



Middleware End-to-End Diagnostic Advisory Group

Pilot Discoveries and Discussion

Fall 2004 Member Meeting

Agenda

- Review of year one deliverables
- Report on Pilot Effort
 - Architecture
 - Architecture
 - Common Event Record (Version 0.5)
 - Pilot Event Flow Topology
 - Usage Examples
 - Findings and Conclusions
 - Next Steps
- Discussion

Review of Year One Goals

- ✓ Engage NMI-EDIT and GRIDS efforts
 - ✓ Focus in intra not inter organizational issues
- ✓ 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
- ✓ Engage other groups and efforts
 - ✓ GRIDS
 - ✓ NMI-EDIT (Shibboleth, LionShare, DIR)
 - ✓ E2E Performance
 - ✓ SurfNet (Detective, UT)
- ✓ Initially focus on simple solutions to quickly aid in the analysis and evolution of middleware diagnostic support through a pilot effort
- ✓ Create a modular architecture to prepare for a rich toolset in year two



ccBay Pilot Architecture

Pilot Design

- Highly modular architecture utilizing standard building blocks.
- Focus on a simple and extensible lightweight design.
- Utilize existing libraries, utilities, modules and standards instead of existing systems.
 - We have looked at many, but there is no existing software that does all we need.
- Use Python as a full-featured development language.
 - Great language for development of pilot and beyond.
 - Widely included with Linux distributions.
 - Works well in the Windows environment.

Pilot Architecture

- Normalization agents to convert raw event and logging data to ccBay XML documents.
 - Each event is a file, more efficient approach needed beyond pilot.
 - For pilot, normalization agents
 - Linux (RH9 and Fedora C2) - kernel events (klogd), user events (slogd)
 - Windows (WIN2K and WIN2003 server) – application, system and security logs, installer (WIE)
 - Snort Events
 - NetFlow Events (Version 5)

Pilot Architecture

- Forwarding agents move XML documents between storage agents.
 - Transfer of XML files to storage agent using SCP for pilot.
 - Simple design for pilot, may not be production approach.

Pilot Architecture *(continued)*

- Storage agents to archive to files or populate database with XML data.
 - Storage agent to populate MySQL database for pilot.
- Transformation agents to anonymize, aggregate and remove events.
 - For pilot, basic anonymization by removing IP addresses.
- Applications to analyze event and logging data.
 - For pilot, basic forensic Web site and CLI tools.

Pilot Architecture *(continued)*

- Management System to manage and control the distributed system.
 - Diagnosis and ccBay event generation for the system itself.
 - For the pilot, management events will be generated for certain error conditions as well as startup/shutdown notifications.
 - Web, GUI and CLI applications for controlling and configuring the overall ccBay network of hosts.
 - Manually configure the ccBay hosts for the pilot.



Common Event Record Version 0.5 for Pilot

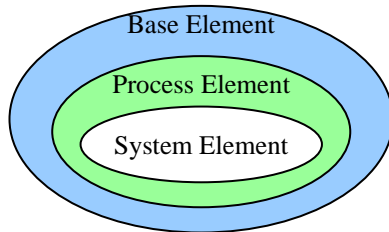
Pilot Common Event Record Design

- Must represent four types of events
 - Application – events that occur within an application
 - Network – occur within a network (flow based)
 - Security – changes or differences in security policy (from IDS or firewalls)
 - System – OS and hardware related events
- Provide a common structure to enable correlation of events (base element)
- Leverage existing data manipulation tools by representing data within an XML framework

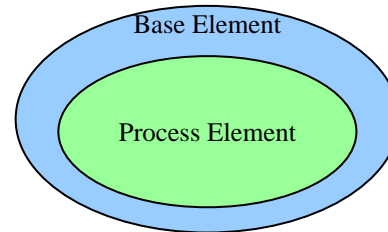
XML Schema Nesting

- ccBay events are constructed as nested XML sub-schemas:
 - System Event
 - A System element inside a Process element inside an Base element.
 - Application Event
 - A Process element inside an Base element.
 - Security Event
 - A Security element inside a Network element inside an Base element.
 - Network Event
 - A Network element inside an Base element.

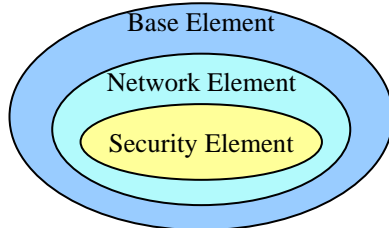
XML Schema Nesting (continued)



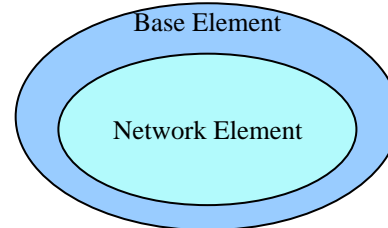
System Event



Application Event



Security Event



Network Event

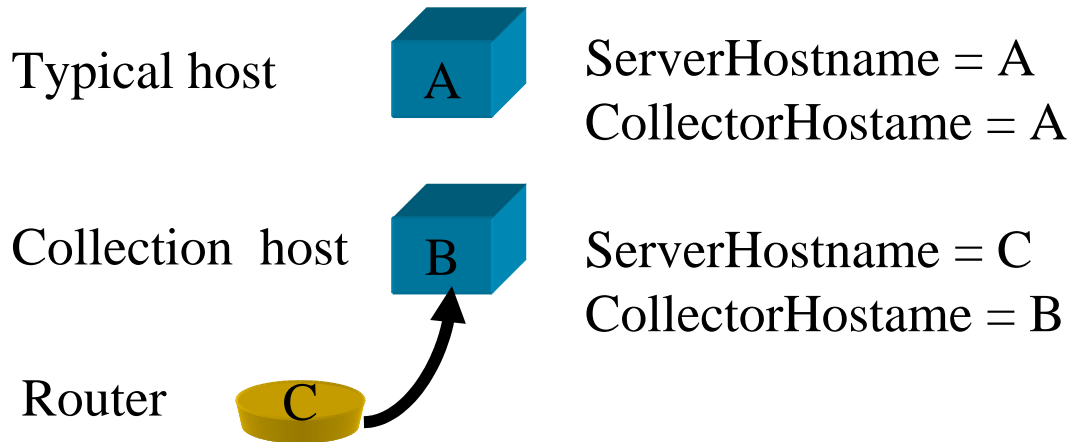
Illustration of nested XML elements.

BaseElement XML Schema

```
<org.Internet2.Middleware.ccBay.BaseElement Version = "0.1">
  <Record>
    <Tag STRING/> <!--Optional -->
    <TimeStart TIMESTAMP/> <!-- Required -->
    <TimeEnd TIMESTAMP/> <!-- Optional -->
    <ServerHostname FQDN/> <!-- Optional -->
    <ServerIP IP_ADDRESS/> <!-- Required -->
    <CollectorHostname FQDN/> <!-- Optional -->
    <CollectorIP IP_ADDRESS/> <!-- Required -->
    <CollectorName STRING/> <!-- Required -->
    <CollectorVersion FLOAT/> <!-- Required -->
    <WarnLevel STRING/> <!-- Required -->
    <EventMessage STRING/> <!-- Required -->
    <org.Internet2.Middleware.ccBay.ProcessElement/> <!-- Optional -->
    <org.Internet2.Middleware.ccBay.NetworkElement/> <!-- Optional -->
  </Record>
</org.Internet2.Middleware.ccBay.BaseElement>
```

BaseElement – Observation Points

- ccBay events can be observed directly or indirectly
 - ServerHostname – the system that ccBay resides on
 - CollectorHostname – the system that produces the events



BaseElement – Warning Levels

- EMERG - system is unusable
- ALERT - action must be taken immediately
- CRIT - critical conditions
- ERR - error conditions
- WARNING - warning conditions pre-error
- NOTICE - normal but significant condition
- INFO - informational
- DEBUG - debug-level events

BaseElement – User Tags

- Meaning can be given to an event by writing a normalizer that inserts a user defined tag based on an event criteria
 - Specific data that enhances the event
 - UserTag = “SwapUsed:1.67GB”
 - UserTag = “cd@cmu.edu changed config”
 - UserTag = “<throughput>993</throughput>”
 - Anonymized user hashes for tracking and debugging
 - UserTag = “user:d3b07384d113edec49eaa6238ad5ff00”

```
<org.Internet2.Middleware.ccBay.ProcessElement Version = "0.1">
  <Record>
    <ProcessID INT/> <!-- Required -->
    <ProcessName STRING/> <!-- Required -->
    <ProcessOwner STRING/> <!-- Required -->
    <org.Internet2.Middleware.ccBay.SystemEvent/> <!-- Optional -->
  </Record>
</org.Internet2.Middleware.ccBay.ProcessElement>
```

SystemEvent XML Schema

```
<org.Internet2.Middleware.ccBay.SystemEvent Version = "0.1">  
  <Record>  
    <Facility STRING/> <!-- Optional -->  
    <Subsystem STRING/> <!-- Optional -->  
  </Record>  
</org.Internet2.Middleware.ccBay.SystemEvent>
```

NetworkEvent XML Schema

```
<org.Internet2.Middleware.ccBay.NetworkElement Version = "0.1">
  <Record>
    <SourceIP IP_ADDRESS/> <!-- Required -->
    <DestinationIP IP_ADDRESS/> <!-- Required -->
    <NextHopIP IP_ADDRESS/> <!-- Required -->
    <InterfaceIn STRING/> <!-- Required -->
    <InterfaceOut STRING/> <!-- Required -->
    <Packets INT/> <!-- Required -->
    <Octets INT/> <!-- Required -->
    <SourcePort INT/> <!-- Required -->
    <DestinationPort INT/> <!-- Required -->
    <IPTypeOfService STRING/> <!-- Required -->
    <Protocol STRING/> <!-- Required -->
    <TypeOfService STRING/> <!-- Required -->
    <SourceAutonomousSystem STRING/> <!-- Required -->
    <DestinationAutonomousSystem STRING/> <!-- Required -->
    <SourceMask OCTAL/> <!-- Required -->
    <DestinationMask OCTAL/> <!-- Required -->
    <org.Internet2.Middleware.ccBay.SecurityEvent/> <!-- Optional -->
  </Record>
</org.Internet2.Middleware.ccBay.NetworkElement>
```

SecurityElement XML Schema

```
<org.Internet2.Middleware.ccBay.SecurityElement Version = "0.1">  
  <Record>  
    <ReferenceSystem STRING/> <!-- Required -->  
    <TCPHeader INT/> <!-- Optional -->  
    <UDPHeader INT/> <!-- Optional -->  
    <ICMPHeader INT/> <!-- Optional -->  
    <RawData STRING/> <!-- Optional -->  
  </Record>  
</org.Internet2.Middleware.ccBay.SecurityElement>
```

System Event Normalization Example

- Raw entry from /var/log/messages:

Jul 29 15:07:27 cmu1 sshd[11157]: Failed password for illegal user Administrator from ::ffff:192.168.2.6 port 4324 ssh2

- Raw entry normalized as XML as a System Event:

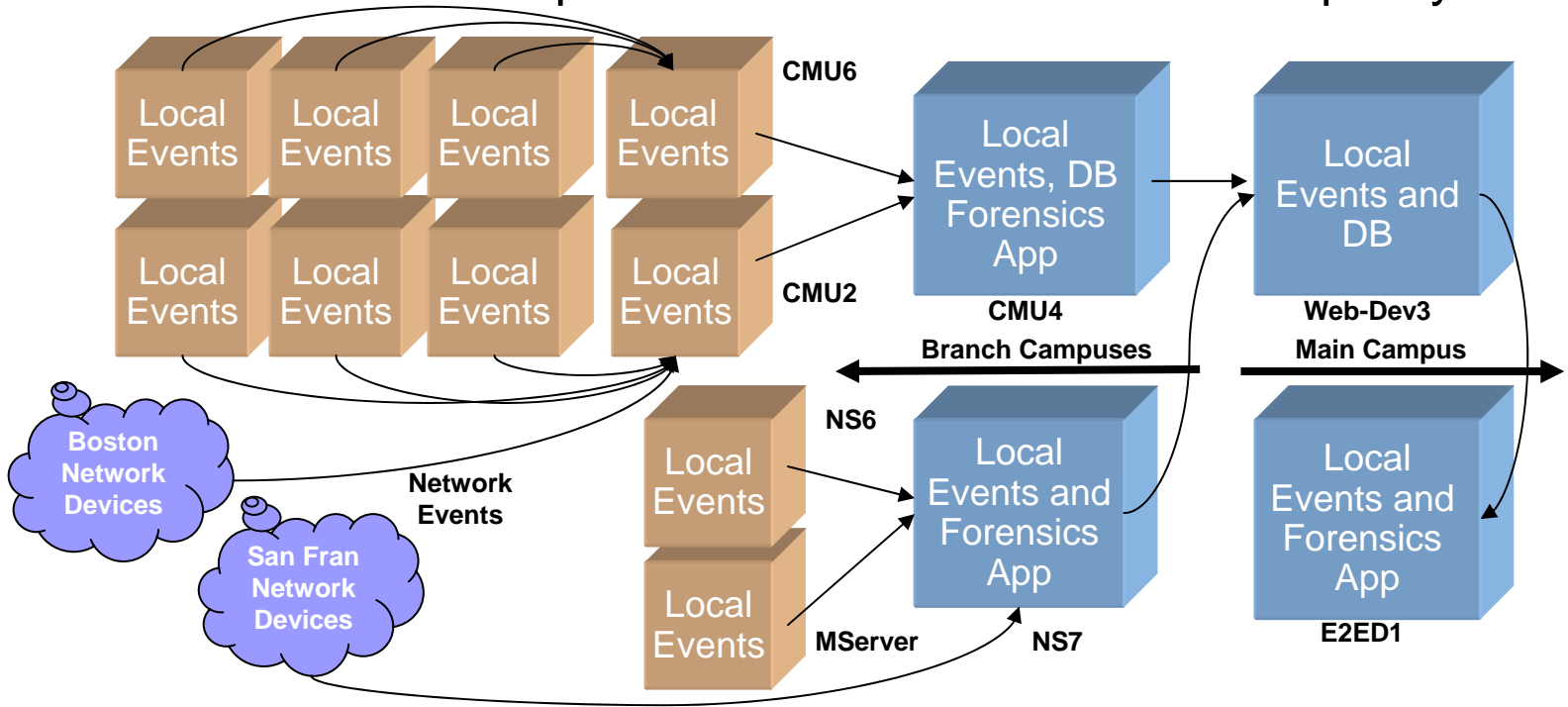
```
<org.Internet2.Middleware.ccBay.BaseElement Version="0.1">
  <Record>
    <TimeStart>Jul 29 15:07:27</TimeStart>
    <ServerHostname>cmu1</ServerHostname>
    <ServerIP>192.168.2.2</ServerIP>
    <CollectorHostname>cmu1</CollectorHostname>
    <CollectorIP>192.168.2.2</CollectorIP>
    <CollectorName>ccbay-slogd</CollectorName>
    <CollectorVersion>0.1</CollectorVersion>
    <WarnLevel>info</WarnLevel>
    <EventMessage>Failed password for illegal user Administrator from ::ffff:192.168.2.6 port 4324 ssh2</EventMessage>
  </org.Internet2.Middleware.ccBay.ProcessElement>
  <Record>
    <ProcessName>sshd</ProcessName>
    <ProcessID>11157</ProcessID>
    <org.Internet2.Middleware.ccBay.SystemElement>
      <Record>
        <Facility>User</Facility>
      </Record>
    </org.Internet2.Middleware.ccBay.SystemElement>
  </Record>
</org.Internet2.Middleware.ccBay.ProcessElement>
</Record>
</org.Internet2.Middleware.ccBay.BaseElement>
```



Pilot Event Flow Topology

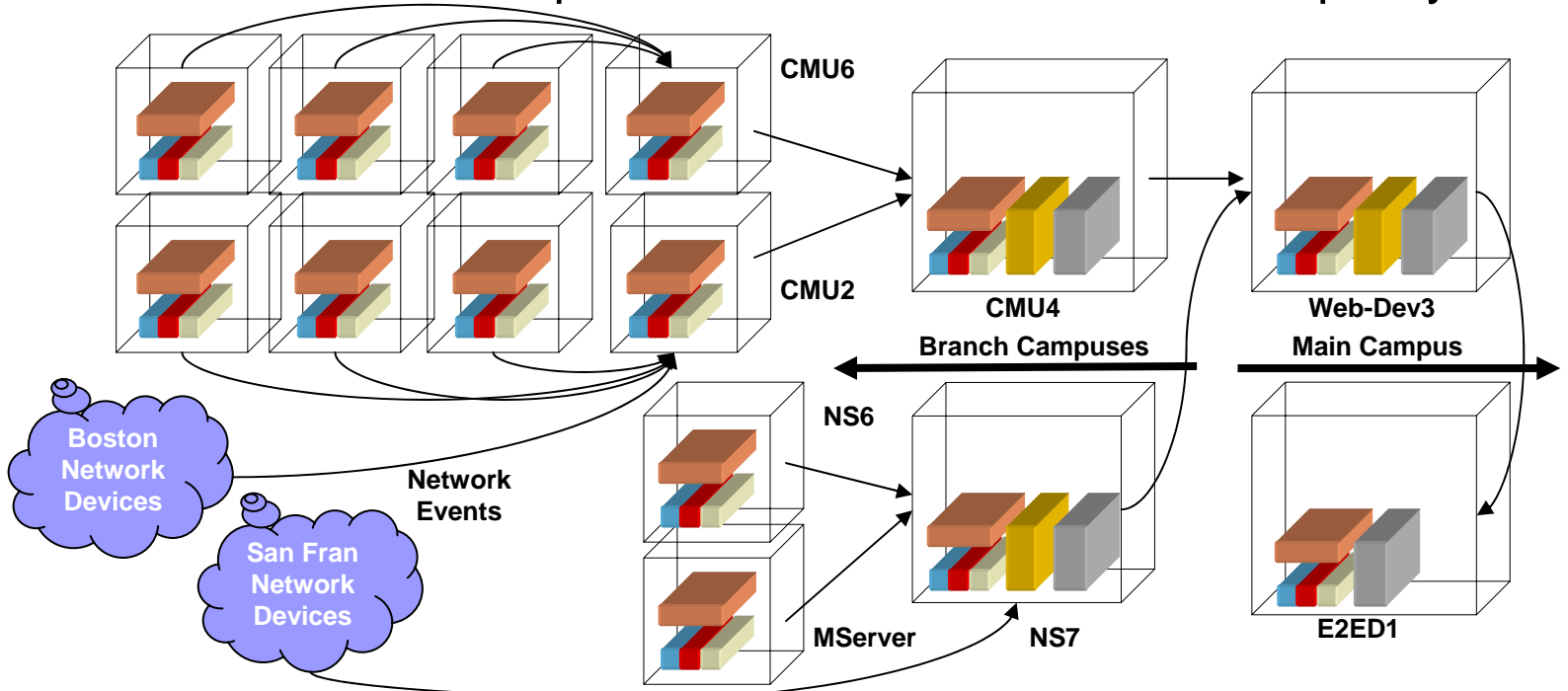
Event Flows

Customized flows to provide the desired function and policy



Event Flows

Customized flows to provide the desired function and policy



Normalization Agent

Transformation Agent

DB Agent

Archive Agent

Management Agent

Forwarding Agent

Application Agent



Use Examples From the ccBay Pilot

Diagnostic Questions Answered

User: network/system administrator who owns DNS

Question: The help desk says that some users can't resolve hosts in the CS domain but they can. What's wrong?

Administrator: *What errors are on the recursive DNS servers?*

```
$ccbay-query --query="ServerIP='192.168.68.17' and "ServerIP='192.168.68.13' and \  
WarnLevel='err'" -s -Ea -l30 | less
```

Displaying 30 of 582 events

```
2004-09-23 21:07:02.30:error:ns7.bigu.edu:192.168.68.17:lin-ulogd:  
ProcessElement:named:3194:zone cs.bigu.edu/IN: refresh: failure trying master  
192.168.24.202#53: timed out
```

Administrator: *What errors are on the authoritative DNS server?*

```
$ccbay-query --query="ServerIP='192.168.8.202' and WarnLevel='err'" -s -Ea -l10 | less
```

Displaying 10 of 82 events

```
2004-09-23 21:33:17.20:error:ans.bigu.edu:192.168.8.202:lin-ulogd:  
ProcessElement:named:3194:updating zone 'cs.bigu.edu/IN': update failed:  
'RRset exists (value dependent)' prerequisite not satisfied (NXRRSET)
```

Administrator: *Problem found, authoritative name server accidentally removed from zone file.*

User: developer (ccBay)

Question: I can verify that the web application on e2ed1.andrew.cmu.edu was having problems around noon when I installed a new version of the application. What is the problem?

Administrator: *What errors are occurring in the web logs on e2ed1.andrew.cmu.edu?*

```
$ccbay-query --query="ServerIP='128.2.6.48' and WarnLevel='err' and ProcessName='httpd' \  
and TimeStart>'2004-09-23:11:5500.00'" -s -Ea
```

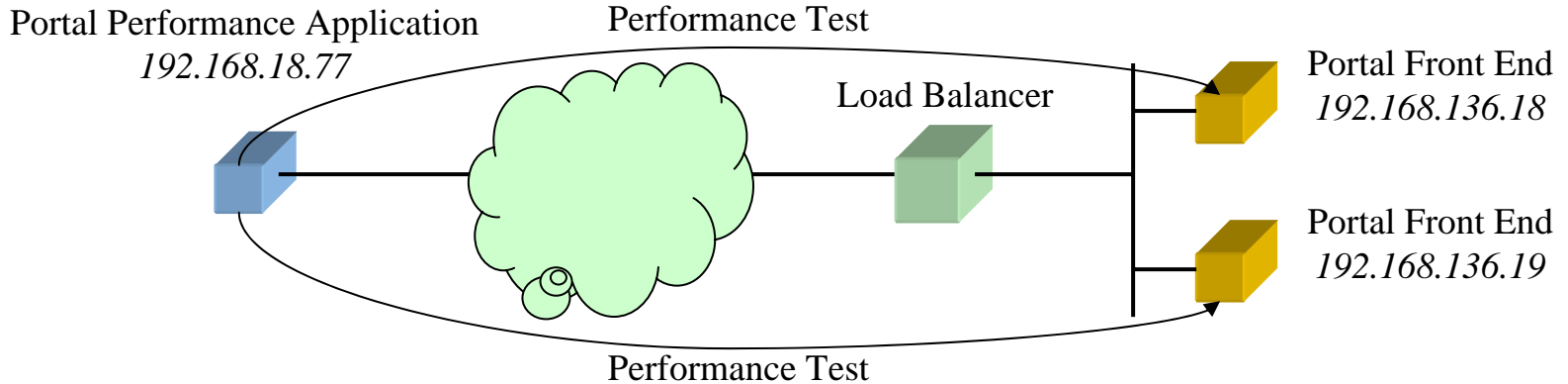
```
2004-09-23 19:01:00.00:err:e2ed1.andrew.cmu.edu:128.2.6.48:apached:ProcessElement:  
httpd:None:Client IP: 67.127.171.89 Message: File does not exist:  
/usr/www/htdocs/scripts/showresults.py?id=4xcF34Df
```

Administrator: *Problem found, wrong file name and path in application.*

Application/Performance Related (future)

User: GRID/system administrator owner of a portal

Question: Users are complaining that the portal much slower today, is it?



Application Related (future)

User: GRID/system administrator owner of a portal

Question: Users are complaining that the portal much slower today, is it?

Administrator: *What operation is taking longer then normal on front end server 192.168.136.18?*

```
$ccbay-query --query="ServerIP='192.168.18.77' and WarnLevel='info' and  
Target='192.168..136.18'" -s -Ep -l30 | less
```

Displaying 30 of 582 events

```
2004-09-23 21:07:02.30:info:portal0.bigu.edu:192.168.136.18:lin-perflogd:
```

```
ProcessElement:portperfcheck:3014:GET:/:resp=2.0sec;POST:/cgi-bin/perfcheck?  
user=test&pass=foo&test=quickperf:total=10.2sec,registration=3.4sec,dbtest=5.6
```

Administrator: *What operation is taking longer then normal on front end server 192.168.136.19?*

```
$ccbay-query --query="ServerIP='192.168.18.77' and WarnLevel='info' and  
Target='192.168..136.19'" -s -Ep -l30 | less
```

Displaying 30 of 582 events

```
2004-09-23 21:07:02.30:info:portal1.bigu.edu:192.168.136.19:lin-perflogd:
```

```
ProcessElement:portperfcheck:3014:GET:/:resp=4.2sec;POST:/cgi-bin/perfcheck?  
user=test&pass=foo&test=quickperf:total=26.8sec,registration=13.4sec,dbtest=11.6sec
```

Administrator: *Need to take 216.91.136.19 out of load balancer pool and examine further.*

User: network/system administrator

Question: The help desk says that some users can use the portal while others cannot? What is the problem?

Administrator: *Are the hosts that can't use the web service getting to the firewall?*

```
$ccbay-query --query="SourceIP='10.2.24.44' and "DestinationIP='192.168.24.84' and \  
DestPort='80' ServerIP='192.168.2.45'" -s -En | less  
2004-09-23 17:18:14.00:2004-09-2300:00:00.00:info:outsidefw.bigu.edu:192.168.168.13:  
netflowd:NetworkElement:10.2.24.44:192.168.24.84:N/A:6:564:3332:  
80:6 (TCP):Active=0.551 ms; BytesPerPacket=94; Flags=0x1b
```

^c

Administrator: *Since we're getting to the firewall, are we getting through to the other side?*

```
$ccbay-query --query="SourceIP='10.2.24.44' and "DestinationIP='192.168.24.84' and \  
DestPort='80' and ServerIP='192.168.2.24'" -s -En -l24
```

Displaying 0 of 0 events

\$

Administrator: *Problem found on firewall/, incorrect netmask.*

User: engineering manager of a web services group

Question: Are there malicious http requests being sent to any of my Web servers?

Answer: The manager requests a count of the events originating from Snort that are Web server related alerts.

```
$ ccbay-query -q "CollectorName='snortd' and ReferenceSystem like '%WEB%'" | wc  
38 837 14304
```

If 'wc' shows a line count other than 0, these Snort alerts need to be investigated with further queries to verify that the Web server accessed returned a status value other than 200 (success). To determine which exploits were attempted, the query above is run again, this time without the 'wc' redirection (next slide)...

Security Related

User: engineering manager of a web services group

Question: What are the malicious http requests being sent to my Web servers?

Answer: The manager requests all events originating from Snort that are Web server related alerts.

```
$ ccbay-query -q "CollectorName='snortd' and ReferenceSystem like '%WEB%'"
```

```
<snip>
```

```
2004-09-24
```

```
02:07:27.00:alert:cmu6:192.168.2.7:snortd:SecurityElement:24.60.136.78:66.33.216.70:N/A:0:0:32767:
```

```
80:TCP:[1:1149:12] WEB-CGI count.cgi access:0:ID: Snort Version 2.2.0RC1 (Build 28)
```

```
Classification: access to a potentially vulnerable web application Priority: 2
```

```
<snip>
```

The manager is shown a detailed list of the Snort Web server related alerts. One of them is displayed here. The manager now needs to verify that none of his Web servers successfully responded to the Web request for this page or any of the other malicious http requests reported by Snort (next slide)...

User: engineering manager of a web services group

Question: Were any of the malicious http requests reported by Snort successfully replied to?

Answer: The manager verifies that all malicious page requests are treated as HTTP errors.

```
$ ccbay-query -q "CollectorName='apached' and EventMessage like 'Status: 200' and EventMessage like 'count.cgi'"  
$
```

The query above asks for all Apache events for the suspected malicious page that resulted in a successful status of 200. In this case, the query application does not display any matching events. This indicates that the Web server did not respond to this request and generated a 404 status or other error value instead.



Advantages and Limitations of the Pilot Implementation

Common Event Record

Advantages of Version 0.5 of the CER

- Enables the correlation of events from different areas (network, application, security and system) in a simple and effective manner
- Gives flexibility for representing specialized meaning that is specific to the environment where it is deployed
 - Example: `<Tag>RunQueue=14</Tag>`
- Assigning metrics about the severity of an event
 - Based on syslog(3) conventions
- OS and platform independent
 - Linux, Fedora and Windows implementations

Common Event Record

Limitations of Version 0.5 of the CER

- Severity metrics not always standard across hosts
- Address scaling issues and consider other data representation formats
- Need to include second order events from the Measurement/Performance domains
- Event message must be parsed to provide a more granular representation of an event to aid in correlation
- Overhead of XML may outweigh the advantages with respect to event record size and the time it takes to parse

Advantages

- Greatly increases the efficiency of the diagnostician by
 - Reducing the time of gathering artifacts of the fault by centralizing vast numbers of events in a common area
 - Provides a way to search and correlate vast amounts of data
 - Centralizing or localizing vast numbers of events
- Event Flow Topology
 - Event collection, storage and transmission based on policy
 - You only collect what you need
 - You only send what you want
 - You can operate on the events you send based on policy
 - Highly flexible configuration to adapt to scale and policy issues

- Advantages continued...
 - Diagnostic applications
 - The needles in the haystack start to jump out very quickly
 - You can get lost with all the “why is this happening?”
 - CLI (ccbay-query)
 - Can feed other diagnostic, reporting alerting tools
 - The workhorse for digging in with SQL commands
 - Web (ccbay-console)
 - Simple and effective web interface
 - Good primer for starting with the ability to save queries
 - Python based
 - Good prototyping environment with lots of libraries
 - Gaining a large adoption base quickly

Limitations

- Even though the architecture is built to scale, the scaling issues are still an unknown
 - Didn't have the hardware to adequately test
 - Main DB was having a hard time keeping up (only a 700Mhz 4 year old Dell desktop) collecting all events
 - SCP as a transport method is inefficient
 - Interpretative language (Python) not efficient
 - XML based event record has too much overhead?

Limitations continued...

- No real Authz/Authn methodology
 - Shibboleth when non-web API ready
- Anonymization method too simple
- No real-time event flow observation capability
 - Need to be able to apply applications not just to the DB or archive, but to the normalizers or any point in the backplane topology
- Basic Management and Configuration
 - Configuration could be much simpler
 - Remote and automated configuration is highly desired
 - Tools needed to manage diagnostic data lifecycle
- Bi-directional network auditing facility needed
 - NetFlow not enough

Conclusion

- As is, ccBay provides a huge value to our environment by answering questions like
 - “How long has that been misconfigured?”
 - “What is that TCP connection for?”
 - “Do I have the firewall configured correctly?”
- The concept is proven but needs more work to take it to the next level with respect to
 - Usability
 - Scale
 - Integration with other diagnostic applications



Building from the Pilot Effort... Next Steps

Next Steps

Mature the Common Event Record

- Solicit input on completeness of version 1.0
- Must be able to morph to new CER formats and providing backward compatibility
- Address scaling issues with respect to the record size and consider other data representation formats
- Include second order events such as Measurement/Performance
- Incorporate a mechanism for more granular correlation of events

Next Steps

Scale the diagnostic backplane

- Adopt a real Authz/Authn methodology
 - We use certificates at this time, but management is an issue
 - Shibboleth non-web version ready
- Provide an anonymization API to provide capability to satisfy policy where necessary
- Transport method evolution (to SOAP?)
 - Remove the dependency of SCP
 - Add real-time flow capability
- Migration from Python or offload compute intensive areas to another language
- Management and Configuration
 - Centralized configuration
 - Keep the configuration work on the clients hands free

Add Applications

- Domain specific
 - Work with middleware application, network, system, security groups to build focused apps based on what we've learned from scenario writing process
 - Discuss performance/measurement with external groups
- Mature and establish a base application with GUI interface for forensics and reporting
 - Reporting – feed appellations like cricket and crystal reports
 - Forensics – need a client GUI interface that is ported to Linux, Mac and Windows

Add Applications

- Build simple but high value tools that extract information from the archive and not the DB
 - Summaries of events
 - For retrieving data that is not sent to the DB
- Version1 of the Event API
 - Acquiring a real-time event flow from any node
 - Simple data locator service (where can I find this data)
 - Querying data repositories directly but be conscious of future capabilities where agents may mine data over multiple repositories

Next Steps

Getting traction with other groups has been difficult but...

- Assemble a ccBay early adopters group
- Composed of key I2 campus stakeholders
 - CMU and Duke thus far
- Soliciting feedback from others on new and existing features
- Involve members in development efforts
 - Retrofitting existing diagnostic applications to use backplane
 - OWAMP, SurfNet Detective, UT
 - Build new normalization agents to provide more granularity of the event and consider a mechanism for registering them
 - To assist in developing core components

Group Discussion

Special thanks to the following to make the pilot effort possible...

- Jim Gargani – Lead Developer
- Ryan Muldoon – Developer
- Mark Poepping