  0.0       Description
...
...
U_ARECDIST ON    CAT  100.0  0.0  0.0       Description
U_INT60    ON    CONT 100.0  0.0  0.0       Description
U_INT600   ON    CONT 100.0  0.0  0.0       Description
U_INT3600  ON    CONT 100.0  0.0  0.0       Description
ENDMEASURES
```


ASCII stat_config File Format Parameters

- Begins with BEGINPARAMS
- Ends with ENDPARAMS
- One parameter entry per line
- Each line's format is
 'Parameter Name' = 'Parameter Value'

ASCII stat_config File Format Parameters Continued

- Relevant parameters
 - Long-term profile half-life
Floating point number from 1 to 365
 - Yellow threshold percentage
Floating point number from 1 to 100
 - Red threshold percentage
Floating point number from 1 to 100
 - Training days
Floating point number from 1 to 365
 - Max sum rare probability
Floating point number from 0.0001 to 0.25

ASCII stat_config File Format Parameters Continued

- Unused parameters
 - AR_HALFLIFE
 - CORR_CUTOFF
 - MIN_EFFN
 - NO-UPDATE-MODE

ASCII stat_config File Format Parameters (Example)

```
BEGINPARAMS  
AR_HALFLIFE=100.0  
PROF_HALFLIFE=20.0  
CORR_CUTOFF=99.0  
MIN_EFFN=100.0  
YELLOW_PERC=0.01  
RED_PERC=0.001  
TRAINING_DAYS=20  
MAXSUMRARE=0.01  
NO_UPDATE_MODE=  
ENDPARAMS
```

Real-time Statistics Configuration

- Activate real-time analysis before performing configuration functions
- Real-time configuration via Customize Menu Live Analysis option
- Profile Cache
Parameters Option (Real-time Instance Configuration Window)
- Measures
Measures Option (Real-time Instance Configuration Window)
- Classes
Classes Option (Real-time Instance Configuration Window)

Real-time Statistics Configuration Continued

- Updater Configuration
Updater Config Option (Real-time Instance Configuration Window)
- Summary Option displays all configuration changes
 - “OK” initiates all changes
 - “Cancel” cancels all changes

Real-time Statistics Configuration Continued

- Browse Menu Instances options provides review of pending reconfigurations
- Manual Update
Manual Update Option (Real-time Instance Configuration Window)
- After manual update confirmed configuration changes will be applied
- Manual updates may take some time, depending on the number of subjects updated

Profile Management

- Profile Mgmt Option (Real-time Instance Configuration Window)
- Back up a profile by making a copy
- Replace backed-up profile with another subject's profile
- Back up a second profile by making a copy
- Delete the profile
- Restore the first profile that was replaced
- Replace the profile with another subject's profile

Real-time Instance Configuration Window

SetUp	Monitor	Browse	Customize	Experiment	Quit	Help
-------	---------	--------	-----------	------------	------	------

* NIDES * Instance Configuration Window INSTANCE: real-time			
Statistics Options:	Measures	Classes	Parameters
Profile Options:	Profile Mgmt	Updater Config	Manual Update
Rulebase Options:	RuleBase		
General Options:	Result Filter	Remarks	
Summary	OK	Cancel	HELP

->->-> TO GET STARTED <-<-<-
Select a Menu Option from the Menu Bar at the Top of this Window

NIDES Test Facility (Discussion)

Test Facility Description

- Tests process audit data in batch mode
- Tests can run concurrently with real-time analysis
- Tests use instances and audit data sets
- Tests are configured prior to execution, NOT during execution

Test Facility Description Continued

- To run a test, an instance and audit data set must be specified
- Test results are written into the **NIDES** result archive
- Test results cannot be reviewed until the test completes
- Test facility should be used to test new configurations prior to using them in real-time operation
- **NIDES** batch runs can be initiated outside of the user interface using the batch-analysis utility program

Audit Data Sets

- Audit data sets are the source of data for NIDES batch runs
- Audit data sets contain NIDES format audit records
- Each audit data set has an index file
- An audit set can be “real” or “virtual”

Audit Data Sets Continued

- Real audit data sets
 - Have a data file
 - Save time when running in batch mode
 - Take time to generate
- Virtual audit data sets
 - Do NOT have a data file --- index file lists the NIDES archive containing the actual data
 - Save space
 - Generated in a matter of seconds
 - Preferred audit data set type

Audit Data Set Creation

- Customize Menu Audit Data Sets option
 - Data extracted from a NIDES audit data archive
 - NIDES audit data archives created with archiver utility program
 - Data set specification --- subject list, time range and type (“real” or “virtual”)
- NIDES utility programs
 - audit2ia and acc2ia convert native format audit records to NIDES format
 - adset_index utility creates an index file for NIDES audit data file, making the file into an audit data set

Test Instances

- Each test must have an instance
- Instances store configuration information and subject profiles
- Instance names and test names are synonymous
- NIDES contains default instance “real-time”, which cannot be used for experiments
- Instances can be created, modified, copied and deleted
- Instances can be reused for multiple tests
- Instance data is stored in `$IDES_ROOT/storage/instances`

Test Configuration Options

- Measures (same as real-time configuration)
- Classes (same as real-time)
- Parameters (same as real-time)
- Profile Mgmt (same as real-time)
- Updater Mode (Used only for test configuration)
- Rulebase (same as real-time)
- Result Filter (same as real-time)
- Remarks (same as real-time)
- Profile Synchronization (Used only for test configuration)

Test Configuration Options

Updater Mode

Updater Mode can be ON or OFF

- Configured via Customize Menu
- When updater is ON, profiles will be updated
 - Profiles updated daily based on audit record timestamps
 - Useful for training profiles
- When updater is OFF, profiles will NOT be updated
 - Useful for detection-performance tests
 - Not appropriate when a newly created instance is used

Test Configuration Options

Profile Synchronization

- Profile synchronization can be ON or OFF
- Configured via Experiment Menu Setup & Exec option (i.e., when test is initiated)
- Synchronizes each profile's last audit record timestamp with test audit data's earliest timestamp
- Default configuration is OFF

Test Configuration Options

Profile Synchronization Continued

- Synchronization ON
 - Useful when timestamps of audit data set are earlier than existing profiles' last update/audit record timestamps
 - Not needed with newly created instances (i.e., no profiles)
- Synchronization OFF
 - Profiles will NOT be updated until audit records timestamps surpass the profiles' last update timestamp
 - Appropriate for newly created instances or when audit data timestamps are later than previously processed audit data

Test Facility Uses

- Test new configurations prior to real-time use
- Rapidly build trained subject profiles
- Analysis of archived audit trails
- Evaluate NIDES performance under various configurations
- Tune NIDES performance

Test Status Reporting & Management

- Experiment Menu Status & Results option provides information on active and completed tests
- Active Test Status Reporting
 - Lists all active **NIDES** batch runs and their start times
 - Updates counts for audit records and alerts approximately every 10 seconds

Test Status Reporting & Management

- Complete test reporting
 - Lists all tests contained in NIDES results archive
 - Shows time test completed and audit record and alert counts
- Test status window functions
 - Viewing completed test results (comparable to Browse Menu Test Results option)
 - Deletion of test results (instance used for test is NOT deleted — i.e., profiles and configuration)

Day 4 — Viewgraphs

Day 4 — Agenda

- Test facility (hands-on)
- NIDES utility programs (hands-on)
- NIDES upcoming events
- Questions & answers

Test Facility (Hands-On)

Test Facility Exercises

- Create audit data archive
- Create “real” and “virtual” audit data sets
- Create test instances
- Profile building test
- Statistics false-positive rate test
- Cross-profiling test
- Rulebase test
- Test status functions
- Test maintenance functions
(tests and instances)

Audit Data Archive Creation

- Convert native format data to NIDES format using audit2ia and acc2ia

```
audit2ia -bsm -i infile -o outfile.Z -host myhost  
acc2ia -i pacct -o outfile.Z -host myhost
```

- Merge files as needed using iamerge

```
iamerge -i1 file1.Z -i2 file2.Z -o merged-file.Z
```

Audit Data Archive Creation Continued

- Process NIDES data file through archiver

```
archiver -i merged-data.Z -o archive-name
```

- Review data via Browse Menu Audit Data option
 - Search criteria are archive name, subject list, and time range
 - Selection of one of eight view options initiates retrieval

Audit Data Browse Window

-- NIDES -- Audit Data Browse Window						
ARCHIVE SELECTION	SUBJECT SELECTION	TIME RANGE SELECTION				
<p>archive_1 archive_2 archive_3 archive_4 archive_5 archive_6</p> <p>Current Selection: archive_3 Number of Records: 568790</p>	<table border="1"><thead><tr><th>Available Subjects</th><th>Subjects to display</th></tr></thead><tbody><tr><td>root user_1 user_5 sys_admin tmp_user admin_user user_16</td><td>ides user_3</td></tr></tbody></table> <p>Subject Options: <input type="button" value="Clear"/> <input type="button" value="All"/></p>	Available Subjects	Subjects to display	root user_1 user_5 sys_admin tmp_user admin_user user_16	ides user_3	<p>From 06/28/93 00:05:02</p> <p>To 07/31/93 23:58:41</p>
Available Subjects	Subjects to display					
root user_1 user_5 sys_admin tmp_user admin_user user_16	ides user_3					
RETRIEVED RECORD COUNT:						
<p>< Data Area ></p>						
<p>View Options: <input type="button" value="Basic"/> <input type="button" value="System"/> <input type="button" value="Host"/> <input type="button" value="User"/> <input type="button" value="Resource"/> <input type="button" value="File"/> <input type="button" value="Misc"/> <input type="button" value="All"/></p> <p><input type="button" value="Done"/> <input type="button" value="SaveToFile"/> <input type="button" value="HELP"/></p>						

Audit Data Browse Working Window

-**- NIDES -** Audit Data Browse Window		
ARCHIVE SELECTION	SUBJECT SELECTION	TIME RANGE SELECTION
archive_1 archive_2 archive_3 archive_4 archive_5 archive_6 Current Selection: archive_3 Number of Records: 568790	Available Subjects Subjects to display root ides user_1 user_3 user_5 sys_admin tmp_user admin_user user_16 Subject Options: <input type="button" value="Clear"/> <input type="button" value="All"/>	From 06/28/93 00:05:02 To 07/31/93 23:58:41
RETRIEVED RECORD COUNT:		
<input checked="" type="checkbox"/> Number of records selected: 1000/4624... <input type="button" value="Stop Retrieval"/>		
< Data Area >		
View Options: <input type="button" value="Basic"/> <input type="button" value="System"/> <input type="button" value="Host"/> <input type="button" value="User"/> <input type="button" value="Resource"/> <input type="button" value="File"/> <input type="button" value="Misc"/> <input type="button" value="All"/>		
<input type="button" value="Done"/> <input type="button" value="SaveToFile"/>	<input type="button" value="HELP"/>	

Audit Data Set Creation

- Create audit data set using `adset_index` utility (“real” data set)
 - Create **NIDES** audit data file (`audit2ia`, `acc2ia`, `iamerge`)
 - Place file in `$IDES_ROOT/storage/adsets` directory
 - Create index for file using `adset_index`

```
adset_index -i input-file -v
```

Audit Data Set Creation Continued

- Create *‘virtual’ audit data set via Customize Menu Audit Data Sets Option
 - Select archive source
 - Select create option and enter audit data set name
 - Specify search criteria (subjects and time range)
 - Select virtual option ‘DMFindex’

Audit Data Set Management Window

-- NIDES -- Audit Data Set Management Window		
AUDIT DATA SETS	SUBJECTS IN DATA SET	TIME RANGE OF DATA SET
anntest.Z oolong.Z nsa_data.Z make_demo_data.Z abc_archive.Z safeguard1.Z rb_demo.data.Z sep92.Z july92.Z aug92.Z	debra luntzel neumann hogan teo tamaru lunt gilham jagan caveh	From <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;">06/28/93 00:05:02</div> To <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;">07/31/93 23:58:41</div>
Current selection: july92.Z Number of records: 201124		
Data set options: <div style="display: inline-block; margin-left: 10px;"> <input type="button" value="New"/> <input type="button" value="Delete"/> </div>		
<input type="button" value="Done"/>		<input type="button" value="HELP"/>

Audit Data Create Window

<p align="center"> *** NIDES *** Create Audit Data Set Window DATA SET NAME: test_adset </p>						
Available Audit Data Archives	Subject Selection	Time Range Selection				
<p> real-time A1-small_archive A2-medium_archive A3-large_archive A4-Xlarge_archive </p> <p> Current archive: A3-large_archive </p> <p> Number of records: 201124 </p>	<table border="1"> <thead> <tr> <th>Available Subjects</th> <th>Subjects to Filter</th> </tr> </thead> <tbody> <tr> <td> caveh debra gilham hogan jagan lunt luntzel neumann root tamaru </td> <td></td> </tr> </tbody> </table> <p> Subject options: <input type="button" value="Clear"/> <input type="button" value="All"/> </p>	Available Subjects	Subjects to Filter	caveh debra gilham hogan jagan lunt luntzel neumann root tamaru		<p> Available time range: 06/28/92 00:05:02 07/31/92 23:58:41 </p> <p> From <input type="text" value="06/28/94 00:15:02"/> </p> <p> to <input type="text" value="07/31/92 23:58:41"/> </p>
Available Subjects	Subjects to Filter					
caveh debra gilham hogan jagan lunt luntzel neumann root tamaru						
<input type="button" value="ADsetFile"/> <input type="button" value="DMFindex"/> <input type="button" value="Cancel"/>		<input type="button" value="HELP"/>				

Instance Management

- Baseline instance (profile building tests)
 - Create instance using Customize Menu Test Instances option (New)
 - Configure profile cache size and any other desired options
- False-positive test instance
 - Copy existing instance containing trained profiles
 - Configure profile updating OFF
 - Configure result filter to “Warning and Above” level
 - Turn OFF alert generating rule groups

Instance Management Continued

┆ Cross-profiling test instance

- Copy existing instance containing trained profiles
- Configure profile updating OFF
- Select subject for cross-profiling
- Replace all subjects' profiles with selected profile
- Configure result filter to "Warning and Above" level
- Turn OFF alert generating rule groups

Instance Management Continued

- Rulebase test instance
 - Create default instance
 - Turn OFF all alert generating rule groups except rules to be tested
 - Turn ON all rule group members needed for test

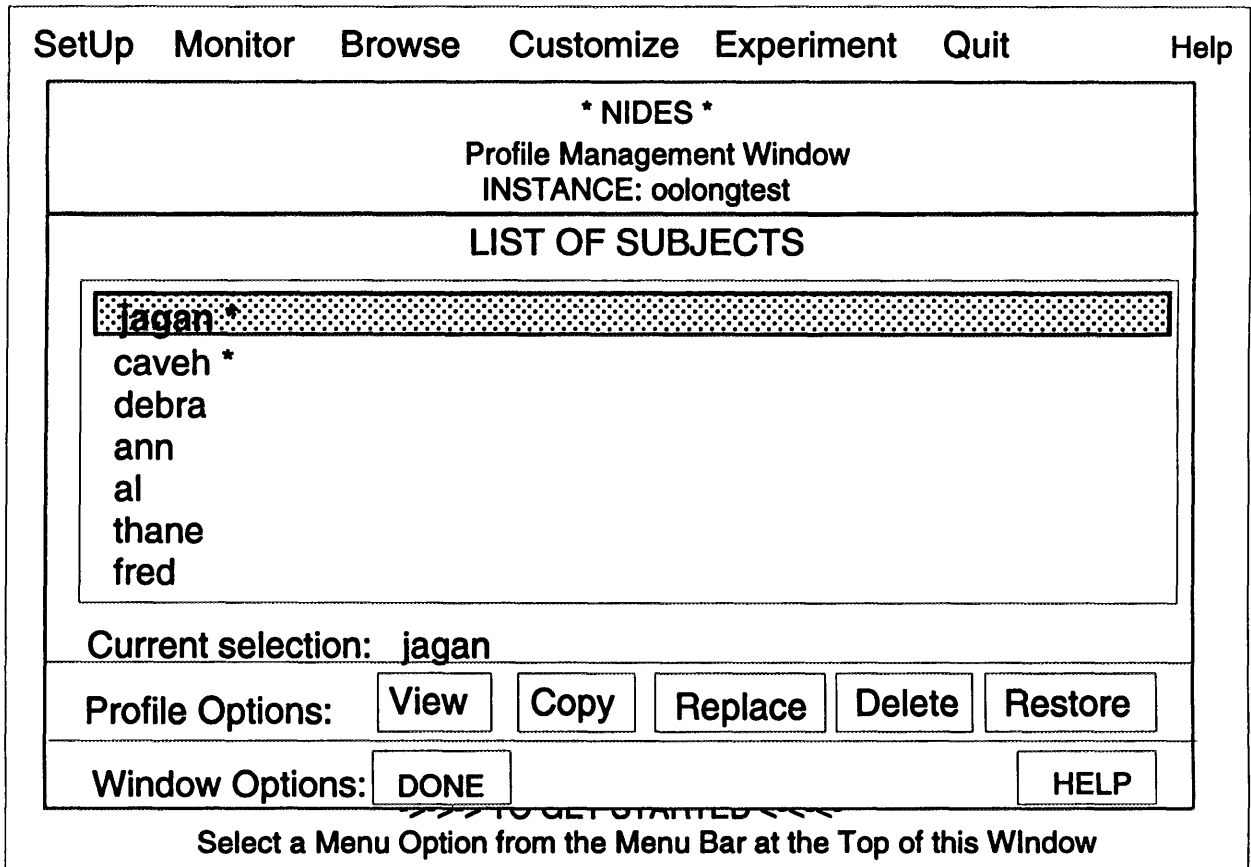
Instance Management Window

- NIDES -	
Instance Management Window	
- Instances -	
<div style="border: 1px solid black; padding: 5px;"><p>real-time</p><p>manual-instance</p><p>test1</p><p>test2</p><p>test3</p></div>	
Current Selection: manual-instance	
<input type="button" value="New"/>	<input type="button" value="Modify"/>
<input type="button" value="Copy"/>	<input type="button" value="Delete"/>
<input type="button" value="SaveToFile"/>	<input type="button" value="Done"/>
<input type="button" value="HELP"/>	

Profile Building Test

- Create baseline instance
- Create audit data set with minimum 1 month of data — 2 months even better
- Verify profile updating is ON
- Execute test
- Review profiles to confirm they are trained

Profile Management Window



Profile View Window

Measure Status

--- NIDES ---
Profile View Window
INSTANCE: real-time SUBJECT: root

Last Profile Update: Thu Apr 21 01:00:00 1994 Number of Profile Updates: 34
 Last Audit Record Timestamp: Thu Apr 21 01:00:00 1994

PROFILE ITEM	Measure Status																																												
<div style="background-color: #cccccc; padding: 2px;">Measure Status</div> Measure Misc Info Categories Q & S values Q distribution table Tails of Q dist'n table Daily Q bin counts T2 distribution table T2 counts (daily) Misc profile data	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: right;">Number of updates:</td> <td style="text-align: right;">34</td> </tr> <tr> <td style="text-align: right;">Number of active measures:</td> <td style="text-align: right;">12</td> </tr> <tr> <td style="text-align: right;">Aged Number of active measures:</td> <td style="text-align: right;">12.0001</td> </tr> </table> <table style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th rowspan="2" style="text-align: left;">MEASURE</th> <th rowspan="2" style="text-align: left;">STATUS</th> <th colspan="3" style="text-align: right;">TRAINING STATUS</th> </tr> <tr> <th style="text-align: right;">ToGo</th> <th style="text-align: right;">Phase</th> <th style="text-align: right;">Effn</th> </tr> </thead> <tbody> <tr> <td>CPU</td> <td>*ON/READY</td> <td style="text-align: right;">0</td> <td></td> <td style="text-align: right;">53494.11</td> </tr> <tr> <td>IO</td> <td>*ON/READY</td> <td style="text-align: right;">0</td> <td></td> <td style="text-align: right;">48238.55</td> </tr> <tr> <td>MEM</td> <td>*ON/READY</td> <td style="text-align: right;">0</td> <td></td> <td style="text-align: right;">54369.24</td> </tr> <tr> <td>LOC</td> <td>OFF/READY</td> <td style="text-align: right;">0</td> <td></td> <td style="text-align: right;">54653.86</td> </tr> <tr> <td>MAIL</td> <td>ON/READY</td> <td style="text-align: right;">0</td> <td></td> <td style="text-align: right;">16828.39</td> </tr> <tr> <td>EDIT</td> <td>ON/TRAINING</td> <td style="text-align: right;">2</td> <td style="text-align: right;">CQT</td> <td style="text-align: right;">35.07</td> </tr> </tbody> </table>	Number of updates:	34	Number of active measures:	12	Aged Number of active measures:	12.0001	MEASURE	STATUS	TRAINING STATUS			ToGo	Phase	Effn	CPU	*ON/READY	0		53494.11	IO	*ON/READY	0		48238.55	MEM	*ON/READY	0		54369.24	LOC	OFF/READY	0		54653.86	MAIL	ON/READY	0		16828.39	EDIT	ON/TRAINING	2	CQT	35.07
Number of updates:	34																																												
Number of active measures:	12																																												
Aged Number of active measures:	12.0001																																												
MEASURE	STATUS	TRAINING STATUS																																											
		ToGo	Phase	Effn																																									
CPU	*ON/READY	0		53494.11																																									
IO	*ON/READY	0		48238.55																																									
MEM	*ON/READY	0		54369.24																																									
LOC	OFF/READY	0		54653.86																																									
MAIL	ON/READY	0		16828.39																																									
EDIT	ON/TRAINING	2	CQT	35.07																																									

SaveToFile
Done
HELP

Statistics False-positive Rate Test

- | Create false-positive rate test instance using baseline instance
- Select audit data set not used to train profiles (i.e., not used for profile building test)
- Execute test

Statistics False-positive Rate Test Continued

- Calculate false-positive rates for red and yellow thresholds
 - Bring up test result window and select test

Red false-positive rate =
Critical level stat results / total results

Yellow false-positive rate =
Warning level stat results / total results

Test Result Window

*** NIDES *** Analysis Results View Window																																										
Test Instance Selection	Subject Selection	Time Range Selection																																								
test1 test2 test3	<table border="1"> <tr> <th>Avail Subjects</th> <th>Subjects to display</th> </tr> <tr> <td>caveh</td> <td>joe</td> </tr> <tr> <td>debra</td> <td></td> </tr> <tr> <td>hogan</td> <td></td> </tr> <tr> <td>root</td> <td></td> </tr> <tr> <td>teo</td> <td></td> </tr> </table>	Avail Subjects	Subjects to display	caveh	joe	debra		hogan		root		teo		From 06/28/92 00:15:02 to 07/31/92 23:58:41																												
Avail Subjects	Subjects to display																																									
caveh	joe																																									
debra																																										
hogan																																										
root																																										
teo																																										
Current test: test2	Subject options: <input type="button" value="Clear"/> <input type="button" value="All"/>																																									
TEST INSTANCE NAME: test2		TIME STARTED: 04/21/94 15:53:42																																								
AUDIT DATA SET:		TIME FINISHED: 04/21/94 17:11:30																																								
-* RECORD COUNTS -*																																										
	ALERTS	CRITICAL																																								
	80	101																																								
	80	101																																								
	122	122																																								
	200895	0																																								
	201118	223																																								
	SAFE	TOTAL																																								
Processed:																																										
Archived																																										
NUM. OF RECORDS: 201118		NUM. OF ALERTS: 80																																								
<table border="1"> <thead> <tr> <th>SUBJ @ HOSTNAME</th> <th>TIMESTAMP</th> <th>AUDREC#</th> <th>SCORE (RED THRESH)</th> <th>TOP 5 MEAS (TOP 5 S-VALUES)</th> </tr> </thead> <tbody> <tr> <td>tamaru @ oolong</td> <td>12/18/92 10:42:35</td> <td>14047</td> <td>0.0000 (0.0000)</td> <td>COMMD HOUR ARECDIST INT60 INT600 (0.000</td> </tr> <tr> <td>tamaru @ oolong</td> <td>12/18/94 10:42:35</td> <td>14048</td> <td>0.0000 (0.0000)</td> <td>COMMD HOUR ARECDIST INT60 INT600 (0.000</td> </tr> <tr> <td>tamaru @ oolong</td> <td>12/18/94 10:42:35</td> <td>14049</td> <td>0.0000 (0.0000)</td> <td>COMMD HOUR ARECDIST INT60 INT600 (0.000</td> </tr> <tr> <td>tamaru @ oolong</td> <td>12/18/94 10:42:35</td> <td>14050</td> <td>0.0000 (0.0000)</td> <td>COMMD HOUR ARECDIST INT60 INT600 (0.000)</td> </tr> <tr> <td>tamaru @ oolong</td> <td>12/18/94 10:42:35</td> <td>14051</td> <td>0.0000 (0.0000)</td> <td>COMMD HOUR ARECDIST INT60 INT600 (0.000)</td> </tr> <tr> <td>tamaru @ oolong</td> <td>12/18/94 10:42:35</td> <td>14052</td> <td>0.0000 (0.0000)</td> <td>COMMD HOUR ARECDIST INT60 INT600 (0.000)</td> </tr> <tr> <td>tamaru @ oolong</td> <td>12/18/94 10:42:35</td> <td>14053</td> <td>0.0000 (0.0000)</td> <td>INT60 COMMD HOUR ARECDIST INT600 (0.000)</td> </tr> </tbody> </table>			SUBJ @ HOSTNAME	TIMESTAMP	AUDREC#	SCORE (RED THRESH)	TOP 5 MEAS (TOP 5 S-VALUES)	tamaru @ oolong	12/18/92 10:42:35	14047	0.0000 (0.0000)	COMMD HOUR ARECDIST INT60 INT600 (0.000	tamaru @ oolong	12/18/94 10:42:35	14048	0.0000 (0.0000)	COMMD HOUR ARECDIST INT60 INT600 (0.000	tamaru @ oolong	12/18/94 10:42:35	14049	0.0000 (0.0000)	COMMD HOUR ARECDIST INT60 INT600 (0.000	tamaru @ oolong	12/18/94 10:42:35	14050	0.0000 (0.0000)	COMMD HOUR ARECDIST INT60 INT600 (0.000)	tamaru @ oolong	12/18/94 10:42:35	14051	0.0000 (0.0000)	COMMD HOUR ARECDIST INT60 INT600 (0.000)	tamaru @ oolong	12/18/94 10:42:35	14052	0.0000 (0.0000)	COMMD HOUR ARECDIST INT60 INT600 (0.000)	tamaru @ oolong	12/18/94 10:42:35	14053	0.0000 (0.0000)	INT60 COMMD HOUR ARECDIST INT600 (0.000)
SUBJ @ HOSTNAME	TIMESTAMP	AUDREC#	SCORE (RED THRESH)	TOP 5 MEAS (TOP 5 S-VALUES)																																						
tamaru @ oolong	12/18/92 10:42:35	14047	0.0000 (0.0000)	COMMD HOUR ARECDIST INT60 INT600 (0.000																																						
tamaru @ oolong	12/18/94 10:42:35	14048	0.0000 (0.0000)	COMMD HOUR ARECDIST INT60 INT600 (0.000																																						
tamaru @ oolong	12/18/94 10:42:35	14049	0.0000 (0.0000)	COMMD HOUR ARECDIST INT60 INT600 (0.000																																						
tamaru @ oolong	12/18/94 10:42:35	14050	0.0000 (0.0000)	COMMD HOUR ARECDIST INT60 INT600 (0.000)																																						
tamaru @ oolong	12/18/94 10:42:35	14051	0.0000 (0.0000)	COMMD HOUR ARECDIST INT60 INT600 (0.000)																																						
tamaru @ oolong	12/18/94 10:42:35	14052	0.0000 (0.0000)	COMMD HOUR ARECDIST INT60 INT600 (0.000)																																						
tamaru @ oolong	12/18/94 10:42:35	14053	0.0000 (0.0000)	INT60 COMMD HOUR ARECDIST INT600 (0.000)																																						
View options: <input type="button" value="StatAlerts"/> <input type="button" value="RBAAlerts"/> <input type="button" value="AllAlerts"/> <input type="button" value="AllResults"/>																																										
<input type="button" value="Done"/>	<input type="button" value="SaveToFile:"/>	<input type="button" value="HELP"/>																																								

Cross-profiling Test

- Create cross-profiling test instance using baseline instance
- Select audit data set not used to train profiles (i.e., not used for profile building test)
- Execute test

Cross-profiling Test Continued

- Calculate detection rates for red and yellow thresholds
 - Bring up test result window and select test

Red detection rate =
Critical level stat results / total results

Yellow detection rate =
Warning level stat results / total results

Rulebase Test

- Create rulebase test instance
- Select audit data set containing rules scenario
- Execute test
- Review results

Profile Viewing

- View profiles using Browse Menu Instances option
- Review training status via “Measures” Option
- Review subject categories, particularly files and directories for potential tmp file filter candidates

Test Status Functions

- Review active test status
 - Experiment Menu Status & Results option
- Review test results
 - Specify subjects and time range
 - Four view options (RBAAlerts, StatAlerts, AllAlerts, and AllResults)
 - Selection of view options initiates retrieval

Test Status Window

-- NIDES --				
Test Status/Results Window				
Tests Running				
Test Instance Name	Audit Data Set	# Records	# Alerts	Time Started
rulebase_test1	audit_data_set_3	10890	10	Thu Apr 21 15:45:41 1994
instance_1	audit_data_set_7	450	0	Thu Apr 21 16:25:33 1994
Tests Completed				
Test Instance Name	Audit Data Set	# Records	# Alerts	Time Completed
stats_test1	audit_data_set_5	556678	100	Fri Apr 15 21:45:41 1994
profile_build_test1	audit_data_set_9	7700	0	Fri Apr 1 02:25:33 1994
prof_test1	audit_data_set_8	1234567	10101	Mon Jan 3 18:12:45 1994
Current Selection: prof_test1				
<input type="button" value="View Results"/> <input type="button" value="Delete Test"/> <input type="button" value="Done"/>				<input type="button" value="HELP"/>

Test Maintenance Functions

- Test deletion via Test Status Window
 - Removes test results only
 - Useful after profile-building test completed
 - Saves disk space
- Instance deletion via Customize Menu
 - Removes test results and instance (profiles and configuration)
 - Useful when results and profiles no longer needed
 - Instances should be deleted when no longer needed
 - Conserves disk space

Utility Programs (Hands On)

Utility Programs Exercises

- acc2ia
- adset_index
- apstat
- archiver
- audit2ia
- batch_analysis
- iamerge
- iapr
- init_priv_user_list
- init_stat_config

NIDES Upcoming Events

Events

- Updated release available in October
 - Bug fixes
 - Performance enhancements
 - Minor feature enhancements
 - Updated rulebase
 - Customizable agen written in PERL
- Additional training course
- Users encouraged to report bugs and recommend enhancements/changes to the **NIDES** software, documentation, and training
- Request course attendees provide feedback by completion of course evaluation survey

Questions & Answers

Worksheets

Default Rulebase Configuration Table							
Instance:							
Date:							
RULEBASE CONFIGURATION							
Rule Name	Def	Config		Rule Name	Def	Config	
		ON	OFF			ON	OFF
AccessPrivateDevice	ON			AccessPrivateFile1	ON		
AccessPrivateFile2	ON			AccessSpecialFile	ON		
BackwardsTime	ON			BadLogin1	ON		
BadLogin2	ON			BadLoginAnomaly	ON		
BadLoginBadPassword	ON			BadPassword1	ON		
BadPassword2	ON			BadPasswordAnomaly	ON		
BadRoot	ON			BadUserExec	ON		
BrokeRoot	ON			ChangeLoginFile	ON		
ChmodOtherUser	ON			ChmodSystemFile	ON		
ClearParanoidUser	ON			ClearSession	ON		
ConsoleLogin	ON			DialInLogin	ON		
DotFile	ON			Exec	ON		
FTPAnomaly	ON			FlagRSH	ON		
GoodLogin1	ON			GoodLogin2	ON		
GoodPassword1	ON			GoodPassword2	ON		
GoodSU1	ON			GoodSU2	ON		
InvisibleDirectory	ON			KnownLogin1	ON		
Leapfrog1	ON			LinkSystemExec	ON		
LocalLogin	ON			Logout1	ON		
Logout2	ON			ModSystemExec	ON		
MultiLogin1	ON			MultiLogin2	ON		
				NoRemote	ON		
ParanoidUser1	ON			ParanoidUser3	ON		
ParanoidUserAnom	ON			PasswordFileAccess	ON		
ReadSystemExec	ON			RemoteExec	ON		
RemoteFile1	ON			RemoteFile2	ON		
RemoteFile3	ON			RemoteLogin	ON		
RemoteMount1	ON			RemoteMount2	ON		
RemoteRootBadLogin	ON			RemoteRootBadPassword	ON		
RunsRareExec	ON			SpecUserExec	ON		
Su1	ON			SuspiciousUser	ON		
TFTPAnomaly	ON			TFTPUse	ON		
TouchSession	ON			TrojanHorse	ON		
TruncateLog	ON						

Rule Group Worksheet

Scenario Description:								
Rule Name:					Priority:			
Description:								
Anomaly?: YES or NO								
Facts					Marks			
Exist	Absent	Assert	Delete	Modify	Exist	Absent	Apply	Remove
Rule Name:					Priority:			
Description:								
Anomaly?: YES or NO								
Facts					Marks			
Exist	Absent	Assert	Delete	Modify	Exist	Absent	Apply	Remove
Rule Name:					Priority:			
Description:								
Anomaly?: YES or NO								
Facts					Marks			
Exist	Absent	Assert	Delete	Modify	Exist	Absent	Apply	Remove

rb_config File Worksheet (Part 1)

Section	Additions	Deletions
DOMAIN		
GENERIC_CONFIG		
HOME_DIR		
KNOWN_LOGIN		
LOG_DIR		
LOGIN_CONFIG		
NOEXEC		
PARANOID_PROG		

rb_config File Worksheet (Part 2)

Section	Additions	Deletions
PRIVATE_DEVICE		
PRIVATE_FILE		
PROGLOCATION		
PROGRAM		
RAREEXEC		
REMOTE_FILE_NO_ACCESS		
REMOTE_FILE_NO_MODIFY		
REMOTE_NO_EXEC		
REMOTE_NOT_OK		

rb_config File Worksheet (Part 3)

Section	Additions	Deletions
ROOT_OK		
SPECIAL_FILE		
SPECIAL_PROGRAM		
SPECIAL_USER		
SYSTEM_SCRIPTS		
TMP_DIRNAME		
TMP_FILE		
USER_TYPE		

Class Configuration Worksheet

Instance:			
Date:			
COMPILERS		EDITORS	
Add	Delete	Add	Delete
MAILERS		SHELL ENVIRONMENTS	
Add	Delete	Add	Delete
NETWORK COMMANDS		LOCAL HOSTS	
Add	Delete	Add	Delete
TMP FILES		WINDOW COMMANDS	
Add	Delete	Add	Delete

Measure Configuration Worksheet (part 1)

Instance:										
Date:										
Measure CONFIGURATION										
Measure	Status		Qmax		Scalar		Min eff-n		H-life	
	def	config	def	config	def	config	def	config	def	config
U_CPU	ON		500		1000		100		100	
U_IO	ON		500		10000000		100		100	
U_MEM	ON		500		10000000		100		100	
U_LOC	OFF		1500		-		100		100	
U_MAIL	ON		100		-		100		100	
U_EDIT	ON		500		-		100		100	
U_COMPILER	OFF		500		-		100		100	
U_SHELL	OFF		500		-		100		100	
U_WINDOW	OFF		500		-		100		100	
U_COMMD	ON		500		-		100		100	
U_COMMDB	OFF		500		-		100		100	
U_COMMDC	ON		500		-		100		100	
U_SYSCALL	OFF		500		-		100		100	
U_DIR	OFF		500		-		100		100	
U_DIRB	OFF		500		-		100		100	
U_DIRNEW	OFF		500		-		100		100	
U_DIRDEL	OFF		500		-		100		100	
U_DIRMOD	OFF		500		-		100		100	
U_DIRREAD	OFF		500		-		100		100	
U_FILENEW	OFF		500		-		100		100	
U_FILEREAD	OFF		500		-		100		100	
U_FILEMOD	OFF		500		-		100		100	
U_FILEDEL	OFF		500		-		100		100	
U_FILETMP	OFF		500		-		100		100	
U_FILE	OFF		500		-		100		100	
U_FILEB	OFF		500		-		100		100	
U_UID	OFF		500		-		100		100	
U_UIDB	ON		500		-		100		100	

Measure Configuration Worksheet (part 2)

Instance:										
Date:										
Measure	Status		Qmax		Scalar		Min eff-n		H-life	
	def	config	def	config	def	config	def	config	def	config
U_SYSERR	ON		500		-		100		100	
U_SYSERRTYP	OFF		500		-		100		100	
U_AUDREC	OFF		500		-		100		100	
U_HOUR	ON		1000		-		100		100	
U_HOURB	OFF		500		-		100		100	
U_DAILY	OFF		10000		-		100		100	
U_DAILYB	OFF		500		-		100		100	
U_RNET	ON		500		-		100		100	
U_RNETTYP	OFF		500		-		100		100	
U_RNETHOST	OFF		500		-		100		100	
U_LNET	OFF		500		-		100		100	
U_LNETTYP	OFF		500		-		100		100	
U_LNETHOST	OFF		500		-		100		100	
U_INTARR	ON		500		172800		100		100	
U_FCLASS	OFF		500		-		100		100	
U_FCLSRD	OFF		500		-		100		100	
U_FCLSWR	OFF		500		-		100		100	
U_ARECDIST	ON		500		-		100		100	
U_INT60	ON		1000		1		100		100	
U_INT600	ON		2000		1		100		100	
U_INT3600	ON		5000		1		100		100	

Statistics Parameters Configuration Worksheet

Instance:		
Date:		
Statistics Parameters CONFIGURATION		
Configuration Item	Default Value	Configured Value
Long-term profile Half-life	20	
Training Period	20	
Red Threshold	0.10	
Yellow Threshold	1.0	
Max Sum Rare Prob	0.10	
Profile Cache Size	5	

Test Information Worksheet

Test Instance:	
Test Type:	
Audit Data Set:	
Profile Updater:	ON or OFF
Profile Synchronization:	ON or OFF
Time Started:	
Time Completed:	
Alert Count:	
RBAAlert Count:	
StatAlert Count:	
Total Records Processed:	
Critical-level Results Generated:	
Warning-level Results Generated:	
Safe-level Results Generated:	
Critical-level Detection Rate: (Critical-level/Total Records):	
Warning-level Detection Rate: (Warning-level/Total Records):	

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

Glossary

© 2000 by the American Psychological Association

Glossary

Accounting Audit Data The standard UNIX accounting system. Designed primarily for keeping track of resource utilization (e.g., connection time, CPU usage) for billing purposes. The accounting records generated are of minimal utility when other forms of audit data are available (e.g., C2 or BSM).

Activity Intensity Measure A group of measures that capture intensity of activity measured in rate of arrival of audit records. Three measures track intensity over the last minute, ten minutes, and hour, comparing the rates observed in real time to the rates as learned in the profile. These are intended to detect intrusions that flood the system with audit records.

Activity Vector Each time the NIDES Statistical Analysis component analyzes an audit record, the first processing step is the construction of an activity vector. This vector of observed measure values (at most one per NIDES measure) is obtained by processing the data contained in the NIDES audit record. For every measure represented in the audit record, the associated audit data is converted to a continuous or categorical value, depending on the type of measure, and placed in the activity vector entry for the measure.

Adset A mnemonic term for Audit Data Set. See Audit Data Set.

Aging Factor The factor by which past data is multiplied so as to fade its value at a desired rate. For a half-life of k audit records, for example, the factor is set at the k th root of $1/2$, so that after k steps the data are faded to one-half of their original contribution. Storing profiles as aged cumulative totals permits relatively compact profile structures and allows the system to adapt to changes in subject behavior. NIDES has a short-term aging factor applied to each audit record and a long-term aging factor applied to daily totals at update time.

Agen One of the core NIDES processes. A single agen process runs on each of the actively monitored target hosts, translating all the supported, native audit data into canonical NIDES audit records, and providing them to the arpool process. The UNIX version of the agen process currently supports three native audit record formats: SunOS BSM version 1, SunOS C2, and standard UNIX accounting.

Alert NIDES has two analysis components that process audit data and determine if a suspicious event has occurred - rulebased and statistics. A resolver component takes the results of the rulebased and statistical analysis and determines if an alert should be reported. Currently, the resolver reports all rulebased results that are critical as alerts.

For the statistical analysis, when the T2 score as of the current audit record exceeds a declaration (red or critical) threshold and the previous audit record did not exceed the threshold, an alert is reported. The threshold is set to achieve a nominal false positive rate (user configurable, 0.1% by default). As the statistical analysis employs a short-term memory of recent activity, an alert occurs on the record that nudges the score above the threshold, but the alert should be considered as reflecting a sequence of unusual activity in the recent past. If subsequent audit records keep the statistical score above the threshold, additional alerts are not reported unless the top (most significant) measure that contributed to the score changes.

Antecedent See Rule Antecedent.

Arpool One of the core NIDES processes. The arpool process accepts canonical NIDES audit records from the agen process on all the actively monitored target hosts and presents the audit records as a single data stream to the analysis components of NIDES.

Archiver One of the core NIDES processes. The archiver process accepts canonical NIDES audit records from the arpool process and stores them on disk, in a compressed format, to facilitate future reference when investigating activity that generated alerts.

Audit Data Set A source of NIDES audit records, generally used as input to run NIDES experiments using the test facility. An audit data set can be either real or *virtual*. A real audit data set consists of a single UNIX file (usually compressed) containing NIDES audit records. A virtual audit data set consists of parameters used to select audit data from an audit data archive; the audit data is retrieved from the specified audit data archive at the time a test is run.

Audit Record Distribution Measure A special measure whose categories are the names of all other measures and which tracks the number of times the respective

measures are touched in the short-term profile. Its purpose is to assess the normalcy of the distribution of the users recent activity across the measures.

Audit Record Half-life See short-term half-life.

Bin Table entry to which an observed value is assigned. For categorical measures, such as ERRYP, there is a one-to-one correspondence between bins and observed category values. For continuous measures there are 32 bins which correspond to value ranges.

Binary Measure A group of measures that track whether or not a given type of activity is observed in the current audit record. Binary measures are used as a mechanism to maintain counts in the audit record distribution measure and do not directly affect the score.

BSM The most recent auditing system developed for SunOS. The BSM (Basic Security Module) generates audit records derived from low-level UNIX activity (e.g., reading, writing, assessing, or deleting a file, changing directory, running a program).

Categorical Measure A measure that assumes values in discrete categories. For some such measures, such as HOUR, the values are known beforehand (the hours 0, 1, 2, . . . , 23). For others, new categories are allocated by NIDES as they are encountered.

Category An observed value (such as error type or hour of use on a 24-hour clock) for categorical measures, or a value range for a continuous measure such as CPU. By logarithmically recoding the ranges of continuous measures, NIDES in fact treats all measures as categorical.

Class A list of commands or objects belonging to the same class of activity (e.g., compilers, editors, or mail commands). Classes are used by the statistical analysis component to determine categories for class measures. The classes used in NIDES are: compilers, editors, mail programs, shell environments, window commands, network commands, local hosts, and temporary file directories.

Class Measure A measure with a predefined set of categories that captures a given class of computer activity. For example, the compiler measure has as its predefined categories the various compilers available on the system. The profile for

this measure tracks the percent of compiler usage attributable to each compiler. This is useful because, for example, compiler usage may comprise a relatively small percentage of total command usage (and hence be somewhat diluted in the command usage measure) but may be especially interesting with respect to intrusion detection.

Consequent See Rule Consequent.

Continuous Measure A measure that takes continuous values, such as CPU in time units.

Cross-profiling An experiment in which data for each subject is tested against the trained profile for each other subject. Long-term profile update is disabled for such experiments.

C2 An older, now obsolete, auditing system developed for SunOS. C2 generates audit records derived from low-level UNIX activity (e.g., reading, writing, assessing, or deleting a file, changing directory, running a program). Its name is derived from a specific security rating described in the Orange Book. It should not be confused with the generic computer security rating of C2.

Detection/Detection Rate A declaration by NIDES that a stream of audit data contains anomalous activity, which can be at a yellow (caution) or red (critical) threshold. Detection rate is the percent of audit records in a given audit data stream that trigger detections.

Effective n The effective length of the short-term profile, which equals the series sum of all powers of the aging factor (or approximately 1.5 times the short-term half-life). This can be thought of as the number of audit records that, after aging, still make a contribution to the short-term profile.

Experiment See Test.

Fact The NIDES rulebased component stores transitory information needed for its analysis in facts. Facts are stored in a database (see Factbase) internal to the rulebased component. The rulebase can define many different kinds of facts. The structures for facts are defined by ptype declarations. Facts are asserted (added) and removed from the internal database by rules during runtime.

Factbase A database of transitory information (See Fact) created, used, and maintained by the NIDES rulebased analysis component. Multiple facts of the same type can be contained in the factbase. If a rule searches the factbase for a fact type that contains multiple entries, the most recently asserted fact matching the rule search specification will be returned to the rule.

False-positive A detection, by the statistical analysis component, for a subject against its own profile.

Half-life The number of audit records (in the case of the short-term profile) or the number of profile updates (in the case of the historical profile) by which time the contribution of a data item to the present cumulative totals is reduced by one half.

Historical effective n The effective count of audit records contributing to the long-term profile. It consists of the sum of all daily totals each weighted by the appropriate power of the long-term aging factor. This value can be thought of as the number of audit records that, after aging, still contribute to the long-term profile.

Historical Profile See Long-term Profile.

IDES_ROOT The NIDES environment variable that determines the directory where the NIDES software resides. This variable must be set prior to running any NIDES software.

Instance An analysis configuration, and the set of profiles associated with that configuration.

Intensity Measure See Activity Intensity Measure.

Inter-arrival Time The difference in timestamps between successive audit records for the same subject. Used by the statistical analysis to monitor intensity (rate of activity in 1-minute, 10-minute, and 60-minute windows) and thereby potentially detect an intrusion that floods the system with audit records.

Long-term Half-life That time interval (measured in profile updates) by which time the contribution of a given data item in the long-term profile is aged out by a factor of one-half. The system default is 20 updates (one month of nonweekend days), configurable by the user.

Long-term Profile For each subject and measure, the observed categories and the observed long-term probabilities for each category, the historical effective n , and the empirical Q distributions. For the subject there is also an empirical score (**T2**) distribution, which is aggregated across all measures. At the end of each day, this profile is aged by the long-term aging factor and combined with the new daily totals.

Max Sum of Rare Category Probabilities (Max Sum Rare Prob) A configurable constant that represents the maximum sum of probabilities of categories classified as rare. Categories are sorted in ascending order of probability and then summed to the largest index for which the sum is less than or equal to this constant. All categories up to and including this index are classified as rare until the next update interval. For numerical stability, this value should be between 0.01 and 0.05 .

Measure A measure is an aspect of subject behavior. This is the unit used by the statistical analysis component of NIDES. The measure is used to monitor activity on a particular dimension of subject behavior. Measure types are continuous (such as CPU in seconds on the present audit record), categorical (such as file name), intensity (rate of arrival of audit records in various time windows), and a special audit record distribution measure to monitor recent types of activity. A single audit record can generate observed values for more than one measure.

Minimum effective n The minimum count of records in the long-term profile that must be accumulated before the scoring mechanism is considered reliable. It is measure-specific.

Native Audit Record An audit record specific to a given auditing system. Native audit records are converted by the agen process into a canonical NIDES audit record format for analysis and storage. Once the audit data are converted, NIDES no longer makes use of a native audit record. The UNIX version of the agen process currently supports three native audit record formats: Sun OS BSM version 1, Sun OS C2, and standard UNIX accounting.

NIDES Audit Record A canonical audit record format capable of representing all supported native audit record information. NIDES audit records are used for analysis and storage. Once the audit data are converted, NIDES no longer makes use of a native audit record.

Orange Book The common name of a document describing different levels of computer security ratings and the associated requirements.

Persistent Storage NIDES maintains databases of many types under its normal operation. These databases include an audit record archive, analysis result archive, instances (user profiles and analysis configuration data) and miscellaneous configuration files (e.g., privileged user lists). All of these databases and files are part of the NIDES persistent storage facility. The persistent storage facility provides a set of library functions to all NIDES components, allowing them to read and write data to the various databases and configuration files.

Profile The statistical analysis component of NIDES generates a profile of behavior for each subject it sees in the audit data stream. The profile is comprised of two parts, a long-term profile and a short-term profile. The long-term profile contains the category probabilities, aged counts, system thresholds, and so forth for each subject, aged with a long-term half-life on the order of several weeks (set to achieve a trade-off between stability and adaptability to new behavior). The short-term profile contains the observed categories and aged counts in the recent past, aged with a short-term half-life of tens to hundreds of audit records (representing minutes to tens of minutes of activity). For computational efficiency, the short-term profile maintains aged counts, while the long-term profile maintains probabilities that do not change between updates.

Profile Snapshot An instantaneous view of the profile available immediately after an update or when a profile is swapped out of the profile cache and into persistent storage. The NIDES profile viewing utilities show the most recent snapshot.

Profile Synchronization A means of adjusting time stamps in experimental data sets that enables updating to take place in the test facility even when the time stamps in the audit data set are earlier than the last update time stamp in the profile.

Profile Training The general procedure of updating profiles, adding and dropping categories, and adjusting the empirical distributions for Q and T2. It proceeds in three stages. In the first, category probabilities are obtained from a number of days of raw data. In the second, the Q distribution is estimated over an additional number of days. Finally, the T2 distribution is estimated, after which

time NIDES is ready to score audit records. In a production environment, profile training continues indefinitely. For experimentation with known masquerader data, profile updating and training are disabled.

Profile update The merging of the historical profile with new information at the end of each day. Long-term probabilities are converted to effective counts (by multiplying by the historical effective n). The new daily counts are summed in, and the results converted back to probabilities. Categories that have too low a probability are folded into a RARE category, which can change daily.

ptype A declaration that defines the structure of facts that are created and stored in the NIDES rulebased components factbase. A ptype declaration is similar in concept to a structure declaration in C. An example of a ptype declaration is

```
ptype [event subject : string,  
      action:string,  
      object:string,  
      time:int]
```

Here the structure for the *event* ptype is defined to contain four fields: subject, action and object are strings, and time is an integer. Using this ptype, facts of type *event* can be added to or removed from the NIDES rulebased components factbase.

Q-score A chi-square-like square difference statistic based on the difference between the short- and long-term profiles for each measure.

QMax A scale value used to assign the Q-score into bins to obtain its empirical distribution.

Rare Probability A configurable system constant (default 0.01 or 1%) used for collapsing categories into a RARE class (which are scored by NIDES as a group rather than as individual categories). Categories whose cumulative sum is less than this constant are tagged as RARE in a given update.

Red/Critical threshold That value which, when exceeded by the T2 score, causes NIDES to issue a red or critical result from the statistical analysis. It is configurable (default of 0.1% seeks to achieve a false positive rate of 0.1% on normal data).

Remote Procedure Call (RPC) An action in which a process calls a procedure that is executed by another process. The NIDES architecture is composed of many processes that communicate via RPCs. For example, when the NIDES analysis components (statistical and rulebased) need an audit record to analyze, both components make an RPC to the arpool process to ask for the next audit record; the arpool process makes an RPC in the form of a response providing an audit record to the analysis processes.

Resolver The NIDES analysis process that receives results from the statistical and rulebased analysis components and determines if an alarm should be reported.

Result A result is generated for every audit record processed by the NIDES analysis components. Results are categorized into three levels: safe, warning, and critical. The level of a result is assigned by the resolver component based on the levels assigned by the statistical and rulebased analysis components. An NIDES alert is reported when the resolver determines that a critical-level result should be assigned alert status.

Rule Antecedent The first part of the two parts that comprise the body of a NIDES rule. The antecedent contains the tests that are performed on the rulebases factbase to determine if a particular condition is met. If the condition is met, the second part of the rule, the consequent, is executed.

Rule Consequent The second part of the two parts that comprise the body of a NIDES rule. The consequent contains a set of actions that are performed if the tests performed in the rules antecedent are satisfied. If the consequent actions are executed, the rule is said to have fired. Actions that may be performed in the consequent of a rule include additions or deletions to the rulebases factbase and generation of an alert report.

Rule Priority A priority assigned to the NIDES rulebased component rules when they are written. The priority determines the order in which rules are tested. Rules with higher priorities are tested first. Higher numbers equate to a higher priority (e.g., a priority of 5 is higher than a priority of 1).

S-value A unitless quantity obtained by inverting the observed Q-score using the Q empirical distribution and a half-normal transform. This results in all measure scores being comparably distributed.

Scalar A value used to scale observed (raw) values to assign them to category (range) bins.

Score The multivariate aggregate statistic on which the statistical analysis bases anomaly detection. Up to various normalizations, it is proportional to the sum of squares of the S values. Also called the T2 score.

Sequence Number Numbers assigned by the NIDES agen and arpool processes to the audit records processed by NIDES. Two sequence numbers are assigned to each audit record. The agen process assigns a target host sequence number that is unique for the duration of the current agen process execution on the target host. This number is referred to as the *target sequence number*. The arpool process assigns a sequence number to all audit records it receives; this number is unique across all NIDES target hosts and monotonically increases for the duration of the current arpool process. This number, referred to as the *audit record sequence number*, is used to identify the audit record when alerts are reported by NIDES. When arpool is first started it begins with a sequence number of 0.

Short-term Half-life See Half-life.

Short-term profile For each subject and measure, the number of counts recently observed for each category in the long-term profile with special handling for new categories. Due to the aging procedure, these counts are generally fractional.

Short-term Profile Length The effective number of audit records in the short-term profile. It is approximately 1.4 times the short-term half-life.

Subject The entity for which NIDES maintains profiles and performs anomaly detection. In the NIDES paradigm, the subject (e.g., a user of the system) initiates actions (e.g., file copy) that act on objects (e.g., files).

Subject Profile See Profile.

Target Host A host computer that is monitored (or can be monitored) by NIDES.

Test A batch run of NIDES with archived data, typically done to examine the impact of parameter changes or establish detection rates

Threshold The NIDES-estimated value for T2 at which a detection is declared. It is set to achieve no greater than some user-specified percent (usually 1% for yellow, 0.1% for red) of false positives.

Training The process by which the NIDES statistical component learns normal activity for a subject. It consists of category training (wherein the system learns the observed categories for each measure), Q training (wherein the system builds an empirical distribution for the Q statistic, which measures the measure-by-measure difference between the long- and short-term profiles), and T2 training (wherein the system establishes the threshold for the measure statistic, which is collected across all active measures). All three phases have a minimum training period before anomaly scoring begins. Training continues in the steady state, permitting a degree of adaptation to new subject behavior.

Training Status The status of a measure with respect to the three training phases (see Training). A measure can be trained (ready to contribute to scoring) or under one of the three phases.

Training Period The length of time (measured in number of profile updates) before measures may contribute to anomaly scoring. It is user configurable. A number of updates equal to one third this quantity (rounding any fraction upward to the next integer) is required before a measure exits each of the three training phases (see Training).

True-positive A detection for a subject (possibly a masquerader) against another subjects profile.

T2 The overall NIDES statistical analysis score on which anomalies are declared, aggregated across all measures. (See Score)

Yellow/Warning threshold That value which, when exceeded by the T2 score, causes NIDES to issue a yellow or warning alert from the statistical analysis. It is configurable (default of 1.0% seeks to achieve a false positive rate of 1.0% on normal data).