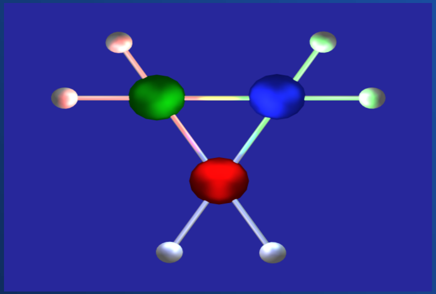
LUARM
An audit engine for Insider misuse detection



George Magklaras

Center for Security, Communications and Network Research

University of Plymouth

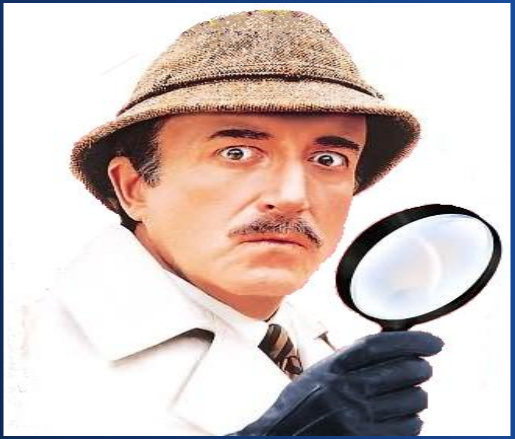
http://www.cscan.org

http://luarm.sourceforge.net

Agenda

- Insider threat specification (what, how, why)
- Requirements for logging insider actions (and forensics)
- LUARM: Architecture and achievements
- What can be facilitated by LUARM and the future

Insiders (visually)



Defining the “insider”

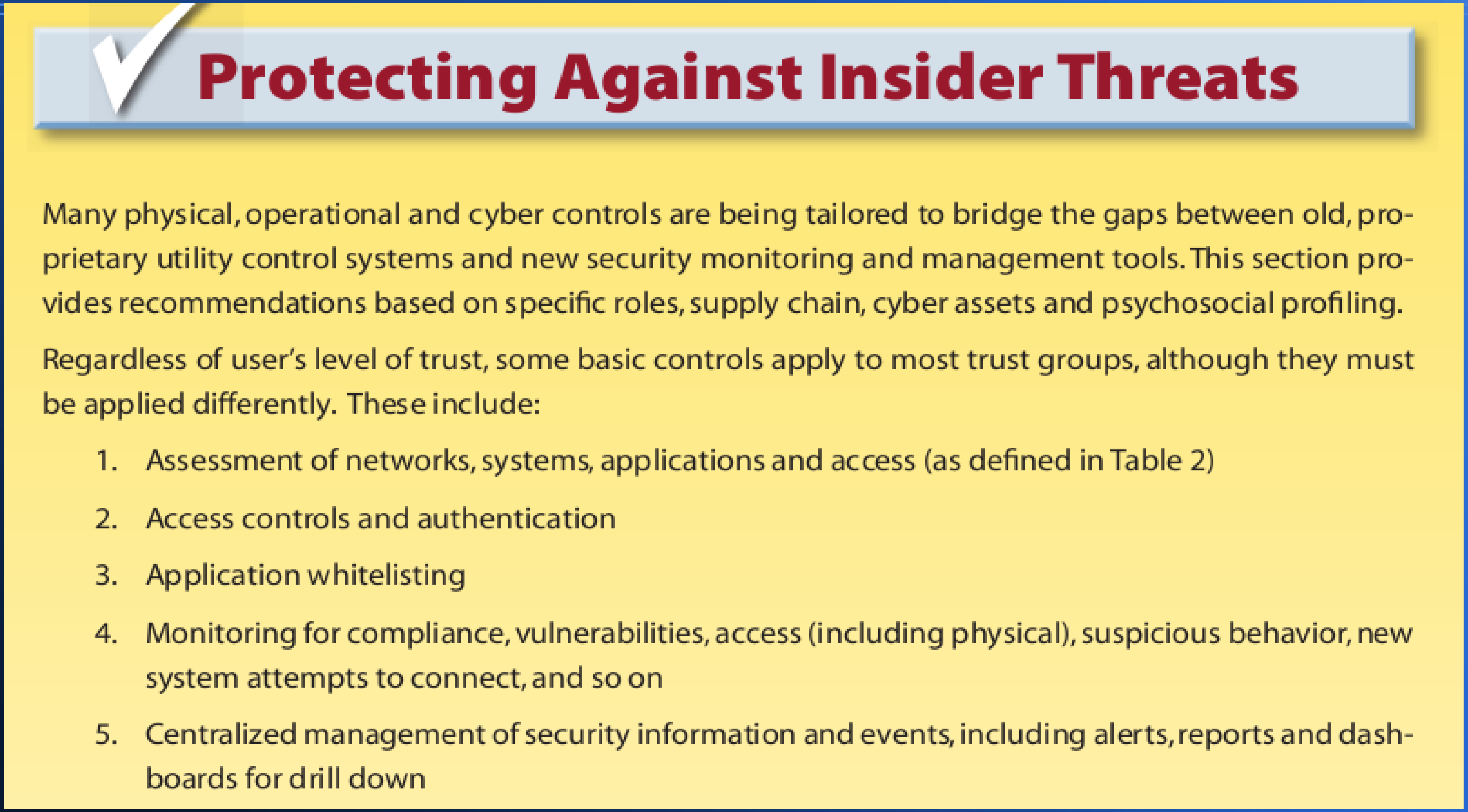
“An insider is a person that has been legitimately empowered with the right to access, represent, or decide about one or more assets of the organization's infrastructure.”

http://www.dagstuhl.de/08302

Defining “Insider Threat Specification”

Insider Threat Specification is the process of using a standardized vocabulary to describe in an abstract way how the aspects and behavior of an insider relate to a security policy defined misuse scenario.

SANS Recommendations



Protecting Against Insider Threats

Many physical, operational and cyber controls are being tailored to bridge the gaps between old, proprietary utility control systems and new security monitoring and management tools. This section provides recommendations based on specific roles, supply chain, cyber assets and psychosocial profiling.

Regardless of user's level of trust, some basic controls apply to most trust groups, although they must be applied differently. These include:

1. Assessment of networks, systems, applications and access (as defined in Table 2)
2. Access controls and authentication
3. Application whitelisting
4. Monitoring for compliance, vulnerabilities, access (including physical), suspicious behavior, new system attempts to connect, and so on
5. Centralized management of security information and events, including alerts, reports and dashboards for drill down

Source: "Managing Insiders in Utility Control Environments" - SANS, Luallen M. E. (2011)

Vital analyst/specialist questions

If I was about to respond to an incident (or even predict it) with a suspected insider and obtain evidence, how would I:

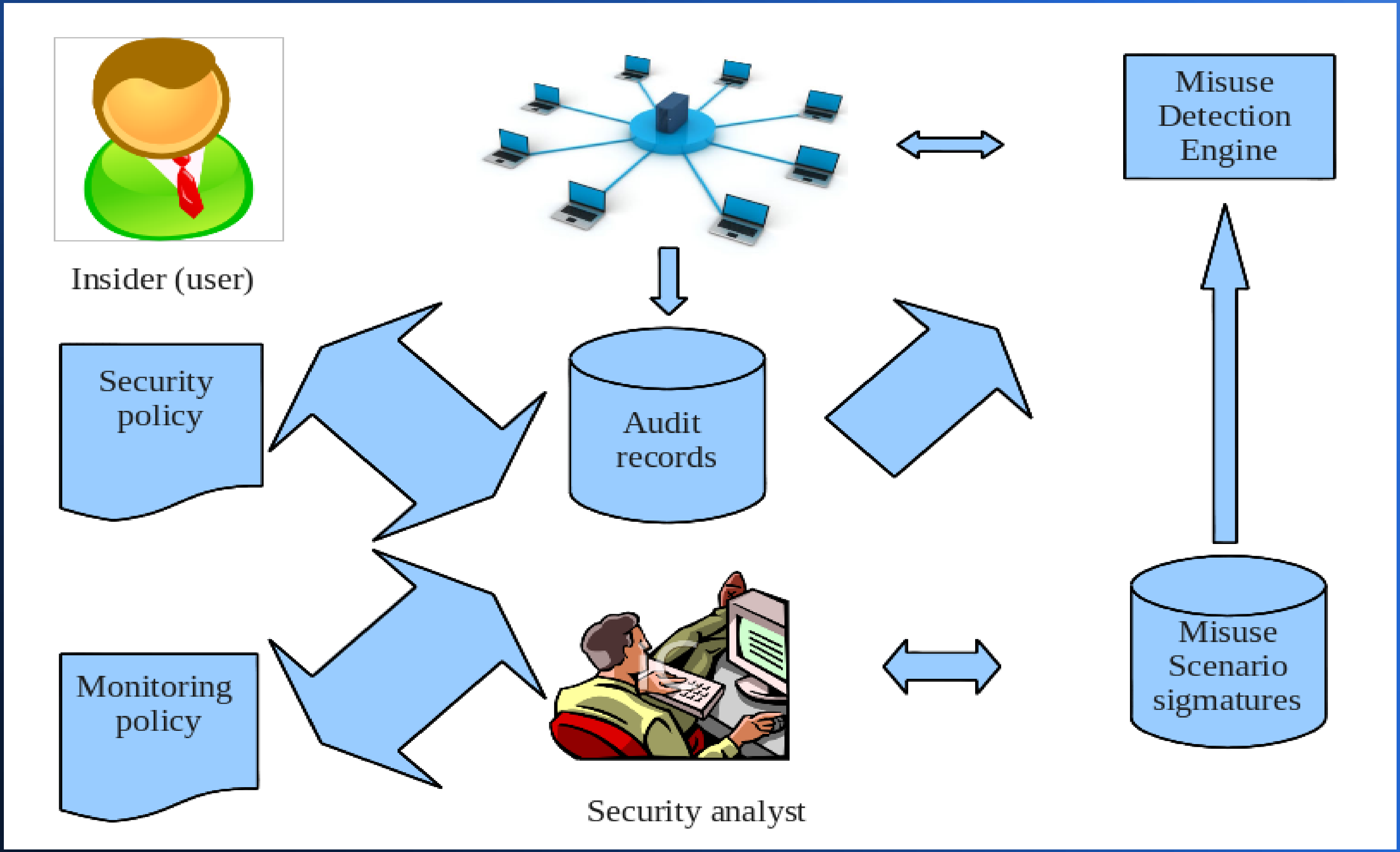
- Assess authentication and access controls?
- Monitor the systems/users in question?
- Present and correlate the collected info to make my case?

Insider Threat Specification monitoring

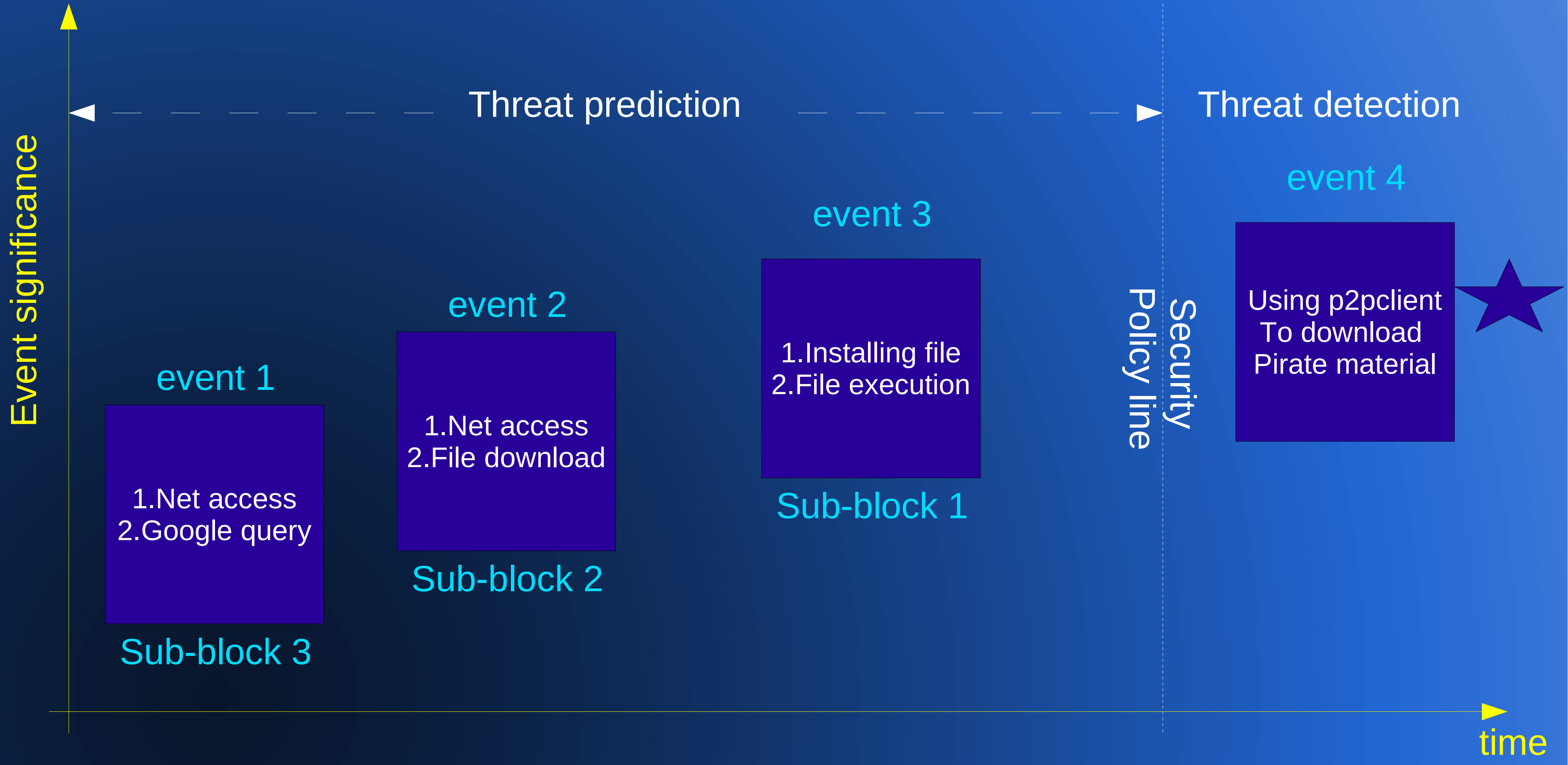
- narrow down the observable aspect and behavioral data to filesystem, process execution, network connection and hardware monitoring levels.

- *“User x was able to launch process b at 16:48:32 which resulted in two connections to websites A and B and as a result left file loic.pro at 16:52:21 in user's x Document area”*

Insider misuse detection information flow



Temporal dimension of an Insider Threat



System-level Insider Threat Logging “wish-list”

- OS agnostic
- The correct timing of records is important.
- Log records should have a well defined format.
- Integrity and availability of log data
- Correlation for user entity accountability.

System-level Insider monitoring and Forensics

Why a logging engine should complement forensics:

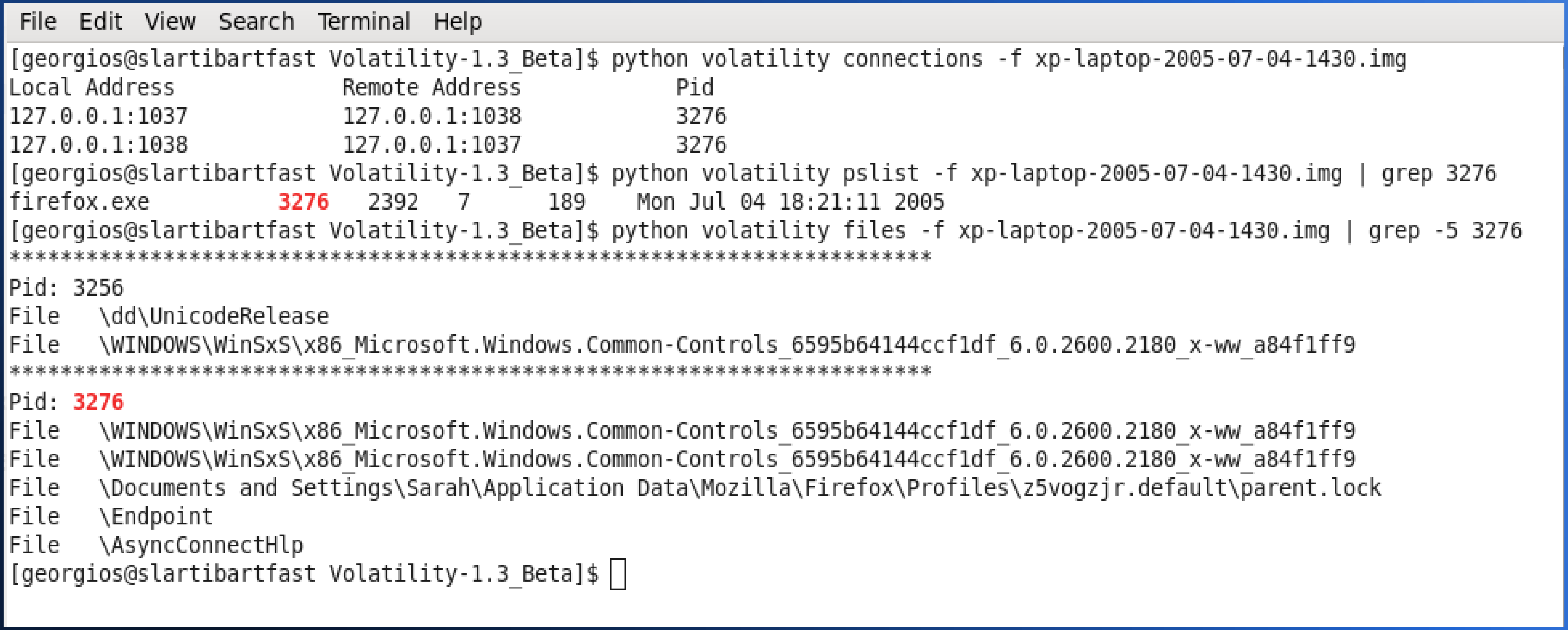
- The “observer effect”: No need to tamper with investigation source media [1].

-”Static” data forensic analysis can give a rather incomplete picture of an incident [1].

-”Dynamic” data forensic analysis (sequence of process events) can be built more easily in a logging engine rather than an OS forensic tool [2].

System-level Insider monitoring and Forensics (2)

Volatile data versus a collection of time-ordered volatile data.



```
[georgios@slartibartfast Volatility-1.3_Beta]$ python volatility connections -f xp-laptop-2005-07-04-1430.img
Local Address             Remote Address            Pid
127.0.0.1:1037            127.0.0.1:1038            3276
127.0.0.1:1038            127.0.0.1:1037            3276
[georgios@slartibartfast Volatility-1.3_Beta]$ python volatility pslist -f xp-laptop-2005-07-04-1430.img | grep 3276
firefox.exe          3276   2392   7      189    Mon Jul 04 18:21:11 2005
[georgios@slartibartfast Volatility-1.3_Beta]$ python volatility files -f xp-laptop-2005-07-04-1430.img | grep -5 3276
************************************************************************
Pid: 3256
File   \dd\UnicodeRelease
File   \WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1ff9
************************************************************************
Pid: 3276
File   \WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1ff9
File   \WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1ff9
File   \Documents and Settings\Sarah\Application Data\Mozilla\Firefox\Profiles\z5vogzjr.default\parent.lock
File   \Endpoint
File   \AsyncConnectHlp
[georgios@slartibartfast Volatility-1.3_Beta]$ 
```

VOLATILE SYSTEMS

Overview of existing logging engines

There are many logging engines/frameworks and Security Event Managers (SEMs) out there. A sample:

- Syslogd[3], WinSyslog[4], RFC 5424
- OpenXDAS [5], Cisco MARS[6]
- Event Data Warehouse [7], Arc Sight Logger 4 [8]

Most of these solutions are geared towards network and application security events and/or data audit compliance.

LUARM

-**L**og **U**ser **A**ctions in **R**elational **M**ode

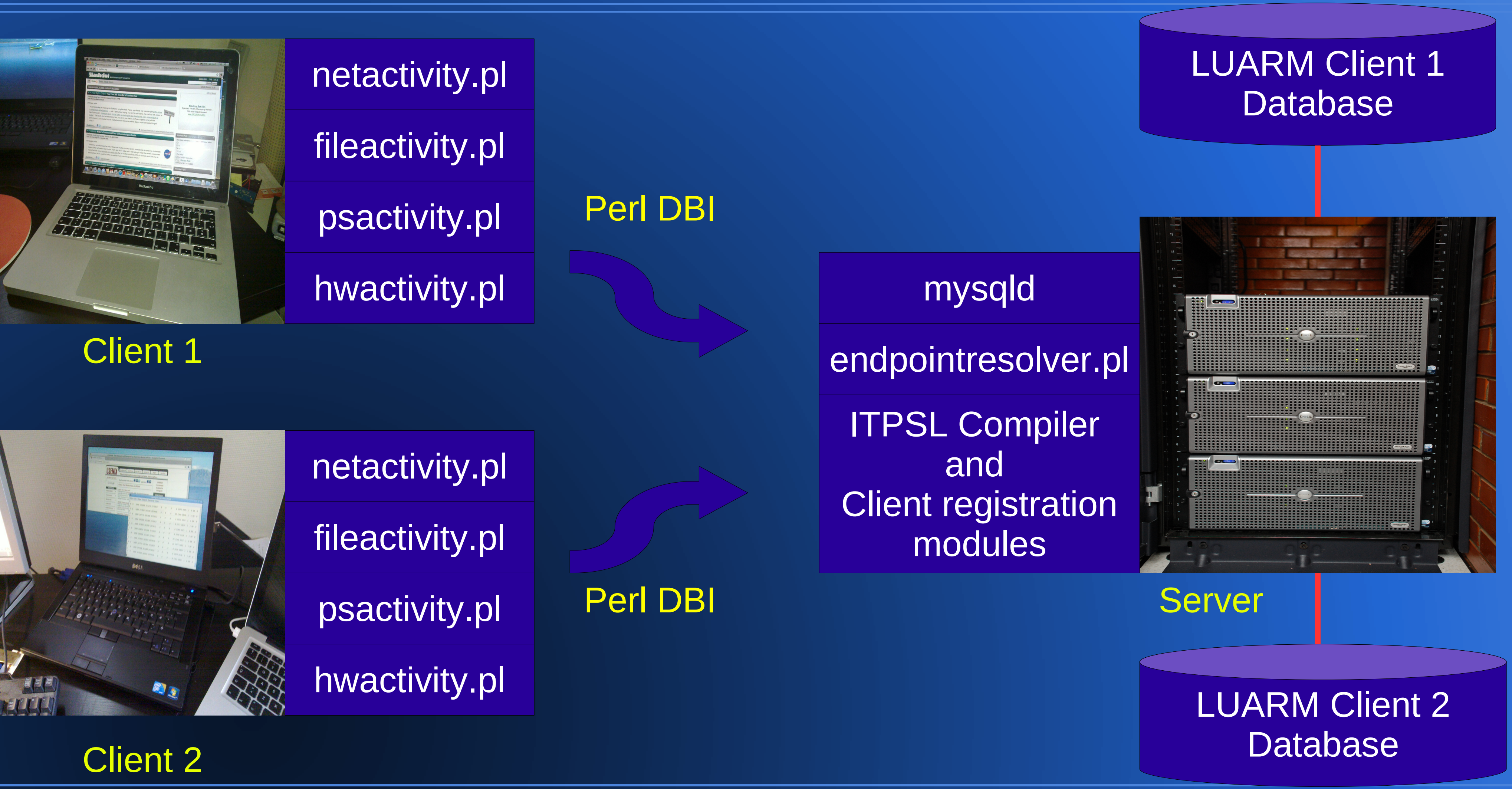
-Written in Perl for rapid prototyping and Open Source.

-Uses MySQL to store the logs in a simple schema.

-Goal: Provide a prototype log engine for insider misuse researchers so that they are:

- able to log user actions in detail.
- able to use the logs to replay/study misuse incidents.
- cross reference logged user data to forensic procedures.

LUARM architecture



netactivity.pl

fileactivity.pl

psactivity.pl

hwactivity.pl

Client 1

netactivity.pl

fileactivity.pl

psactivity.pl

hwactivity.pl

Client 2

Perl DBI

Perl DBI

mysqld

endpointresolver.pl

ITPSL Compiler
and
Client registration
modules

Server

LUARM Client 1
Database

LUARM Client 2
Database

LUARM relational schema

fileaccessid	bigint
md5sum	text
filename	varchar
location	varchar
username	tinytext
application	text
fd	tinytext
pid	int
size	bigint
cyear	int
cmonth	tinyint
cday	tinyint
chour	tinyint
cmin	tintint
csec	tinytint
dyear	int
dmonth	tinyint
dday	tinyint
dhour	tinyint
dmin	tinyint
dsec	tinyint

fileinfo
table

endpointinfo	bigint
md5sum	text
transport	tinytext
sourceip	tinytext
sourcefqdn	tinytext
destip	tinytext
destfqdn	tinytext
sourceport	smallint
destport	smallint
ipversion	smallint
cyear	int
cmonth	tinyint
cday	tinyint
chour	tinyint
cmin	tinyint
csec	tinytint
dyear	int
dmonth	tinyint
dday	tinyint
dhour	tinyint
dmin	tinyint
dsec	Tinyint
usermame	tinytext
pid	int
application	text

netinfo
table

psentity	bigint
md5sum	text
username	tinytext
pid	smallint
ppid	smallint
pcpu	decimal
pmem	decimal
command	text
arguments	mediumtext
cyear	int
cmonth	tinyint
cday	tinyint
chour	tinyint
cmin	tinyint
csec	tinytint
dyear	int
dmonth	tinyint
dday	tinyint
dhour	tinyint
dmin	tinyint
dsec	Tinyint
usermame	tinytext
pid	int

psinfo
table

LUARM query examples

-Find all accesses of the file 'prototype.ppt' by users 'toms' OR 'georgem' between 9:00 and 14:00 hours on 23/10/2009.

```
SELECT * FROM fileinfo WHERE filename='prototype.ppt' AND ((username='toms') OR
(username='georgem')) AND cyear='2009' AND cmonth='10' AND cday='23' AND chour >= '9'
AND chour <= '13' AND cmin >= '0' AND cmin >= '59';
```

-Find all USB devices that were physically connected to the system when users 'toms' OR 'georgem' were logged on 23/10/2009.

```
SELECT * from hwinfo WHERE devbus='usb' AND ((userslogged RLIKE 'toms') OR
(userslogged RLIKE 'georgem')) AND cyear='2009' AND cmonth='10' AND cday='23' AND chour
>= '9' AND chour <= '13' AND cmin >= '0' AND cmin >= '59';
```

Audience alertness test

-What does the following do (hint: psinfo is the process execution table) and what does the sequence of actions of the examples specify?

```
select * FROM psinfo WHERE ((command='cp') OR (command='mv')) AND
(arguments RLIKE 'prototype.ppt' AND arguments RLIKE '/media') AND
((username='georgem') OR (username='toms')) AND cyear='2009' AND cmonth='10'
AND cday='23' AND chour >= '9' AND chour <= '13' AND cmin >= '0' AND cmin >=
'59';
```

LUARM deployment hardware specs

-MySQL LUARM server:

-4 Gbytes of RAM and 4 processing cores

-Disk space consumption in Gigabytes

$$D_{cons} = n_{clients} \times 18 \times d_{archive}$$

Example: 150 clients for 365 days of archiving ~ 1 Tbyte

-Data network: At least 100 Mbits/sec, maximum 20 Kbits/sec per client.

-LUARM client:

-2 processing cores and up to 300 Megs of RAM

-Up to 30% of a single core on a moderately busy system.

LUARM achievements

-A prototype that proves Insider Threat oriented logging is feasible.

-Four large organizations and a number of beta testers (14) have deployed the engine with success in detecting 18 real misuse incidents and even a number of external events.

- Half of these incidents required the assistance of a forensic examiner who used LUARM data for guidance beyond existing forensic tool-kits.

-Deployment feedback indicates that apart from minor bugs, the engine is scalable to provide comprehensive monitoring for up to 300 client systems.

LUARM's future

-Strengthen the prototype security (SSL between client/server, cryptography deployment for temporal data written on disk)

-Windows and OSX platform ports.

-Ability to sanitize the audit data:

- LUARM does not log file contents/network traffic payload.
- still plenty of scope to protect the audit data from misuse (multi party access authentication, pseudo-anonymity of logged records.

-Enabling logging through hypervisor/VM engines.

References

[1] Hay B., Nance K., Bishop M. (2009), “Live Analysis Progress and Challenges”, IEEE Security & Privacy, Volume 7, Number 2, pages 30-37.

[2] Adelstein F. (2006), “Live Forensics: Diagnosing Your System without Killing it First”, Comm. ACM, vol.49, no.2, 2006, pages 63-66.

[3] http://en.wikipedia.org/wiki/Syslogd

[4] http://www.winsyslog.com/en/

[5] The OpenGroup's Distributed Audit System: http://openxdas.sourceforge.net/

[6]Cisco's Monitoring and Analysis Report System: http://www.cisco.com/en/US/products/ps6241/tsd_products_support_reference_guides.html

[7] Event Data Warehouse product: http://www.sensage.com/products/event-data-warehouse.php

[8] Arcsight logger appliance: http://www.arcsight.com/products/products-logger/

[9] Meier M. (2004), “A Model for the Semantics of Attack Signatures in Misuse Detection Systems”, 7th Information Security Conference, LNCS Volume 3225, Springer, Berlin/Heidelberg, pp. 158-169 .

Questions/joining the development

gmagklaras@gmail.com

http://luarm.sourceforge.net/