Progress in Near-Real Time Attack Detection at the Platform Level

Dr. Bruce Gabrielson (BAH)
CND R&T PMO
22 September 2010

UNCLASSIFIED//FOR OFFICIAL USE ONLY

Detection Objective

The overall objective of the this task was to architect and implement a capability that will enable automated parsing, normalization, *extraction*, *aggregation*, *filtering* and then *detection* of attack patterns based on log and log like data in near real time depending on local network settings. We call this the Audit Data Extraction Utility (ADEU).

The Detection Concept

- Real World Problems
  - Audit logs are created in many different variations.
    - Attack identification using multi-platform analysis nearly impossible.
  - Collecting all audit and audit like data and then identifying attacks in near real time is difficult within the current architecture.
    - The massive amount of data overloads our network resources.
    - Dynamic anomaly detection using audit logs creates many false positives.
- Practical Solution
  - Not all log data is needed.
    - By minimizing the data elements based on detection needs, a deployed agent can collect only the audit data required to match defined attack use cases using static analysis.
    - White-listing regular non-malicious log entries further reduces excessive data collection
    - Data normalization to an evolving standard supports automated multi-platform analysis.

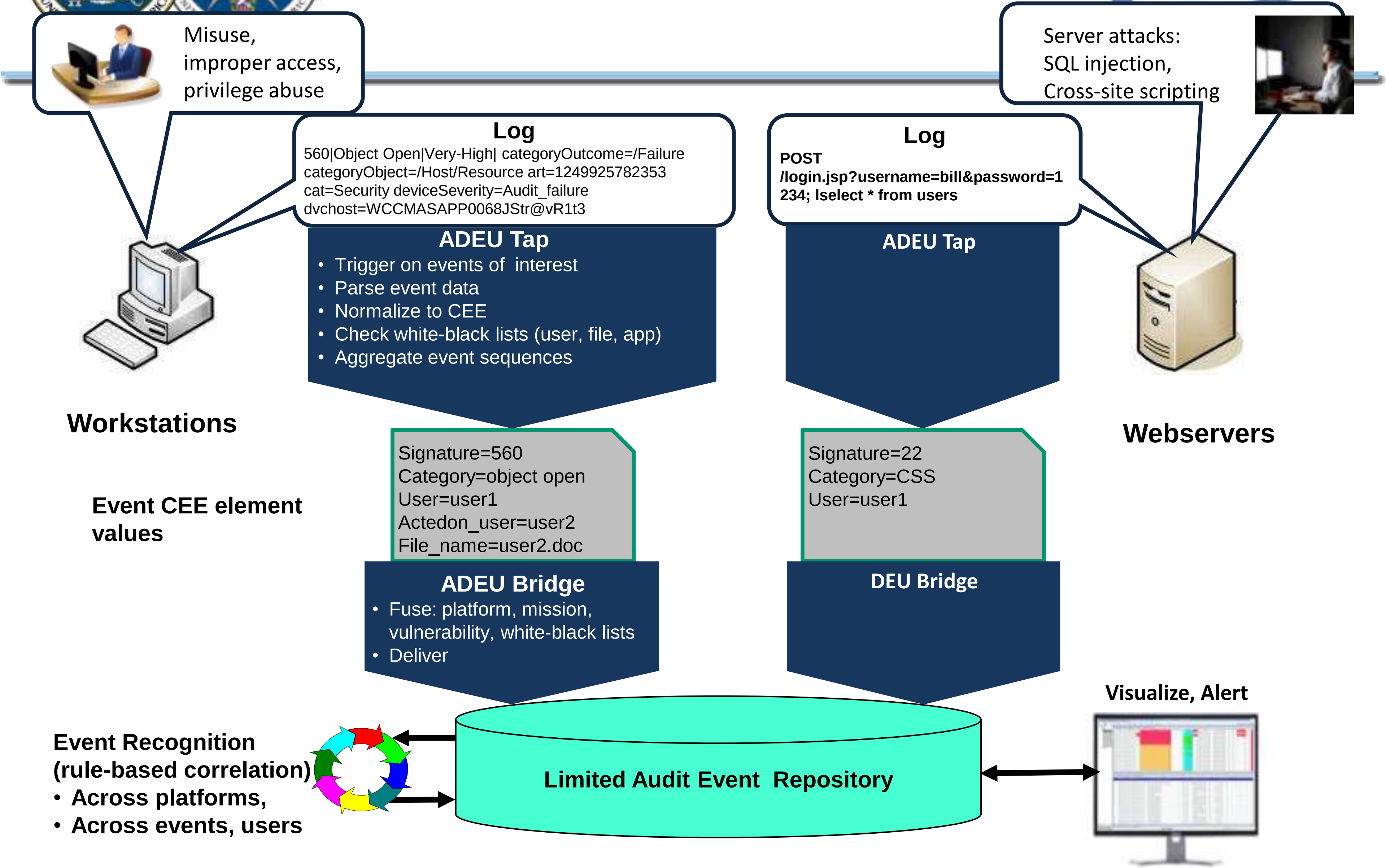
Design Approach to Reduce Collection Needs

- To reduce the actual log data necessary for detection, a more focused approach than currently available in industry was developed.
  - The combination of data calls and research initiatives produced a vetted list of insider threat use cases for windows workstations.
  - Additional research, vendor collaboration, and data calls within the financial community resulted in the development of insider threat use cases for Linux workstations, and Apache/IIS web servers.
  - New research underway for routers, printers, and firewalls.

Data Normalization

- The Common Event Expression (CEE) is a standardized log language for event interoperability in IT systems
  - Standardizes how computer/device events are described, logged, and exchanged.
    - The log syntax, transport, and taxonomy are under development.
    - Using CEE requires a format for expressing audit data.
  - The Event Management Automation Protocol (EMAP) is the standardized format to express, enumerate, measure, and interact with audit event data.
    - The EMAP framework will be interactive with and have similarities to the Security Content Automation Protocol (SCAP) in its construction.

ADEU Architecture



Misuse,
improper access,
privilege abuse

Server attacks:
SQL injection,
Cross-site scripting

Log

```
560|Object Open|Very-High| categoryOutcome=/Failure
categoryObject=/Host/Resource art=1249925782353
cat=Security deviceSeverity=Audit_failure
dvchost=WCCMASAPP0068JStr@vR1t3
```

Log

```
POST
/login.jsp?username=bill&password=1
234; lselect * from users
```

ADEU Tap

- Trigger on events of interest
- Parse event data
- Normalize to CEE
- Check white-black lists (user, file, app)
- Aggregate event sequences

ADEU Tap

Workstations

Webservers

Event CEE element values

Signature=560
Category=object open
User=user1
Actedon_user=user2
File_name=user2.doc

Signature=22
Category=CSS
User=user1

ADEU Bridge

- Fuse: platform, mission, vulnerability, white-black lists
- Deliver

DEU Bridge

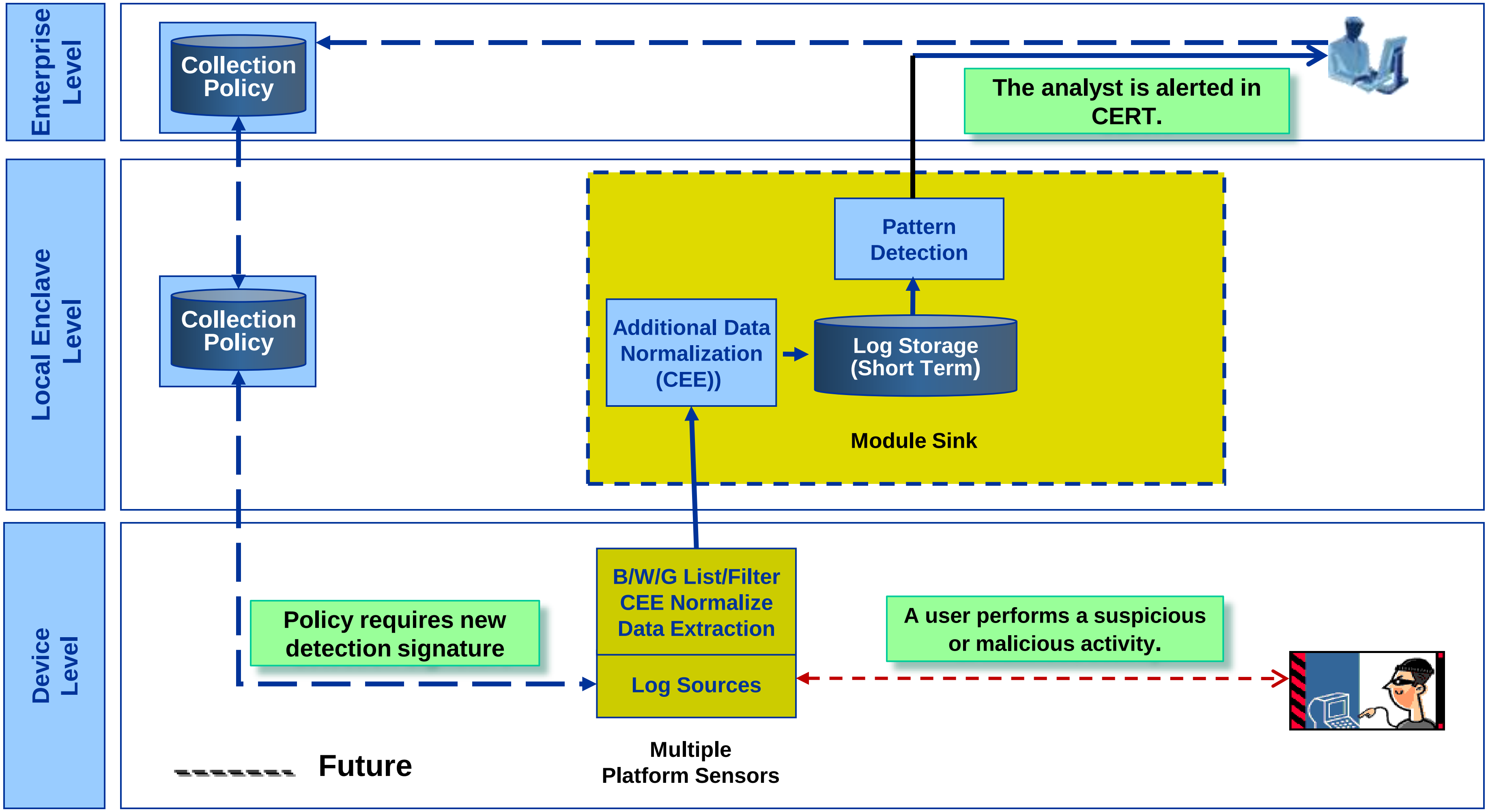
Event Recognition (rule-based correlation)

- **Across platforms,**
- **Across events, users**

Limited Audit Event Repository

Visualize, Alert

ADEU Data Flow



Enterprise Level

Collection Policy

The analyst is alerted in CERT.

Local Enclave Level

Collection Policy

Pattern Detection

Additional Data Normalization (CEE))

Log Storage (Short Term)

Module Sink

Device Level

Policy requires new detection signature

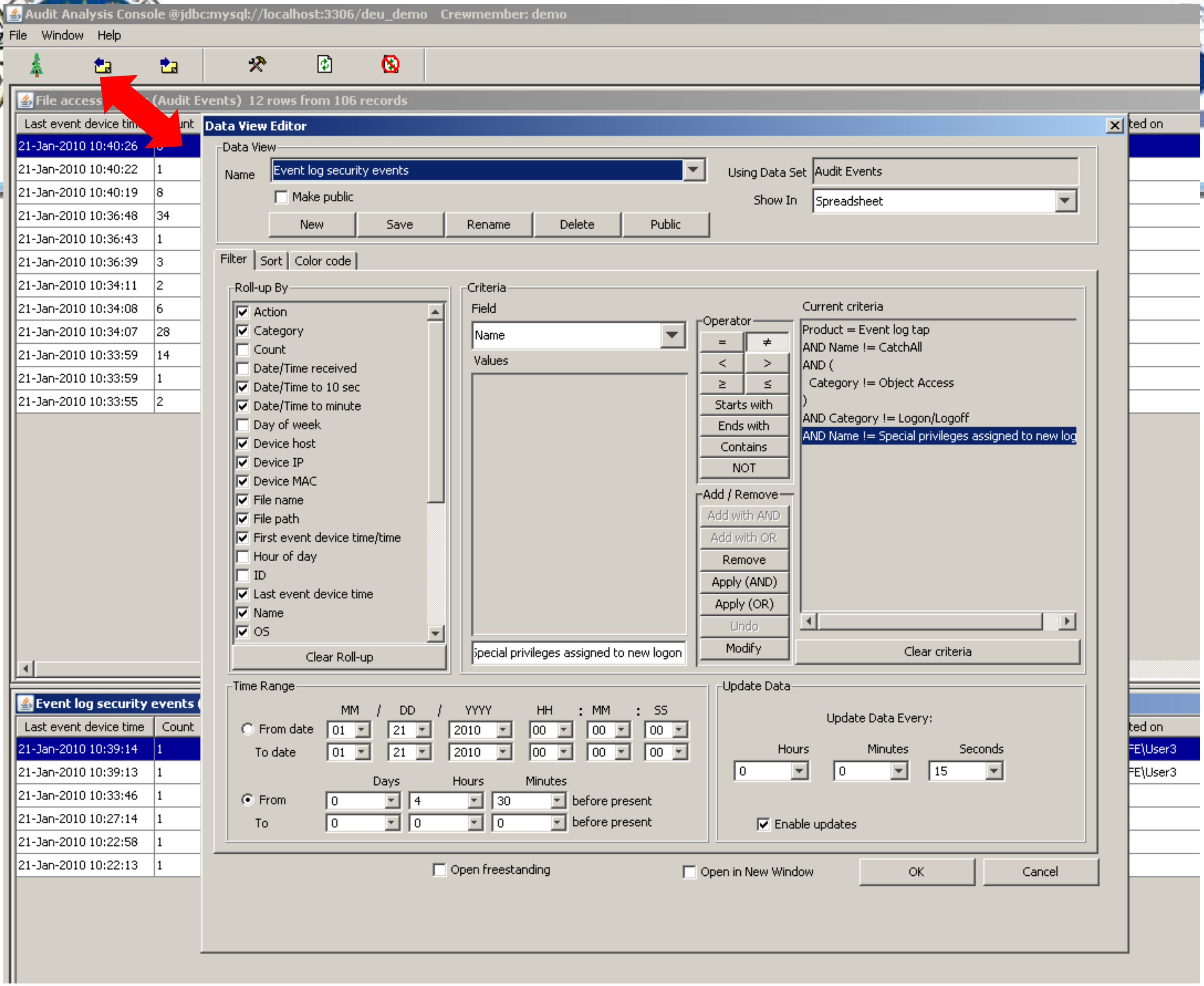
B/W/G List/Filter
CEE Normalize
Data Extraction

Log Sources

A user performs a suspicious or malicious activity.

Future

Multiple Platform Sensors



Audit Analysis Console @jdbc:mysql://localhost:3306/deu_demo Crewmember: demo

File access (Audit Events) 12 rows from 106 records

Last event device tim	unt	ted on
21-Jan-2010 10:40:26	6	
21-Jan-2010 10:40:22	1	
21-Jan-2010 10:40:19	8	
21-Jan-2010 10:36:48	34	
21-Jan-2010 10:36:43	1	
21-Jan-2010 10:36:39	3	
21-Jan-2010 10:34:11	2	
21-Jan-2010 10:34:08	6	
21-Jan-2010 10:34:07	28	
21-Jan-2010 10:33:59	14	
21-Jan-2010 10:33:59	1	
21-Jan-2010 10:33:55	2	

Event log security events (

Last event device time	Count	ted on
21-Jan-2010 10:39:14	1	FE\User3
21-Jan-2010 10:39:13	1	FE\User3
21-Jan-2010 10:33:46	1	
21-Jan-2010 10:27:14	1	
21-Jan-2010 10:22:58	1	
21-Jan-2010 10:22:13	1	

Data View Editor

Data View

Name: Event log security events

☐ Make public

Using Data Set: Audit Events

Show In: Spreadsheet

Roll-up By

- ☑ Action
- ☑ Category
- ☐ Count
- ☐ Date/Time received
- ☑ Date/Time to 10 sec
- ☑ Date/Time to minute
- ☐ Day of week
- ☑ Device host
- ☑ Device IP
- ☑ Device MAC
- ☑ File name
- ☑ File path
- ☑ First event device time/time
- ☐ Hour of day
- ☐ ID
- ☑ Last event device time
- ☑ Name
- ☑ OS

Criteria

Field: Name

Values: ipecial privileges assigned to new logon

Current criteria

```
Product = Event log tap
AND Name != CatchAll
AND (
  Category != Object Access
)
AND Category != Logon/Logoff
AND Name != Special privileges assigned to new log
```

Time Range

	MM	DD	YYYY	HH	MM	SS
○ From date	01	21	2010	00	00	00
To date	01	21	2010	00	00	00

	Days	Hours	Minutes	
● From	0	4	30	before present
To	0	0	0	before present

Update Data

Update Data Every:

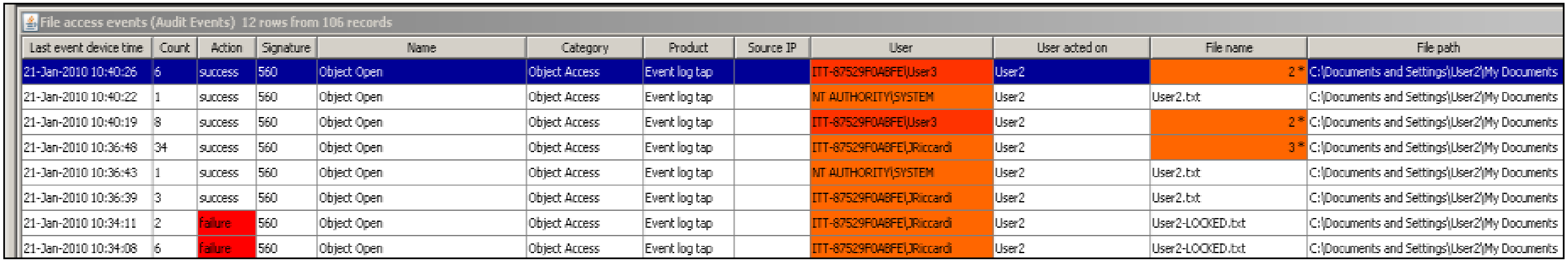
Hours	Minutes	Seconds
0	0	15

☑ Enable updates

☐ Open freestanding ☐ Open in New Window

The Data View Editor is the heart of DEU. It is invoked by the tools button and used to define the content and presentation of a window including:

Pattern Match Display



File access events (Audit Events) 12 rows from 106 records

Last event device time	Count	Action	Signature	Name	Category	Product	Source IP	User	User acted on	File name	File path
21-Jan-2010 10:40:26	6	success	560	Object Open	Object Access	Event log tap		ITT-87529F0A8FE\User3	User2	2 *	C:\Documents and Settings\User2\My Documents
21-Jan-2010 10:40:22	1	success	560	Object Open	Object Access	Event log tap		NT AUTHORITY\SYSTEM	User2	User2.txt	C:\Documents and Settings\User2\My Documents
21-Jan-2010 10:40:19	8	success	560	Object Open	Object Access	Event log tap		ITT-87529F0A8FE\User3	User2	2 *	C:\Documents and Settings\User2\My Documents
21-Jan-2010 10:36:48	34	success	560	Object Open	Object Access	Event log tap		ITT-87529F0A8FE\JRiccardi	User2	3 *	C:\Documents and Settings\User2\My Documents
21-Jan-2010 10:36:43	1	success	560	Object Open	Object Access	Event log tap		NT AUTHORITY\SYSTEM	User2	User2.txt	C:\Documents and Settings\User2\My Documents
21-Jan-2010 10:36:39	3	success	560	Object Open	Object Access	Event log tap		ITT-87529F0A8FE\JRiccardi	User2	User2.txt	C:\Documents and Settings\User2\My Documents
21-Jan-2010 10:34:11	2	failure	560	Object Open	Object Access	Event log tap		ITT-87529F0A8FE\JRiccardi	User2	User2-LOCKED.txt	C:\Documents and Settings\User2\My Documents
21-Jan-2010 10:34:08	6	failure	560	Object Open	Object Access	Event log tap		ITT-87529F0A8FE\JRiccardi	User2	User2-LOCKED.txt	C:\Documents and Settings\User2\My Documents

Simple correlation with white listed filtering provides easily understood alert indications.

- – **File access event pattern matches (Windows log text).**
- – **Event number, user, owner and file information are extracted from events**
- – **Event correlator aggregates access to 3 different files with same owner within 30 sec: *3* entry in File name column.* Orange color code denotes multiple files.**
- – **Event correlator detects access by user other than owner. Orange color code in user column highlights this observation.**
- – **Event correlator detects that User3 access privilege has been changed within the last hour. Red color code in User column denotes combination of user-not-owner and user privilege change.**

ADEU “Flag” Lists

- Detection of non-persistent memory executable.
- Generic, configurable capability to assign a flag value based on an event attribute:
  - White-listed application (normal, ignore)
  - Red-listed application (malware)
  - Red-listed document (critical doc)
  - Black-listed IP address (known bad)
  - Yellow-listed user (suspect)
- Lookups executed client-side for false-positive reduction using Prefetch.
- Implemented via ADEU transformation plug-in API.

Proof of Concept Results

- Phase 1 Proof-of-Concept -12 August 2009.
  - Proved that we could deploy an ADEU tap on Windows workstations, extract specific log data elements, normalize to the CEE library format, and then match against our pre-determined attack patterns in near real time.
  - Demonstrated ADEU can extract all log and log-like data elements from Windows workstations as necessary.
- Phase 2 Proof of Concept -18 February 2010.
  - Proved that we can securely parse, extract and normalize CEE selected data elements from multiple network platforms and store for comparison in a simple database for pattern correlation in near real time.

Current/Future Development Steps

- Research
  - Additional platform module and use case research in process.
  - Ability to capture and hash malicious executables and rootkit detection
- Functional Testing
  - HBSS ADEU (AEM) functional testing is currently underway for HBSS integration.
- Phase 3 (Operational Pilots)
  - Pilot deployment of extraction modules on current and additional platform types at various organizations (Fall 2010).
    - Both Windows and Linux workstations will use HBSS deployment mechanisms.
    - Web servers will use ADEU Bridge deployment

Questions

Ms. Kelly Hughes

khughe@nsa.gov

Dr. Bruce Gabrielson (cont.)

bcgabri@nsa.gov