EDDY
End-to-end Diagnostics DiscoverY

A Framework for
Comprehensive Diagnostics

Chas DiFatta (chas@cmu.edu)
Mark Poepping (poepping@cmu.edu)
Diagnostics...?

You discover your car has a flat tire...
• You fix it you move on

It’s flat again a week later...
• Valve problem?
• Nail in the driveway?
• Neighbor kid busting my chops?

Can you check all failure possibilities?
• Might help if you knew when air started leaking
Why Diagnostics?

• Things break and it matters
• Systems built to ‘get it working’, not to be ‘fixed’
  – How to meter/maintain/fix after installation?
  – Only the end problem diagnostician knows
• Software reuse and layered infrastructures create dynamic dependencies
  – Diagnostic data may not be available at all
  – Certainly doesn’t follow service path
  – Minimally ‘out of band’, often ‘out of question’
Problems discovered...

Banes of the Distributed System Diagnostician

• Limited **access** to slices of diagnostic data
• **Discovering** valuable information in a sea of data
• **Correlating** different diagnostic data types
• Providing evidence for **non-repudiation** of a diagnosis
• Finding **time** to create tools to transfer diagnostic knowledge to less skilled organizations and/or individuals (automation)
Who are the Distributed System Diagnosticians?

In IT (lots of other diagnostic domains):
• Applications Support Personnel
• Systems Administrators
• Network Support Staff
• Security Response Folks
• Managers of Computing Infrastructure
• Help Desk
• Ordinary Users
Thinking about the Problem
[An Architecture for Diagnostic Infrastructure]

Sensing Technology
• State, transaction info, whatever…the ability to collect anything

Orchestration
• Data acquisition/normalization/transport, getting the
  – Instrumentation data you want
  – In the format that you need it in
  – Where you want it

Diagnostic Information – first stage of finding the needle in a stack of needles
• Generic translation and statistical methods
• Simple event correlation, visualization, longitudinal pattern analysis
• Data Lifecycle (must be policy driven)

Domain Analytics
• Detailed analyses, situational diagnosis, specialized UI’s
• Significant automation of the domain and implementation autonomies
What to do? EDDY

Enable correlation
• Common Event Record (CER) – a way to format event information to make it easier to process

Provide transport
• Diagnostic Backplane – a way to move CER’s around to make it possible to automate processing of events

Some simple event orchestration methods
• Normalize, transform, visualize, store, anonymize
A Few Details

Common Event Record
- Seen, normalized, type, GUID, severity
- Extensible payload, leverage domain data formats

Event Backplane
- Event Channel – data push
- Query Channel – data pull
- Control Channel – agent configuration
EDDY Agent Framework

Functionality (filter/route)

Security

Network

Application

System

Environmental

Normalizers

Storage Analysis

Application (in/out band)

Transformation

Visualization

API

Tools

API

NMS

API

AMS

API

Alert

CMU Confidential and Proprietary Information. All Rights Reserved
EDDY Extensibility and Scalability

You don’t need all the data, pick off only what you need...

- SNMP/RMON events
- Snort/IDS events
- Email logs
- Web logs
- System logs
- MS-MOM events
- Application events
- Network Flow events
- Environmental events

Normalization
Transformation
Transformation Second Order

Header
Header
Raw Payload
Header
Header
Specialized Payload
Header
Diagnostic Hypothesis

CMU Confidential and Proprietary Information. All Rights Reserved
Complications

Event Scale (ex. >10K network flows/sec)
Data Lifecycle (collection, filter, anonymize, aggregate, archive, scour)
Data Access Security
Site Configuration (day to day min to min deltas)
Federating Diagnostic Analyses
An Illustration

You discover that your gateway has a routing problem...
• Furrow an eyebrow, fix it and move on

You discover another routing problem a week later...
• Configuration or firmware problem?
• Downstream BGP problem?
• Grad student busting my chops?
• How many potential failure scenarios?
An Illustration (2)

What’s Involved?

- Peer network routers
- Routing process on your route server
- Traffic to/from route server
  - Through edge router (or not)
- Maybe:
  - Resources on route server
  - Information from IDS
Separate Event Domains

Service Log Info
Flow Info
System Info

Botted Hosts
Internal Scan/Attack
Combined Event Domains

Network Peer

Local Network

Federated Managers

Networking
Security
Systems

Host A
Host B
Host C

Botted Hosts
Internal Scan/Attack

ICS

IDS
Internal Probe

Service Log Info
Flow Info
System Info
Combined Event Domains

- LocalRteUpdate(Peer->Local; “the usual”)
- LocalRtrFlow(src=botctl, dst=hostA)
- LocalIDS(BotCTL:src=botctl, dst=B)
- InternalFlow(ICMP:src=hostA, dst=RteSrv)
- RteSrv(SysWarn:LowMemory)
- RouteUpdate(Local->Rtr;”missing a few”)
Federated Event Domains

0  LocalRteUpdate(Peer->Local; “the usual”)
1  RemoteRteUpdate(Peer->Local; “the usual”)
2  LocalRtrFlow(src=RRteSrv,dst=LRteSrv)
3  RemRtrFlow(src=RRteSrv,dst=LRteSrv)
4  LocalRtrFlow(src=botctl,dst=hostA)
431 LocalIDS(BotCTL:src=botctl, dst=B)
432 InternalFlow(ICMP:src=hostA, dst=RteSrv)
1234 RteSrv(SysWarn:LowMemory)
1235 LocalRteUpdate(Local->Rtr;”missing a few”)
1240 InternalFlow(src=LRteSrv, dst=LocalRTR)
What EDDY is

- Architecture for cross domain diagnostics
- An enabling technology that provides
  - Event ledger
  - Dissemination and correlation infrastructure,
    - Afford research access to event data (anonymized)
  - A development platform for diagnostic research
    - Domain specific
    - Domain agnostic
What EDDY is not

- A system/network/application/security management platform
- The analysis engine, it enables the analysis to happen with domain expertise
Ongoing Efforts

- **Architecture**: A solution for integrating the diagnosis of distributed network and systems
- **Standards**: Defining the next generation of event auditing (working with IBM and others)
- **Open Source Prototype**: An efficient event dissemination platform that can be installed on the end system or within network devices
- **Center for Diagnostic Research**: CIDAT
  - Concentrate, coordinate engineering on real data in support of other efforts
  - Large scale event observatory to accommodate a wide variety of events for research
Campus interactions...

ISO - Traffic analysis [demo]
• Security diagnostic applications
ISAM – Email message transport
• Email diagnostic applications
Computer Science – Dragnet
• Forensic analysis and auditing methods in real-time.
School of Architecture – Intelligent Workplace
  – Sensing the environment
Campus interactions...

CyLab – Reiter/Wing
  – Network Telemetry

Civil Engineering – CenSCIR
[Center for Sensed Critical infrastructure and Research]
  • Large scale orchestration of environmental events from externally and internally located sensors

Other discussions – PDL, Cert, ECE
[Parallel Data Lab, Computer Emergency Response Team, Electrical and Computer Engineering]
  • Data center large scale computing applications
  • Security applications
  • Distributed systems diagnosis
**Status**

- **Development**
  - Initial release (Munster 0.5) targeted at developers - 4/1/06
    - EDDY Agent Framework
    - TLS Scripts to support transport security
    - Sample EDDY Agents
      - 18 Normalization, Transformation & Display agents.
    - Agent Manager - start/stop EDDY agents on a host.

- **Outreach**
  - Involving others in the development process
  - Expand to other use cases external to CMU
  - Working with industry leaders on proposed standards and methods

- **Support**
  - Sponsored by the National Science Foundation under the NSF Middleware Initiative - Grant No. OCI-0330626
  - Soliciting partners in both industry and government
Want to Learn More?

• Web site
  – www.cmu.edu/eddy

• Mailing list
  – Eddy-info@lists.andrew.cmu.edu
EDDY
End-to-end Diagnostics Discovery
A Framework for Comprehensive Diagnostics

Chas DiFatta (chas@cmu.edu)
Mark Poepping (poepping@cmu.edu)