NMI End-to-End Diagnostic Advisory Group BoF
Outline

- Progress Report
  - Review of goals
  - Problem analysis from the perspective of the user
  - Survey
  - Architecture
    - Event data
    - Dissemination
- Present Efforts
- Timeline
- Conclusion
Review of Year One Goals

- Engage NMI-EDIT and GRIDS efforts
- Gather feature requirements from the perspective of the user, origin and target/application operators
- Be conscious of privacy feature requirements with respect to architecture, but defer implementation until year two
- Survey and report on market, research activities, and standards efforts in this space
  - Initially focus on simple solutions to quickly aid in the analysis and evolution of middleware diagnostic support
  - Create a modular architecture to prepare for a rich toolset in year two
Outline

• Progress Report
  • Review of goals
  • Problem analysis from the perspective of the user
  • Survey
  • Architecture
    • Event data
    • Dissemination

• Present Efforts

• Timeline

• Conclusion
Requirements Gathering

• Identify diagnosticians in specific areas
  • Administrators, operators, engineers and developers
  • Widening effort to include the Security and Network Performance diagnosticians

• Capture requirements by writing detailed scenarios
  • Day in the life before/after
  • Top 10 questions that they need to solve
  • Participants - Shibboleth, MACE-DIR, GRiDS, P2P (Lionshare), E2EP, IM
Scenario Process Findings

• No access to the diagnostic data
• Correlating different diagnostic data types
  • Application, network, system and security
• Discovering valuable information in a sea of data
• Providing evidence to prove or repudiate a diagnosis
• Finding time to create tools to transfer knowledge to help less skilled organizations and/or individuals diagnose problems
Outline

• Progress Report
  • Review of goals
  • Problem analysis from the perspective of the user
  • Survey
  • Architecture
    • Event data
    • Dissemination

• Present Efforts

• Timeline

• Conclusion
Survey of Diagnostic Tools

• Broken down into three areas
  • Highly focused based on solving a specific problem
  • Log file analysis and management
  • Broad scope, combining data from many different sources to form a comprehensive solution

• Lots of tools
• Effort far from complete
• Living document
Survey Findings

• Highly specialized
  • Deep – TCP performance, LDAP tuning, etc.
  • Broad – Albilene load, Web statistics, etc.
  • Beginning to see signs of comprehensive tools (MOM)

• Most focused vertically only using network, host, or application diagnostic data types

• Measurement based vs. event, active vs. passive

• Difficult to correlate events and anomalies between tools

• Limited to one type of input data format. I.e. they can’t take advantage of other rich sources of diagnostic data
Diagnostic Methods

• Passive
  • Log files
  • Network flow data

• Active
  • Injecting transactions
  • Correlating with passive techniques

• Performance Measurement
  • Specific instrumentation tests
  • Summarization from multiple elements
  • Can map into the event domain
Outline

• Progress Report
  • Review of goals
  • Problem analysis from the perspective of the user
  • Survey
  • Architecture
    • Event data
    • Dissemination

• Present Efforts

• Timeline

• Conclusion
What if…

• The collection and management of a wide variety of event data was made simple
• All application, host and network events could be correlated with each other
• There existed an API where developers could rapidly build new tools that use this rich set of data and its management infrastructure
Architecture Goals

• Expose a wide array of diagnostic data to diagnostic tool developers
• Make it simple to add new data types
• Construct a scalable, secure and reliable collection and dissemination facility
• Provide methods to manipulate, search, and retrieve the data
• Create an simple API to use the methods
• Deploy the methods on popular development platforms
Steps to Enable Diagnostic Applications

- Establish the common event record
- Enable the collection of events from a wide array of event sources
  - Network: NetFlow, SNMP, RMON, etc
  - Security: IDS, Snort, firewalls, etc
  - Applications: Shib, Dir, IM, P2P, smtpd, named, httpd, Kerberos, etc
  - Hosts: /var/log/*, Syslog, etc
Steps to Enable Diagnostic Applications (2)

- Build tools to create dissemination infrastructures that,
  - Allows access to the diagnostic data
  - Provides operators to filter, anonymize, aggregate, tag, store and archive the data
  - Enables pipelining of data operators to organize and manipulate diagnostic data based on an organization or federations policies
  - Provide a common API so applications can access the diagnostic data
What types of input events?

- **Application and Host**
  - Syslog
  - `/var/log/*`
  - http-access/error
  - MS specific (security, application and event logs)
  - Application specific (AFS, SHIB, LDAP, SMTP, etc.)
  - “many others”

- **Network**
  - NetFlow
  - SNMP
  - RMON
  - Others?

- **Security**
  - IDS
  - Cisco CNS
  - Snort
  - Others?
Enabling Diagnostic Applications With a Common Event Descriptor

Diagnostic applications (Middleware, Network, Security) can extract event data from multiple data sets.

- Dissemination Network
- Collection and Normalization of Events
- Middleware Related Events
- Network Related Events
- Security Related Events
Diagnostic applications (Middleware, Network, Security can extract event data from multiple data sets)

- **Meas:**"OWAMP-I2" OWAMP:start test
- **Normal:**"core" netflow:flow rec
- **Error:**"primary dns" DNS:can’t recurse
- **Critical:**"Aston App" SHIB:shire not found
- **Meas:**"core-q" SNORT:unknown ser
- **Critical:**"Firewall-A" IDS:excessive flows
Event Record

Event Descriptor Meta Field

- Version Number
- **Observation Description Pointer**
- ID – unique event identifier
- Time - start/stop
- IP Address(es) – source/(destination)
- **Source Class** – application, network, system, compound, bulk, management
- **Event Name Tag** – Native language ID, user defined
- Status – normal, informational, warning, measurement, critical, error, etc.
- **Major Source Name** – filename, Netflow, Syslogd, SNMP, shell program, etc.
- **Minor Source Name** – logging process name (named), SNMP variable name, etc.
- **Raw Data Encoding Mechanism** – Binary, ASN1, ASCII, XML, etc.
- **Raw Event Data Description Pointer**
Event Record

Event Descriptor Meta Field

- **Observation Description Pointer**
  - Address type of observer (IPV4, IPV6, MAC, etc.)
  - Address of observer
  - Address type of collection agent (IPV4, IPV6, MAC, etc.)
  - Address of collection agent
  - Source Type (file, stream, polled, interrupt)
  - Collection agent name (Netflow.1.0, named.2.3, etc.)
Event Record

Event Descriptor Meta Field

- Event Descriptor
- Raw Event Data

• Raw Event Data Description Pointer
  • Schema of raw event data
  • Parser code
Event Record

Event Descriptor Meta Field

- **Event Name Tag** – (null), user defined (can be multiple tags)
  - Examples:
    - "astronomy-app"
    - "ShibUserHandle=foo"
    - "WebFrontEnd"
Event Record Overhead

Event Descriptor Meta-Field

- Version Number – 1 byte
- **Observation Description Pointer** – 4 bytes
- ID – 10 bytes
- Time – 24 or 12 bytes
- IP Address(es) – (8 or 16 bytes) * 2 for IPV6
- Source Class – 1 byte
- **Event Name Tag** – 0 to 16 bytes typical (can be as large as 256)
- Status – 1 byte
- **Major Source Name** – 0 to 32 bytes typical (can be as large as 256)
- **Minor Source Name** – 0 to 16 bytes typical (can be as large as 256)
- Raw Data Encoding Language - 1 byte
- **Raw Event Data Description Pointer** – 4 Bytes
Simple Implementation

Consolidation of Web Events

Collection Module 🥧 Processing Module 🥧
Simple Implementation

Enterprise Web Enabled Application

- AuthN Server
- Web Server
- Target App
- Diagnostic Host

- HTTP-access
- HTTP-error
- /var/log/messages
- /var/log/cron
- Syslog
- App logs
- Shib logs

Collection Module ⃝ Processing Module ⚡
Advanced Implementation

Enterprise Event Collection

- HTTP-access
- HTTP-error
- /var/log/messages
- /var/log/cron
- Syslog
- App logs
- Shib logs
- NetFlow
Complex Implementation

Enterprise

Collection Modules | Processing Modules

Federation

Alerting apps, filtering data to federation and API to NMS

Massive collection, normalization, filtering or tagging

Archive, querying

Reporting, performance, and forensic apps

Federation specific reporting, performance and forensic apps

14 April 2004
Internal Operators

Data Management/Access
- Internal Monitoring
- AuthN/AuthZ
- Control and Configuration

Data Repository
- Archive
- DB

Record Manipulation
- Filter
- Aggregator
- Anonymizer
- Tagging
Data flows can be constructed to provide the desired function and policy within an enterprise or federation.
Backplane Elements

Network, System, & Application Events

Collection Agent
CPI
Manipulation

Access
Control
Secure and Reliable Transport

Diagnostic Apps
API
Query
Storage
Manipulation

Collection Module
Processing Module
Outline

• Progress Report
  • Review of goals
  • Problem analysis from the perspective of the user
  • Survey
  • Architecture
    • Event data
    • Dissemination

• Present Efforts

• Timeline

• Conclusion
Present Efforts

- Joint effort between security, E2EP and E2ED
  - Do we share common problems building diagnostic tools and their supporting infrastructures?
  - Can we benefit from each other's diagnostic efforts and the data collected?
  - Will a common event record to enable correlation?
  - Can we share an event collection and decimation infrastructure that enables the evolution of diagnostic applications?
Present Efforts (2)

• Participants
  • Eric Boyd (I2)
  • Chas DiFatta (CMU)
  • Ken Kingenstein (I2)
  • Russ Hobby (I2)
  • Cheryl Munn-Fremon (I2)
  • Mark Poepping (CMU)
  • Marty Schulman (I2)
  • Matt Zekauskas (I2)
Pilot Objectives

- **Study the normalization** strategies of the diagnostic data
- **Build an ultra modular architecture** where the impact of its evolution is minimized
- **Collect and distribute** the data in a highly flexible manner via piping of diagnostic data streams
- **Leverage resources** of other initiatives where possible to achieve a common goal
- **Provide simple operators** to manipulate data
- **Enable basic forensic applications**
Pilot Feature Roadmap

• Collection Module
  • CPI Version 1
  • Management layer – summary reporting only
  • Event record format version 1
  • 5+ agents (1 sample)
  • Local configuration

• Processing Module
  • API Version 1
    • Simple querying and filtering of stored records
    • Can connect to real-time stream
    • Authentication TBD
  • No ‘traditional db’, records stored in files
  • Event record operators – filter(regex), T, Y and tag
  • 3 Tools - forensic, reporting, and sample
  • Local configuration with pipelining
Outline

• Progress Report
  • Review of goals
  • Problem analysis from the perspective of the user
  • Survey
  • Architecture
    • Event data
    • Dissemination

• Present Efforts

• Timeline

• Conclusion
# Year One Activity Timeline

<table>
<thead>
<tr>
<th>Activities</th>
<th>Status</th>
<th>Sep – Nov 03</th>
<th>Dec – Feb 04</th>
<th>Mar – May 04</th>
<th>Jun – Aug 04</th>
<th>Sep 04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Startup</td>
<td>Done</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Startup</td>
<td>Done</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End-user Requirement Definition and Survey</td>
<td>Done</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architecture and Design</td>
<td>Active</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development, Testing and Distribution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Postmortem</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Major Milestones**
- Advisory Group Formed
- BOF at I2 Oct 03
- User Requirements Finalized
- Architecture/Design Finalized
- Internal Release
- Beta Release
- Pilot Release

14 April 2004
Conclusion

- Seeing a common need for a diagnostic framework spanning E2EP, security and middleware
- Looking for participants in the Pilot
  - Developers (collection agents, applications)
  - Testers
- For more information,
  - [http://middleware.internet2.edu/e2ed](http://middleware.internet2.edu/e2ed)
  - Or contact chas@cmu.edu
Outline

• Progress Report
  • Review of goals
  • Problem analysis from the perspective of the user
  • Survey
  • Architecture
    • Event data
    • Dissemination

• Present Efforts

• Timeline

• Conclusion