# Cyber Security for PLM
# *Progress Report*

***Elisa Bertino***
CS Department, Cyber Center, and CERIAS
Purdue University

Joint work with:
**Lorenzo Bossi**, **Syed Hussain, Shagufta Mehnaz**, **Asmaa Sallam**
CS Department and Cyber Center
Purdue University

PURDUE UNIVERSITY

## PLM Project Progress Report

Results to date:

- Research agenda on security for PLM – the agenda has been completed and the agenda document distributed to PLM IB on December 2013
- Protection from insider threat:
  - Design and implementation of an anomaly detection system for relational database management systems (DBMS) supporting role-based access (RBAC) control
    - **the system is able to profile both users and application programs**
  - Development of an interface for the Oracle and SQLServer DBMS
  - *Initial design for file system access profiling and monitoring*
  - *Initial design for profiling data use by users*
- *Initial work on security for embedded systems*

## Anomaly Detection System for Relational Databases

PURDUE UNIVERSITY

## The Insider Threat – Why it is so Dangerous

- Insiders know where the data is and probably know how to access it.
- Outsiders probably don't know as much about what is stored in which systems.  This creates a window of opportunity to detect the intrusion while the outsider is feeling his way around
  - "Feeling your way around" is actually a pattern we can search for
- Insiders may be able to disguise their activities within normal day to day activities
- Requirement
  - We cannot make life difficult for users who are just doing their jobs.

PURDUE UNIVERSITY

## Guiding Recommendation

From “*Spotlight On: Insider Theft of Intellectual Property Inside the United States Involving Foreign Governments or Organizations*”, CMU/SEI, May 2013

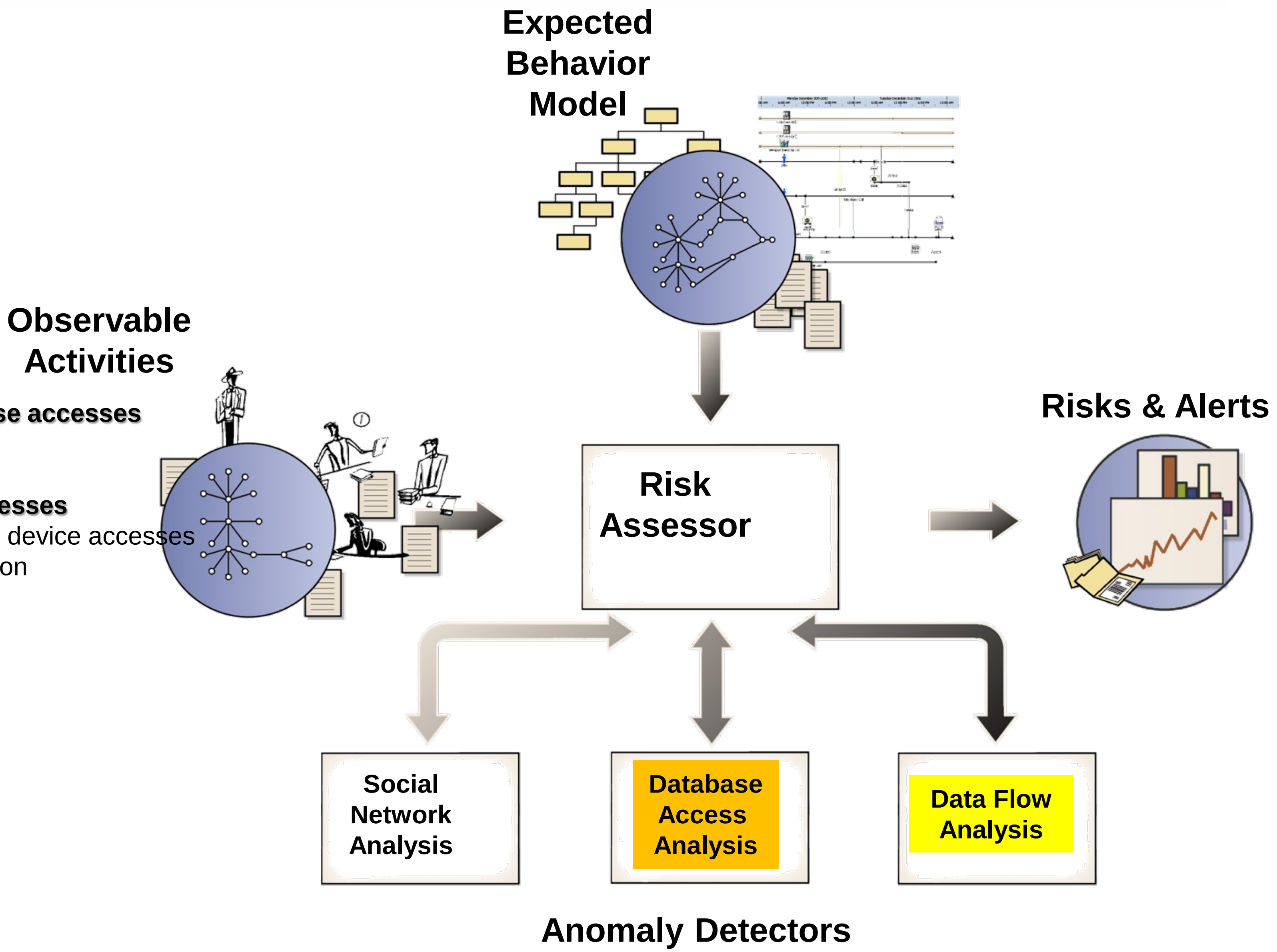
• Recommdendation3:

**Monitor Intellectual Property Leaving the Network**

- •Identify critical information and track its location, access, modification, and transfers
- •Implement technical controls that log the access and movement of critical information that employees
  - •Download from company servers
  - •Email from the organization’s network to personal accounts
  - •Download to removable media
- •Many cases involved downloading source code, executables, or excessive amount of data before leaving the organization

# Our Guiding Idea

- **database accesses**
- printing
- email
- **file accesses**
- external device accesses
- encryption

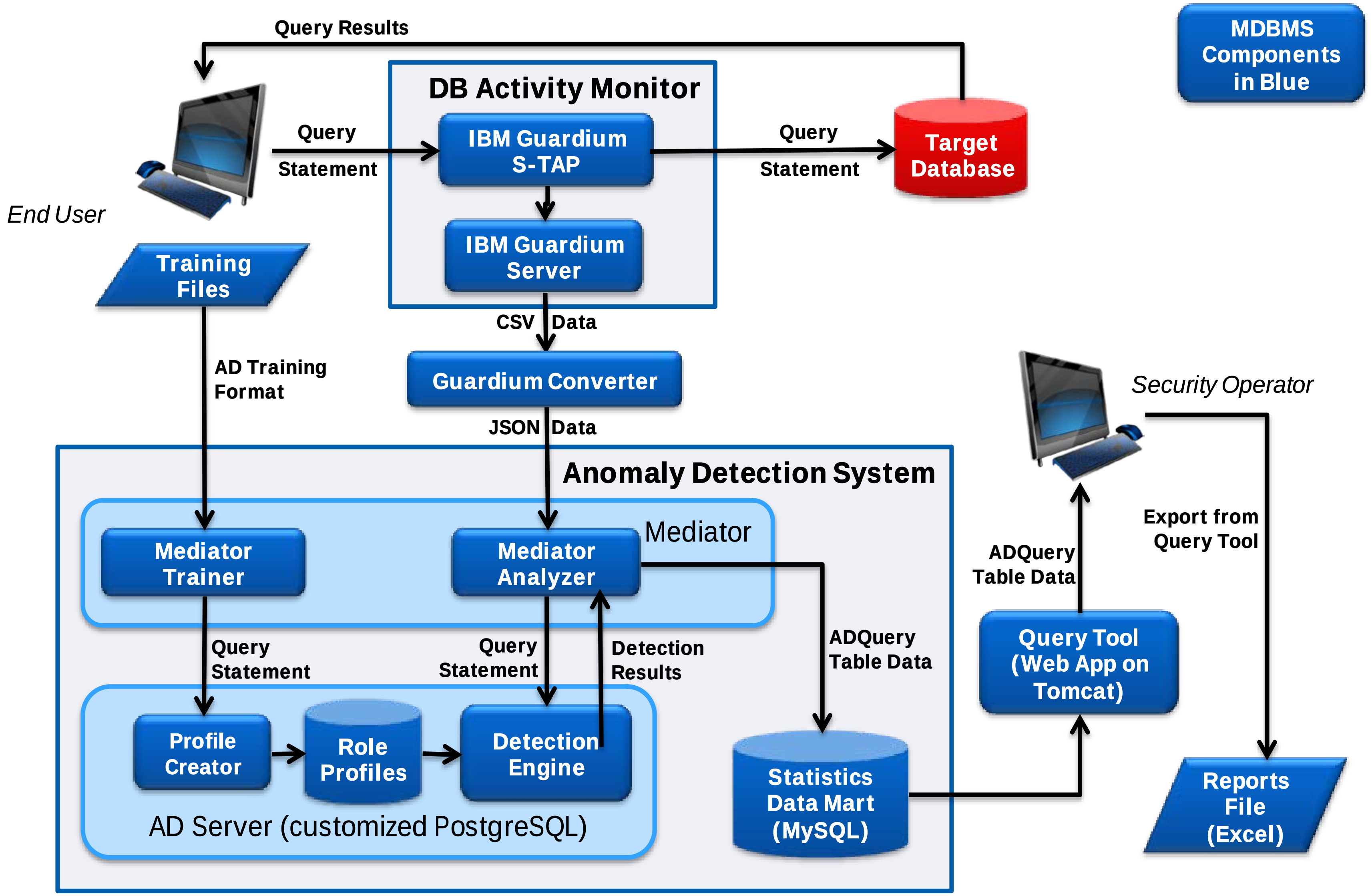


## Approach

- RBAC-administered databases
  - ❑ Access permissions are associated with roles
  - ❑ Users are assigned to roles
- Goal: Detect anomalous database accesses by roles
- Strategy:
  - ❑ Build profiles of normal role behavior
    - o Mine database traces stored in log files
    - o Extract access pattern from queries acquired during a “Training Phase”
    - o Create profiles of roles from queries submitted by users
  - ❑ Use these profiles to detect anomalous behavior (Detection Phase)

PURDUE UNIVERSITY

# System Architecture



## The Classifier

- Creating Profiles ≡ Training the classifier
- “**Classification** is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations”
- We use the NBC (Naïve Bayes Classifier) with the MAP (Max-Aposteriori Probability) decision rule
- Given an input query → Identify which role (most probably) this query came from → Compare it with the actual role of the user submitting the query
- *Recent progresses*
  – Developed and integrated into the system multi-label classification techniques
  – Developed and integrated into the system clustering techniques in order to support the case in which roles are not used

## Characterizing Query Results

- Previous (data-centric) approaches execute query and inspect the output results to compute statistics
- Instead: we use the DBMS **optimizer** to find the selectivity of each table in the query
- The *selectivity* of a table of the query is the portion of the table that is estimated to appear in the query result
  - Range: [0 … 1]
  - e.g., query with sel = 0.2 will retrieve 20% of the table
- Our approach uses the selectivity estimated by the query optimizer in order to profile the amount of data retrieved by the queries and detect anomalies at run-time

## How to Profile and Monitor Application Programs with respect their Database Accesses?

PURDUE UNIVERSITY

## Our Solution: DetAnom

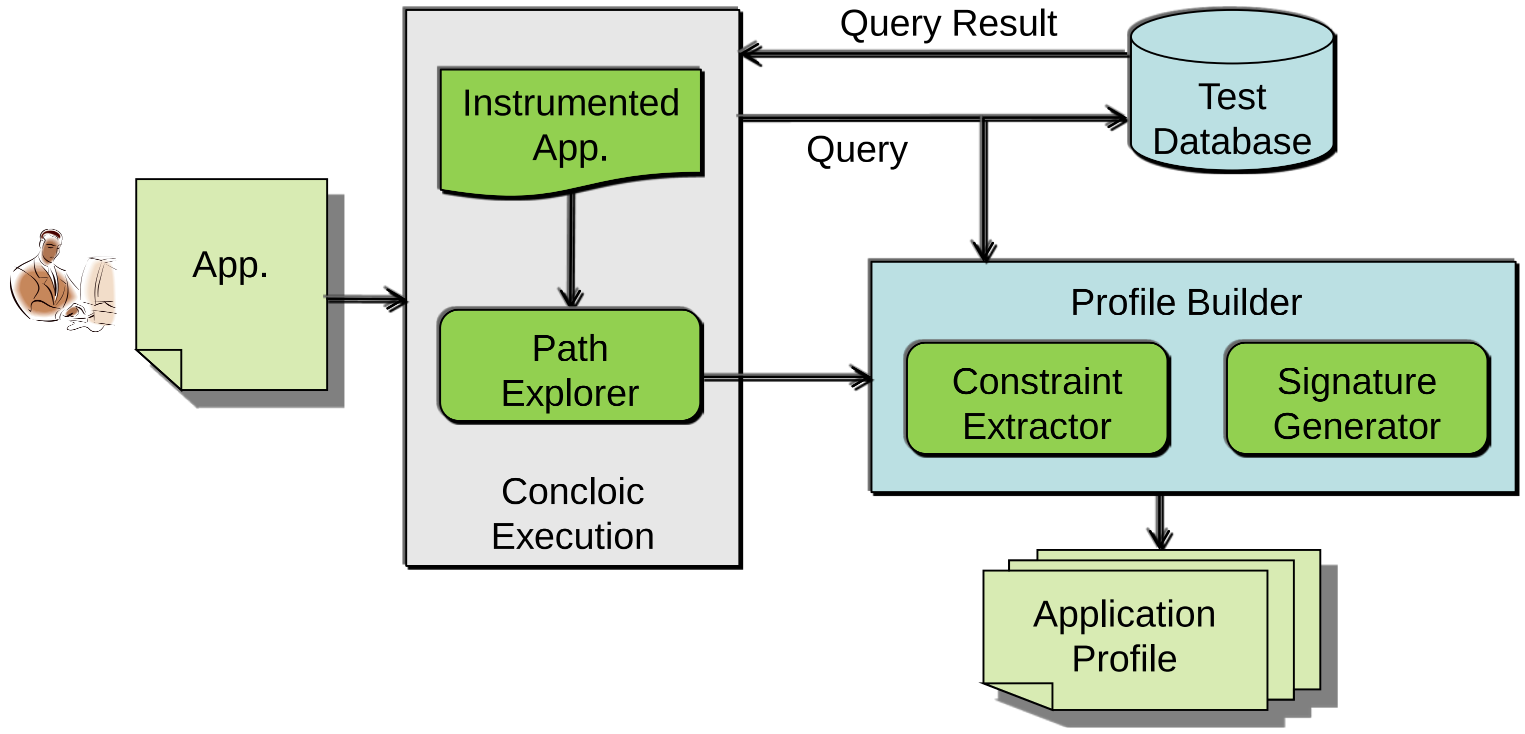
- DetAnom consists of two phases:
  - the *profile creation phase* and the *anomaly detection phase*.
- Profile creation phase:
  - we create a profile of the application program to succinctly represent the application's normal behavior in terms of its interaction with the database.
  - for each query, we create a signature and also capture the corresponding constraints that the application program must satisfy to submit the query.
  - major issue:
    - exploring all possible execution paths of an application program requires identifying all possible combinations of program inputs
    - to make our profiling technique close to complete and accurate, we adopt concolic testing that generates program inputs automatically to cover all execution paths.
- Anomaly detection phase:
  - whenever a query is issued,
    - mismatch in query signature or the constraint -> anomalous
    - otherwise -> legitimate
  - however, depending on the number of paths covered in concolic execution, the *anomaly detection phase* follows either `*strict'* or `*flexible'* policy.

## Concolic Testing

- Concolic testing is a program analysis technique that explores all possible execution paths by running the program both symbolically and concretely.
- The program to be tested is first concretely executed with some initial random inputs.
- Then the concolic execution engine examines the branch conditions along the executed path’s control-flow and uses a decision procedure to find inputs that reverse the branch conditions.
- This process is repeated to discover more inputs that trigger new control-flow paths, and thus more program states are tested.
- The concolic execution uses a bounded depth-first search (bounded DFS) to explore the execution paths.
  - tradeoff between the exploration of more execution paths and termination of the current path if its length is significantly long.

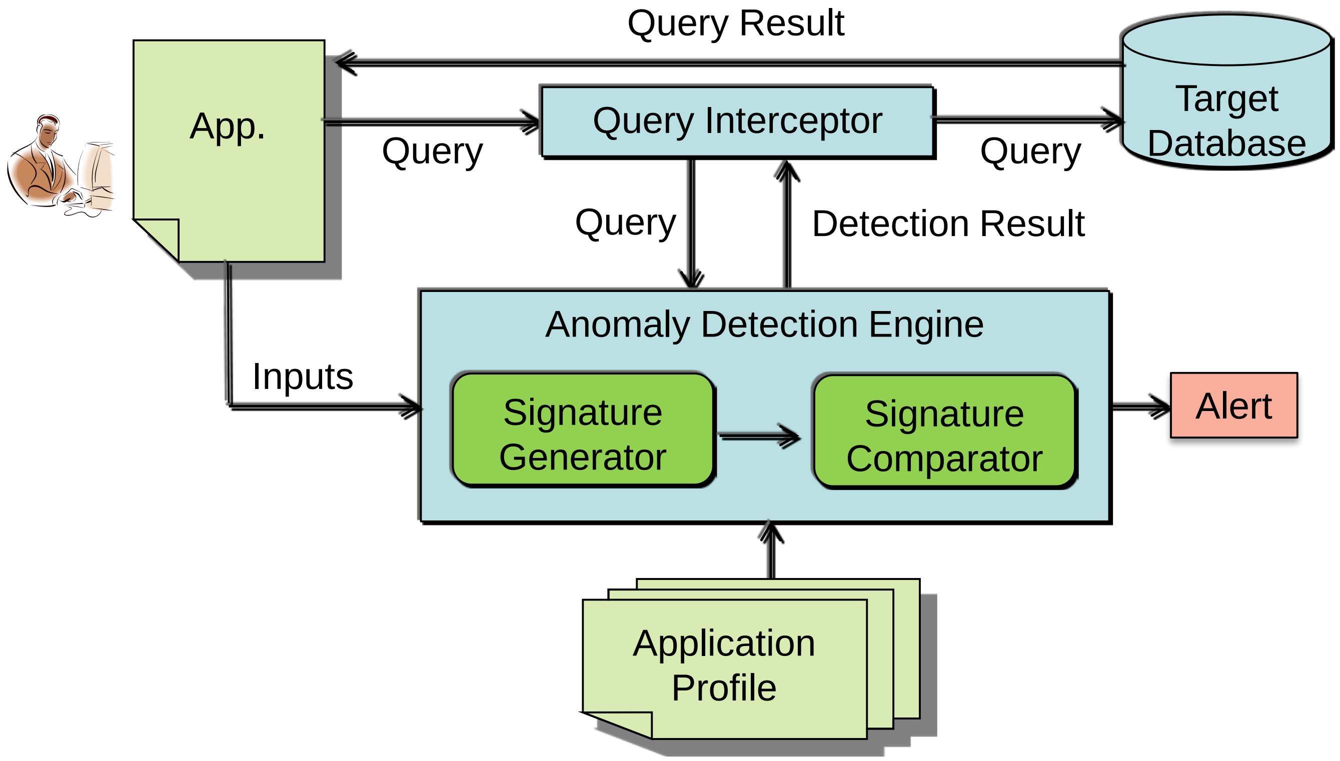
# Profile Creation Phase

The application program is given as input to the concolic execution module



# Anomaly Detection Phase

Queries issued by the application program are first verified by the anomaly detection engine (ADE) and then forwarded to the target database



## Signature Generation

**SQL query structure:**

```
SELECT[DISTINCT] {TARGET-LIST} FROM {RELATION-LIST} WHERE
{QUALIFICATION}
```

| Table ID | Table name | Attribute ID | Attribute name | Type |
|---|---|---|---|---|
| 100 | PersonalInfo | 1 | employee_id | varchar(10) |
| | | 2 | employee_name | varchar(50) |
| 200 | WorkInfo | 1 | employee_id | varchar(10) |
| | | 2 | work_experience | number |
| | | 3 | salary | number |
| | | 4 | performance | varchar(20) |

**Example:** SELECT employee_id, work_experience FROM WorkInfo
WHERE work_experience > 10

**Signature:** {1, {{200, 1}, {200, 2}}, {200}, {{200, 2}}, 1}

- The leftmost 1 represents the SELECT command.
- {200, 1}, and {200, 2} represent the IDs of attributes employee_id and work_experience, respectively.
- 200 represents the ID of the table WorkInfo.
- {200, 2} represents the attribute used in the WHERE clause, i.e, work_experience.
- The rightmost 1 corresponds to the number of predicates in WHERE clause.

PURDUE UNIVERSITY

# Constraint Extraction

```
public static void salaryAdjustment(int profit, int
     investment){
    Statement s;
    ...
    int employee_count = 0;
    if(profit >= 0.5 * investment){
        String query1 = "SELECT employee_id,
             work_experience FROM WorkInfo WHERE
             work_experience > 10";
        resultSet1 = s.executeQuery(query1);
        resultSet1.last();
        if(resultSet1.getRow() > 100){
        String query3 = "SELECT employee_id FROM
             WorkInfo WHERE work_experience > 10 AND
             performance = 'good'";
        resultSet3 = s.executeQuery(query3);
        ... // do other operations
        } else{
            String query2 = "UPDATE WorkInfo SET salary
                 = salary * 1.2";
            s.executeUpdate(query2);
    }else{
        String query4 = "SELECT p.employee_name FROM
             PersonalInfo p, WorkInfo w WHERE
             performance = 'poor' AND p.employee_id =
             w.employee_id";
        resultSet2 = s.executeQuery(query4);
        ... // do other operations
    }
}
```

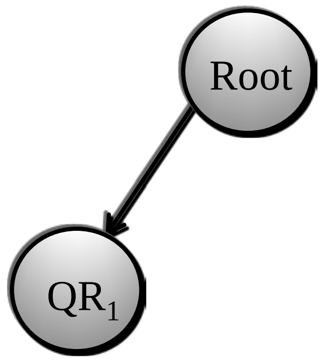
$c_1$: `1.0 x1 – 0.5 x2 >= 0.0,`
Here, x1 and x2 correspond to variables `profit` and `investment`, respectively.

# Profile Creation

```
public static void salaryAdjustment(int profit, int
     investment){
    Statement s;
    ...
    int employee_count = 0;
    if(profit >= 0.5 * investment){
        String query1 = "SELECT employee_id,
             work_experience FROM WorkInfo WHERE
             work_experience > 10";
        resultSet1 = s.executeQuery(query1);
        resultSet1.last();
        if(resultSet1.getRow() > 100){
        String query3 = "SELECT employee_id FROM
             WorkInfo WHERE work_experience > 10 AND
             performance = 'good'";
        resultSet3 = s.executeQuery(query3);
        ... // do other operations
        } else{
            String query2 = "UPDATE WorkInfo SET salary
                 = salary * 1.2";
            s.executeUpdate(query2);
    }else{
        String query4 = "SELECT p.employee_name FROM
             PersonalInfo p, WorkInfo w WHERE
             performance = 'poor' AND p.employee_id =
             w.employee_id";
        resultSet2 = s.executeQuery(query4);
        ... // do other operations
    }
}
```

$c_1$: 1.0 $x1$ – 0.5 $x2$ >= 0.0
sig(query$_1$)= {1, {{200, 1}, {200, 2}}, {200}, {{200, 2}}, 1}
QR$_1$ = <sig(query$_1$), $c_1$>

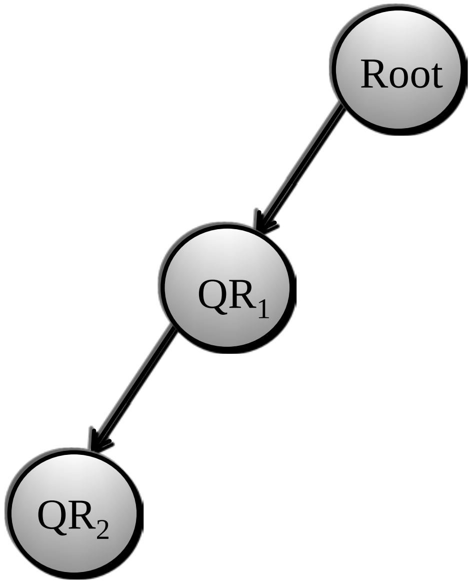
Application Profile



# Profile Creation

```
public static void salaryAdjustment(int profit, int
     investment){
    Statement s;
    ...
    int employee_count = 0;
    if(profit >= 0.5 * investment){
        String query1 = "SELECT employee_id,
             work_experience FROM WorkInfo WHERE
             work_experience > 10";
        resultSet1 = s.executeQuery(query1);
        resultSet1.last();
        if(resultSet1.getRow() > 100){
        String query3 = "SELECT employee_id FROM
             WorkInfo WHERE work_experience > 10 AND
             performance = 'good'";
        resultSet3 = s.executeQuery(query3);
        ... // do other operations
        } else{
            String query2 = "UPDATE WorkInfo SET salary
                 = salary * 1.2";
            s.executeUpdate(query2);
    }else{
        String query4 = "SELECT p.employee_name FROM
             PersonalInfo p, WorkInfo w WHERE
             performance = 'poor' AND p.employee_id =
             w.employee_id";
        resultSet2 = s.executeQuery(query4);
        ... // do other operations
    }
}
```

Application Profile



$c_2$: x3 ≤ 100.0
sig(query2)={2, {{200, 3}}, {200}, {∅}, 0}
$QR_2$ = <sig($query_2$), $c_2$>

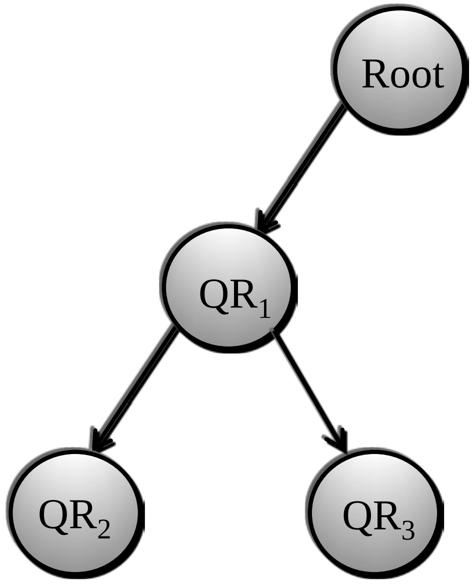
PURDUE UNIVERSITY

# Profile Creation

```
public static void salaryAdjustment(int profit, int
     investment){
    Statement s;
    ...
    int employee_count = 0;
    if(profit >= 0.5 * investment){
        String query1 = "SELECT employee_id,
             work_experience FROM WorkInfo WHERE
             work_experience > 10";
        resultSet1 = s.executeQuery(query1);
        resultSet1.last();
        if(resultSet1.getRow() > 100){
        String query3 = "SELECT employee_id FROM
             WorkInfo WHERE work_experience > 10
             performance = 'good'";
        resultSet3 = s.executeQuery(query3);
        ... // do other operations
        } else{
            String query2 = "UPDATE WorkInfo SET salary
                 = salary * 1.2";
            s.executeUpdate(query2);
    }else{
        String query4 = "SELECT p.employee_name FROM
             PersonalInfo p, WorkInfo w WHERE
             performance = 'poor' AND p.employee_id =
             w.employee_id";
        resultSet2 = s.executeQuery(query4);
        ... // do other operations
    }
}
```

$c_3$: x3 > 100.0
sig(query$_3$)={1, {{200, 1}}, {200}, {{200, 2}, {200, 4}}, 2}
QR$_3$ = <sig(query$_3$), $c_3$>

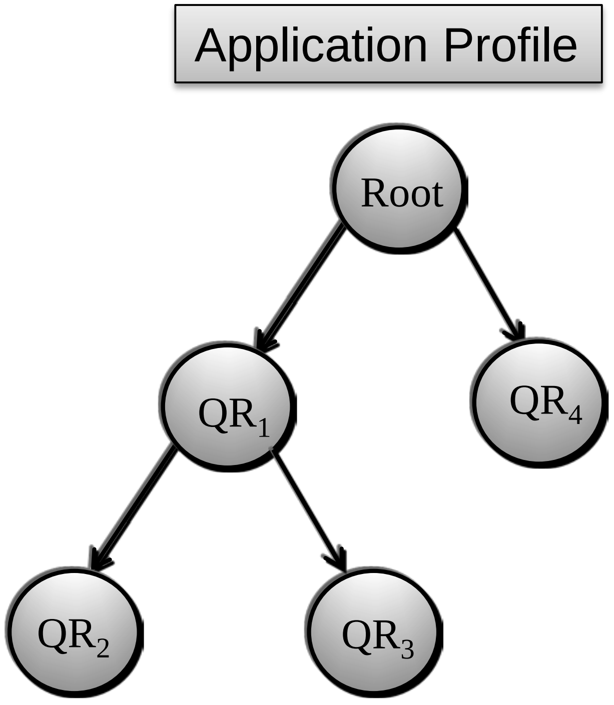
Application Profile



PURDUE UNIVERSITY

# Profile Creation

```
public static void salaryAdjustment(int profit, int
     investment){
    Statement s;
    ...
    int employee_count = 0;
    if(profit >= 0.5 * investment){
        String query1 = "SELECT employee_id,
             work_experience FROM WorkInfo WHERE
             work_experience > 10";
        resultSet1 = s.executeQuery(query1);
        resultSet1.last();
        if(resultSet1.getRow() > 100){
        String query3 = "SELECT employee_id FROM
             WorkInfo WHERE work_experience > 10 AND
             performance = 'good'";
        resultSet3 = s.executeQuery(query3);
        ... // do other operations
        } else{
            String query2 = "UPDATE WorkInfo SET salary
                 = salary * 1.2";
            s.executeUpdate(query2);
    }else{
        String query4 = "SELECT p.employee_name FROM
             PersonalInfo p, WorkInfo w WHERE
             performance = 'poor' AND p.employee_id =
             w.employee_id";
        resultSet2 = s.executeQuery(query4);
        ... // do other operations
    }
}
```



$c_4$: 1.0 $x1$ – 0.5 $x2$ < 0.0
sig($query_4$)= {1, {{100, 2}}, {100, 200}, {{200, 4}, {100, 1}, {200, 1}}, 2}
$QR_4$ = <sig($query_4$), $c_4$>

## Anomaly Detection

- When the application program starts executing, the ADE module sets the root node of the AP as the parent node ($v_p$).
- Upon receiving a query along an execution path of the program, the ADE:
  - considers all the children of $v_p$ as candidate nodes
  - takes the inputs from the executing application
  - verifies for each candidate node whether the inputs satisfy the constraint in $QR_i$.
  - lets the SG sub-module generate the signature of the received query and the SC sub-module compare it with the signature stored in $QR_i$, i.e., sig(query$_i$).
  - checks if the inputs satisfy constraint $c_i$ of a candidate $QR_i$
  - expects the program to execute the query associated with the satisfied $c_i$.
- If the signatures match, the query is considered as legitimate.
  - the verification outcome is then passed to the QI module which then sends the legitimate query to the target database for execution.
- If the signatures mismatch, the query is considered as anomalous.
  - the *SC* sub-module raises a flag and the *ADE* takes next steps based on either `*strict*' or `*flexible*‘ policies.

## How to Profile Data Flows and Detect Anomalous Data Flows?

PURDUE UNIVERSITY

## Scenarios

### Profile Creation

**Scenario**

1- User1 accesses Postgres and redirects data to a file
\0 /file/path/**file1.txt**
SELECT …

2- User1 Prints the file using *lp* command
*lp* **file1.txt**

**Action**

Add **file1.txt** to Monitored List.

Add to *Provenance storage* that User1 usually prints data

### Detection Phase

**Scenario**

1- User1 accesses Postgres
\0 /file/path/**file2.txt**
SELECT …

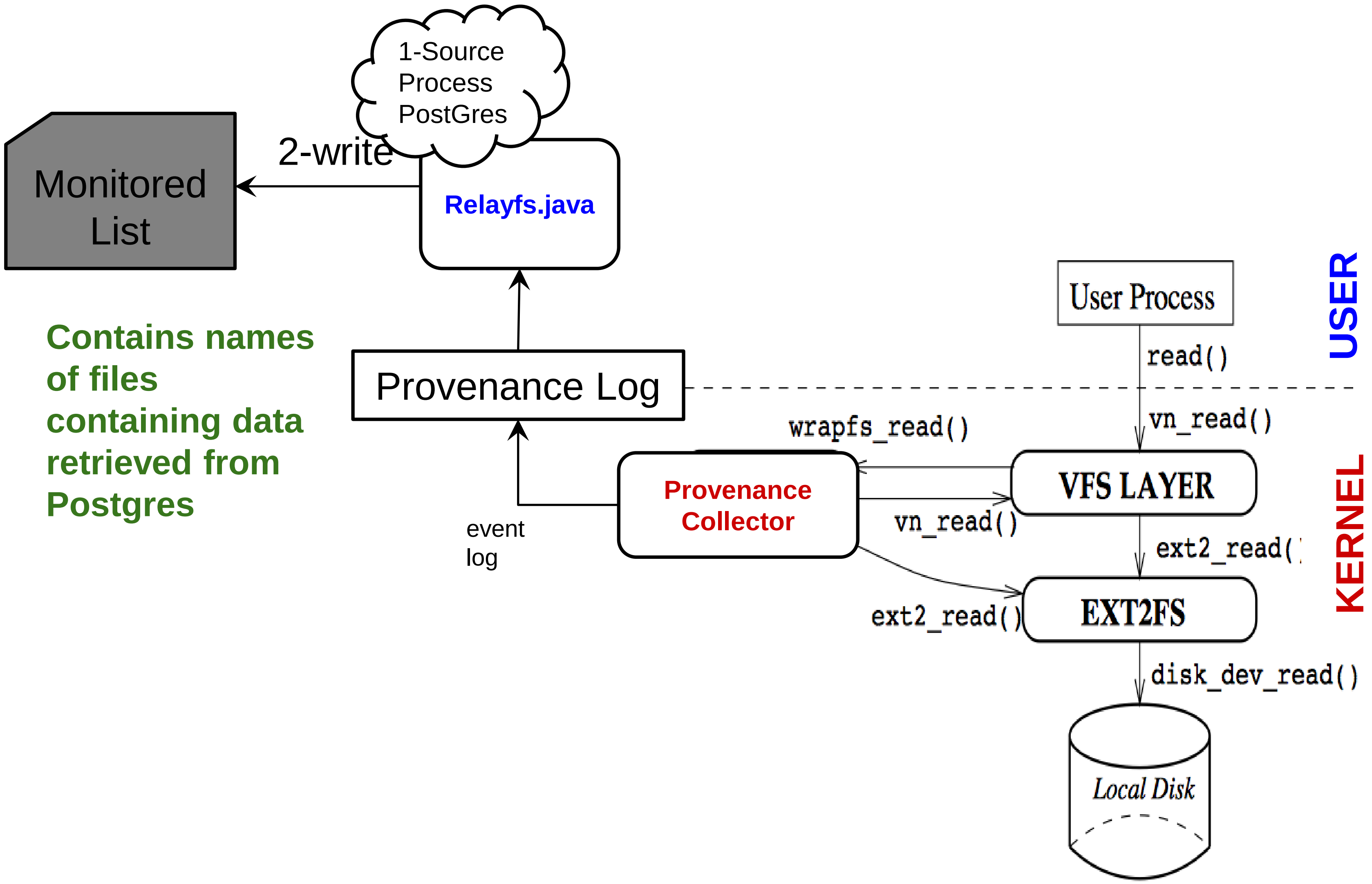
2- User1 Emails the file using *mail/mut* command
*mail* **file2.txt**

**Action**

Add **file2.txt** to monitored list.

Since user1 usually prints data from PG not emails, this action is malicious.

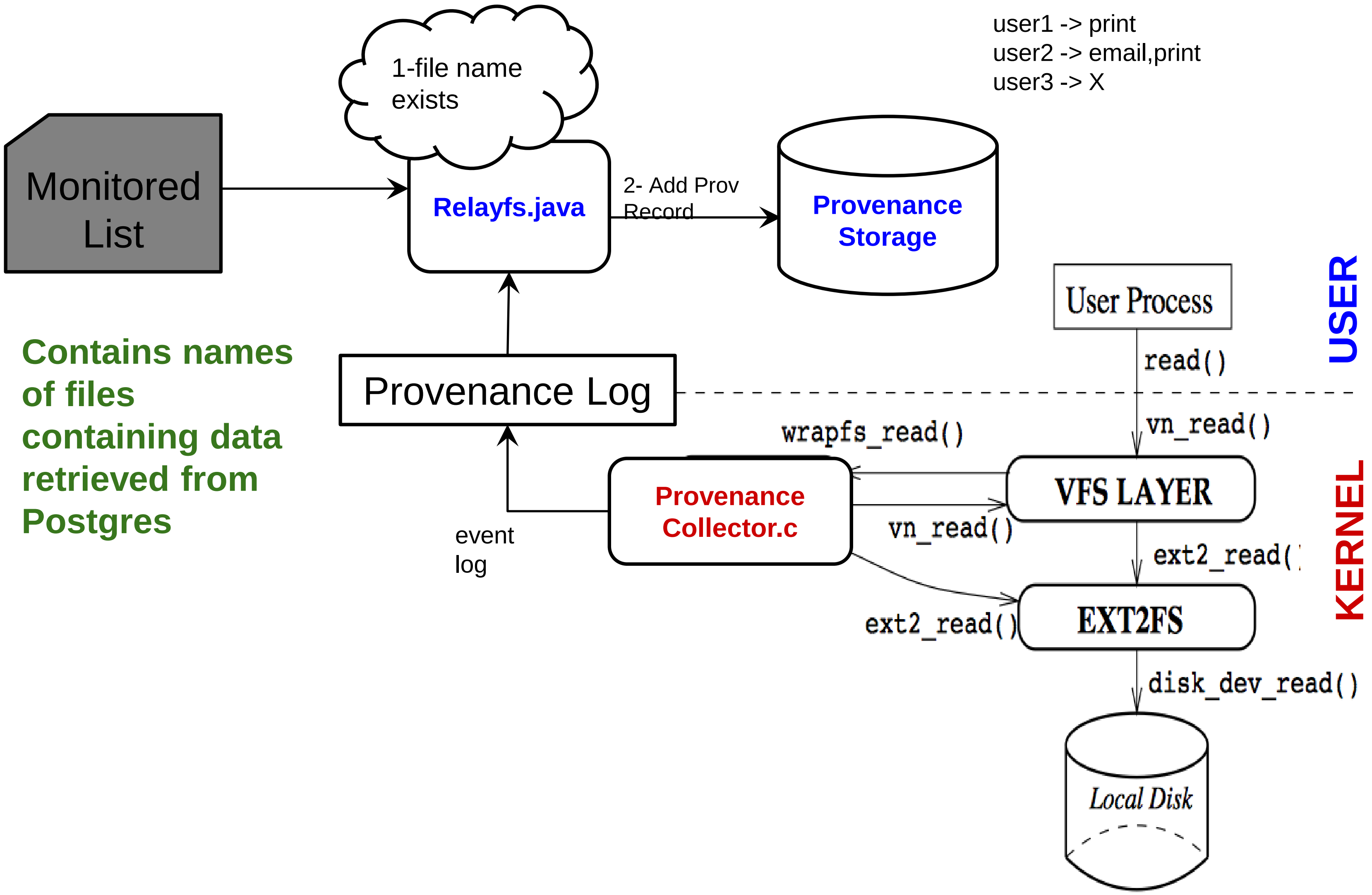
# Profile Creation 1



Contains names of files containing data retrieved from Postgres

# Profile Creation 2

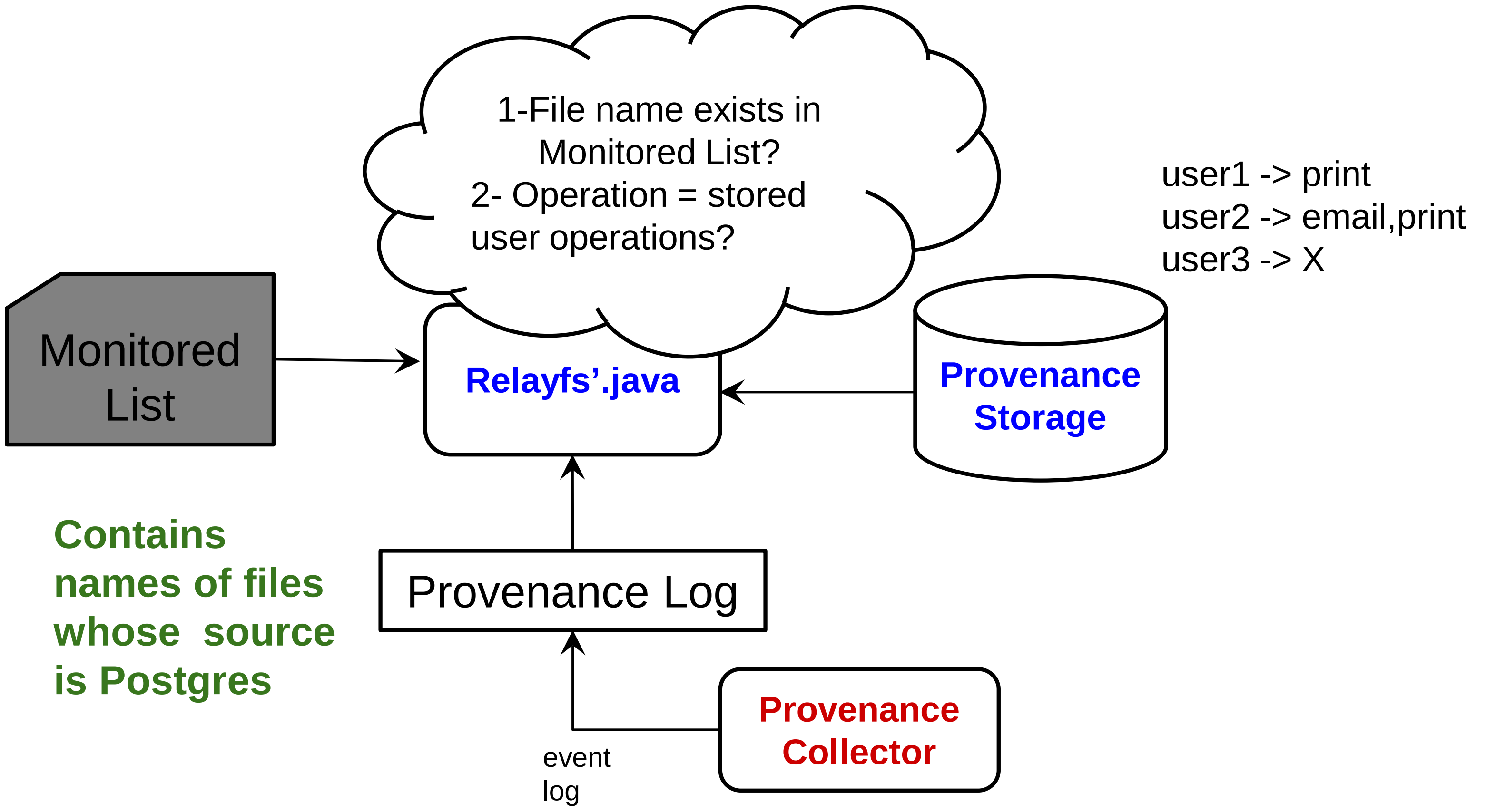
user1 -> print
user2 -> email,print
user3 -> X



Contains names of files containing data retrieved from Postgres

PURDUE UNIVERSITY

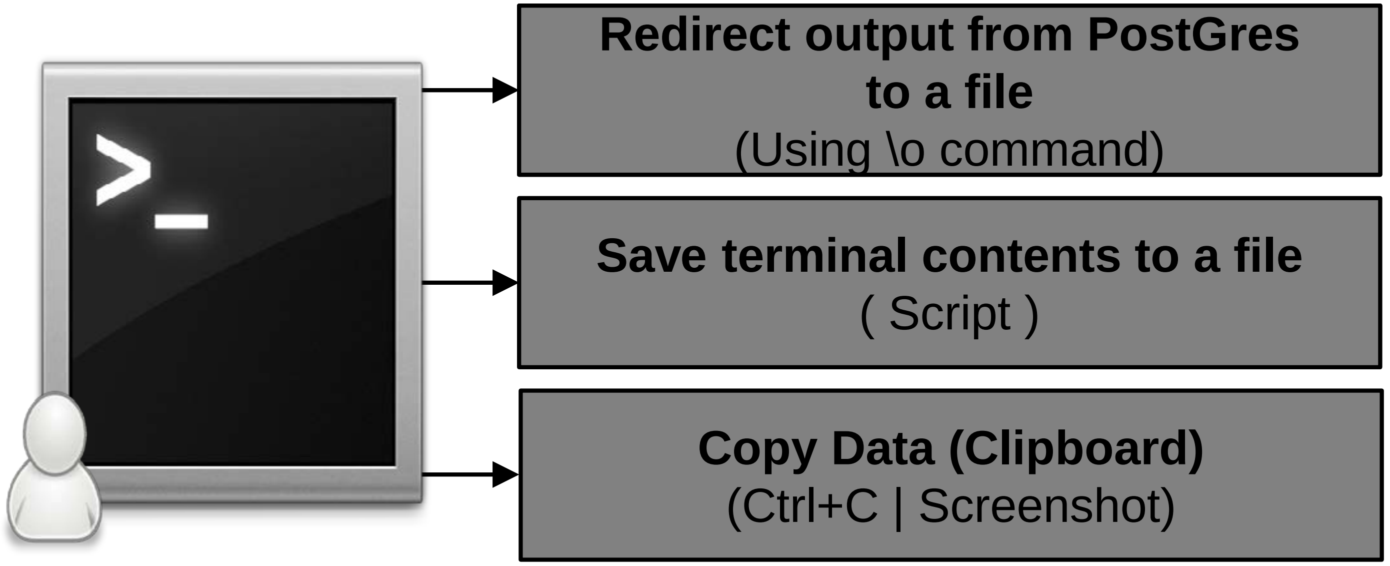
# Detection Phase



Contains names of files whose source is Postgres

PURDUE UNIVERSITY

## Creation of the Monitored List
## *Scenarios*



**Redirect output from PostGres to a file**
(Using \o command)

**Save terminal contents to a file**
( Script )

**Copy Data (Clipboard)**
(Ctrl+C | Screenshot)

PURDUE UNIVERSITY

## On-Going and Future Work

- Track what the program/user does with the data
- Apply our anomaly detection techniques to data stores other than relational databases
- Take query frequencies into account
- Consider profiles evolving with time/tasks

## Insider Threat in Critical Infrastructures

## Definitions

The President’s National Infrastructure Advisory Council defines the insider threat as follows:

> “The insider threat to critical infrastructure is one or more individuals with the access or inside knowledge of a company, organization, or enterprise that would allow them to exploit the vulnerabilities of that entity’s security, systems, services, products, or facilities with the intent to cause harm.”
>
> “A person who takes advantage of access or inside knowledge in such a manner commonly is referred to as a “malicious insider.””

Definitions from FEMA – Emergency Management Institute
http://www.training.fema.gov/emi.aspx

PURDUE UNIVERSITY

## The Scope of Insider Threats

Insider threats can be accomplished through either physical or cyber means and may involve any of the following:

| Threat | Involves |
|---|---|
| **Physical or information-technology sabotage** | Modification or damage to an organization's facilities, property, assets, inventory, or systems with the purpose of harming or threatening harm to an individual, the organization, or the organization's operations |
| **Theft of intellectual property** | Removal or transfer of an organization's intellectual property outside the organization through physical or electronic means (also known as economic espionage) |
| **Theft or economic fraud** | Acquisition of an organization's financial or other assets through theft or fraud |
| **National security espionage** | Obtaining information or assets with a potential impact on national security through clandestine activities |

## Examples of Actual Incidents

| Sector | Incidents |
|---|---|
| **Chemical** | **Theft of intellectual property.** A senior research and development associate at a chemical manufacturer conspired with multiple outsiders to steal proprietary product information and chemical formulas using a USB drive to download information from a secure server for the benefit of a foreign organization. The conspirator received $170,000 over a period of 7 years from the foreign organization. |
| **Commercial Facilities** | **Theft of intellectual property.** A consultant in the commercial facilities industry downloaded the organization’s proprietary software and, upon termination, tried to sell it to another organization for nearly $7 million.<br><br>**Theft of intellectual property.** The supervisor of a maintenance crew with access to every office at a research and development facility surreptitiously collected diskettes, blueprints, and other types of confidential research information, and then attempted to sell the information to a rival company. |
| **Communications** | **Economic fraud.** A group of insiders at a wireless telecommunications firm created clones of more than 16,000 customer cell phones. Over a period of 6 months, the insiders made approximately $15 million worth of unauthorized calls, many of which were international. |
| **Critical Manufacturing** | **Physical sabotage.** A disgruntled employee entered a manufacturing warehouse after duty hours and destroyed more than a million dollars of equipment and inventory. |

PURDUE UNIVERSITY

## Examples of Actual Incidents (cont'd)

| Sector | Incidents |
|---|---|
| **Defense Industrial Base** | **National security threats.** Two individuals, working as defense contractors and holding U.S. Government security clearances, were convicted of spying for a foreign government. For over 20 years, they stole trade and military secrets, including information on advanced military technologies.<br><br>**Information-technology sabotage.** A system administrator served as a subcontractor for a defense contract company. After being terminated, the system administrator accessed the system and important system files, causing the system to crash and denying access to over 700 employees. |
| **Emergency Services** | **Information-technology sabotage.** An information technology employee in a telecommunications company that ran an emergency 911 system brought down the system by deleting data on three servers that handled emergency calls. The employee then stole over 50 backup tapes to further amplify the attack. |
| **Energy** | **Information-technology sabotage.** An oil-exploration company hired a temporary consultant to assist in setting up a control system that enabled communication with offshore platforms and detection of pipeline leaks. The consultant's request for permanent employment was rejected and the contract ended. During the 2 months following termination, the ex-employee actively planted malicious programs on the organization's systems and temporarily disabled the control system. |

# Examples of Actual Incidents (cont'd)

| Sector | Incidents |
|---|---|
| **Transportation Systems** | **Information-technology sabotage.** Two employees sabotaged the system controlling the traffic lights of a major city. The sabotage took 4 days to fix, greatly affecting traffic. |
| **Water and Wastewater Systems** | **Information-technology sabotage.** An electrical supervisor developed executable program applications for a supervisory control and data acquisition (SCADA) system used by the water industry. After termination, the ex-employee installed a malicious program on one of the organization's critical systems, damaging the SCADA system.<br><br>**Physical sabotage.** A water treatment plant employee allegedly manually shut down operating systems at a wastewater utility in an attempt to cause a sewage backup to damage equipment and create a buildup of methane gas. |

## Organizational Factors that Embolden Malicious Insiders

***Access and Availability***

- Ease of access to materials and information
- Ability to exit the facility or network with materials or information

***Policies and Procedures***

- Undefined or inadequate policies and procedures
- Inadequate labeling
- Lack of Training

Time Pressure and Consequences

- Rushed employees
- Perception of lack of consequences

# Questions???