1st Line of Defense for Databases

ORACLE

James Sadler, CISSP/CISA
Oracle NZ Security Specialist

Safe Harbour Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

- Defining the problem?
- Do we need another firewall?
- The Oracle Database Firewall

Why did I choose this topic?

- Citibank
- Lockheed Martin
- HB Gray
- Sony Music
- Sony Playstation Network
- EMC/RSA
- Epsilon
- Sega
- 7-Eleven
- Fox TV

How is Data Compromised?

verizon business

2010 Data Breach Investigations Report

WHO IS BEHIND DATA BREACHES?

70% resulted from external agents (-9%)

48% were caused by insiders (+26%)

11% implicated business partners (-23%)

27% involved multiple parties (-12%)

Including the USSS cases in this year's report shook things up a bit but didn't shake our worldview. Driven largely by organized groups, the majority of breaches and almost all data stolen (98%) in 2009 was still the work of criminals outside the victim organization. Insiders, however, were more common in cases worked by the USSS, which boosted this figure in the joint dataset considerably. This year's study has by far improved our visibility into internal crime over any other year. Breaches linked to business partners continued the decline observed in our last report and reached the lowest level since 2004.

HOW DO BREACHES OCCUR?

Related to the larger proportion of insiders, Misuse sits atop the list of threat actions leading to breaches in 2009. That's not to say that Hacking and Malware have gone the way of the dinosaurs; they ranked #2 and #3 and were responsible for over 95% of all data comprised. Weak or stolen credentials, SQL injection, and data-capturing, customized malware continue to plague organizations trying to protect information assets. Cases involving the use of social tactics more than doubled and physical attacks like theft, tampering, and surveillance ticked up several notches.

48% involved privilege misuse (+26%)

40% resulted from hacking (-24%)

38% utilized malware (<>)

28% employed social tactics (+16%)

15% comprised physical attacks (+6%)

ORACLE

Target of Data Breaches

900 Million records

verizon business

2010 Data Breach Investigations Report

Type	Category	% Breaches	% Records
Database Server	Servers & Applications	25%	92%
Desktop Computer	End-User Devices	21%	1%

Web app/server	Servers & Applications	19%	13%
Payment card	Offline Data	18%	<1%
POS server (store controller)	Servers & Applications	11%	<1%
Laptop computer	End-User Devices	7%	<1%
Documents	Offline Data	7%	<1%
POS terminal	End-User Devices	6%	<1%
File server	Servers & Applications	4%	81%
Automated Teller Machine (ATM)	End-User Devices	4%	<1%
FTP server	Servers & Applications	2%	3%
Mail server	Servers & Applications	2%	4%

ORACLE

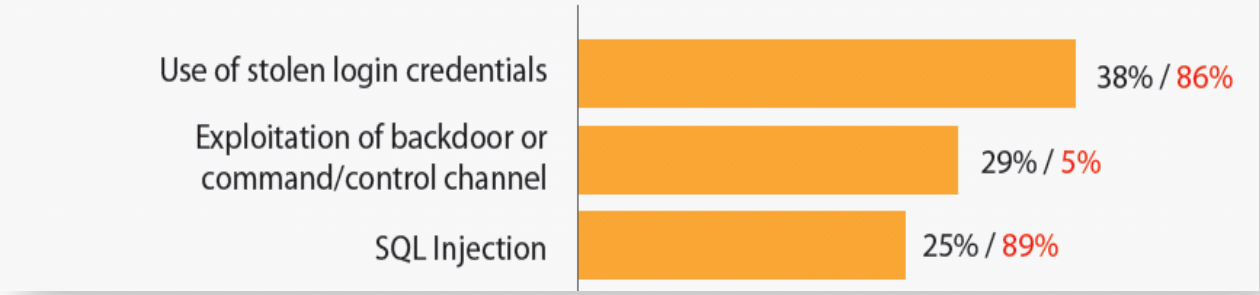
verizon business

2010 Data Breach Investigations Report

"The versatility and effectiveness of SQL injection make it a multi-tool of choice among cyber criminals."

- SQL injection attacks against databases are responsible for 89% of breached data
- SQL injection vulnerabilities are endemic
- Require code changes to be made to an application.

Figure 21. Types of hacking by percent of breaches within Hacking and percent of records



ORACLE

“Often, preventing SQL injection attacks is based more on hope than systematic protection.”

IOUG independent oracle users group
For the Complete Technology & Database Professional

2011 IOUG DATA SECURITY SURVEY

Only 48%
not aware of all databases with sensitive data

66%
not sure if web applications subject to SQL injection

37%
do not use, or are unsure of using, a network-based database firewall solution

ORACLE

No. 1 Software flaw

- CWE-89 : Improper Neutralization of Special Elements used in an SQL Command

  http://cwe.mitre.org/top25/#CWE-89

- Using a badly designed application to extract content from a database

Why is it number 1?

Because Applications

- Trust users
- Tend to be not designed defensively
- Are given high levels of privilege
- Each application is unique

(Ab)users subvert the application to access to the database

When a web app goes bad

- The application code

  ProdQuery= " SELECT ProdName, ProdDescription FROM Stock WHERE
  ProdNumber = " & ProdRequest.QueryString("ProdID") “

- The URL

  http://www.acme.com/products/products.asp?ProdID=1234

- The executed query

  SELECT ProdName, ProdDescription FROM Stock WHERE ProdNumber = 1234

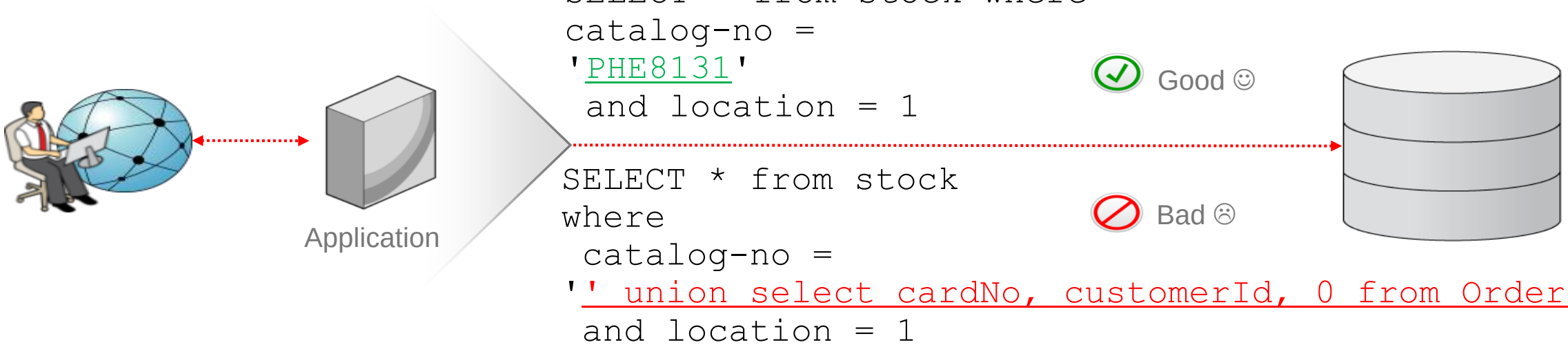
- The attack

  http://www.acme.com/products/products.asp?ProdID=1234 or 1=1

- The new query

  SELECT ProdName, ProdDescription FROM Stock WHERE ProdNumber = 1234 OR 1=1

A worked example



How we expect the application to interact with the database

Parameters for SQL come from user input (e.g. a web browser).

The application layer accepts the values for `catalog-no` and `location` (‘**PHE8131**’, ‘**1**’) and pastes them into the pre-canned query template.

```
SELECT * from stock where catalog-no = '________' and location =
```

Output:

Description	*Price*	*# in Stock*
Star Trek - The Next Generation Season 2	39.35	15
Star Trek - The Next Generation Season 3	39.35	12
Star Trek - The Next Generation Season 4	39.35	13
Star Trek - The Next Generation Season 5	39.35	17

Assembling an *abnormal* SQL statement

Instead of inputting a normal value for `catalog-no`, the user enters

```
' union select cardNo,  customerId, 0 from Orders --
```

And the database receives the following query

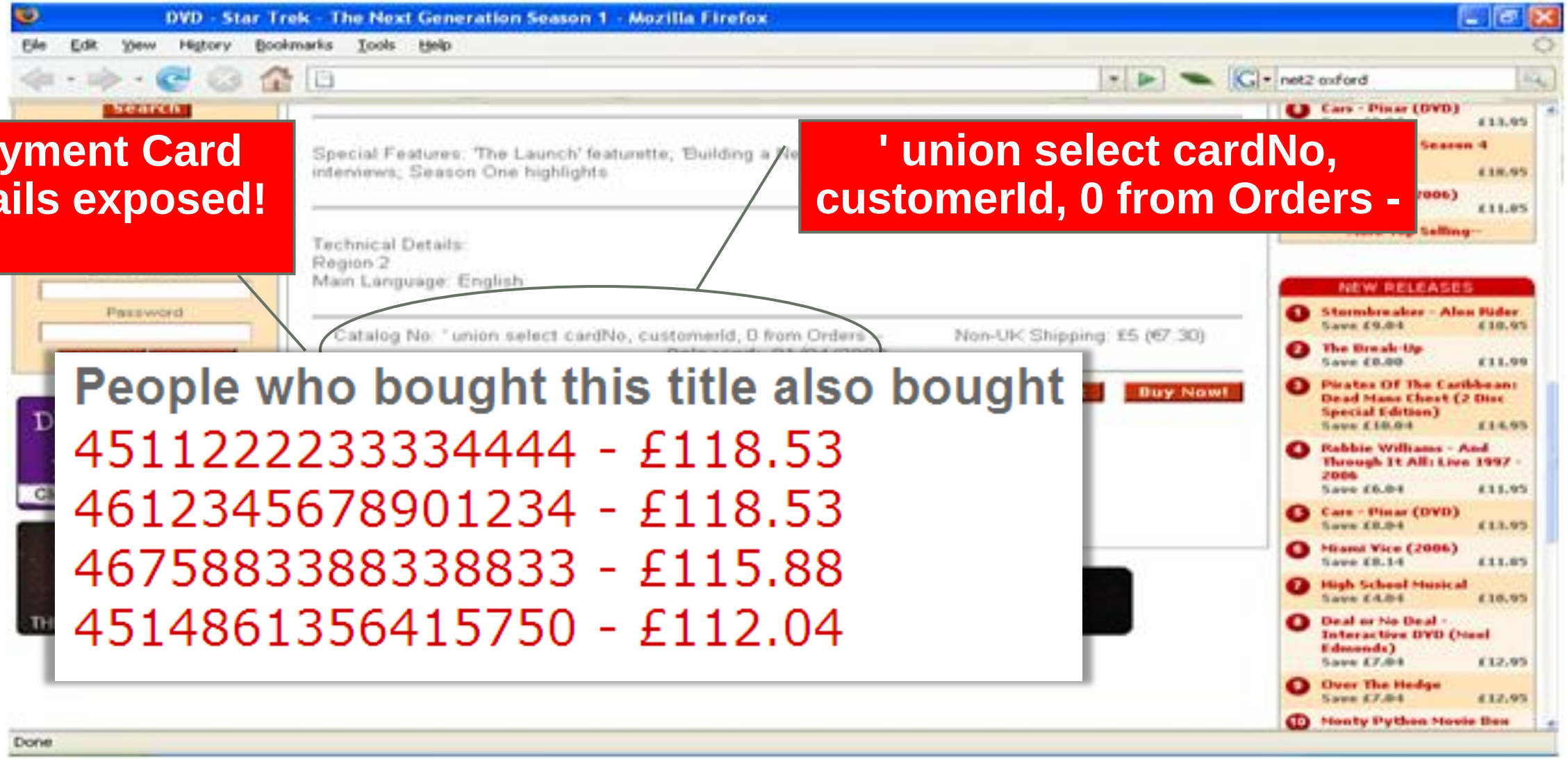
```
SELECT * from stock where catalog-no = '_______'  and location =
                        _____________________________' and location =
```

Output:

Payment Card details

Description	*Price*	*# in Stock*
4511222233334444	11853	0
4612345678901234	11853	0
4675883388338833	11588	0
4514861356415750	11204	0

What does the attacker actually see?



Payment Card details exposed!

' union select cardNo, customerId, 0 from Orders -

People who bought this title also bought

4511222233334444 - £118.53

4612345678901234 - £118.53

4675883388338833 - £115.88

4514861356415750 - £112.04

More SQL Injections

```
SELECT * from stock where catalog-no = 'PHE8131' and location = 1
```

```
SELECT * from stock where catalog-no = ''--' and location = 1
```

```
SELECT * from stock where catalog-no = '' having 1=1 -- ' and location = 1
```

```
SELECT * from stock where catalog-no = '' order by 4--' and location = 1
```

```
SELECT * from stock where catalog-no = '' union select cardNo,customerId,0
from Orders where name = 'John Smith'--' and location = 1
```

```
SELECT * from stock where catalog-no = '' union select min(cardNo),1,0 from
Orders where cardNo > '0'--' and location = 1
```

```
SELECT * from stock where catalog-no = 'PHE8131' and location = 1; drop
table stock
```

Challenges of SQL Injections

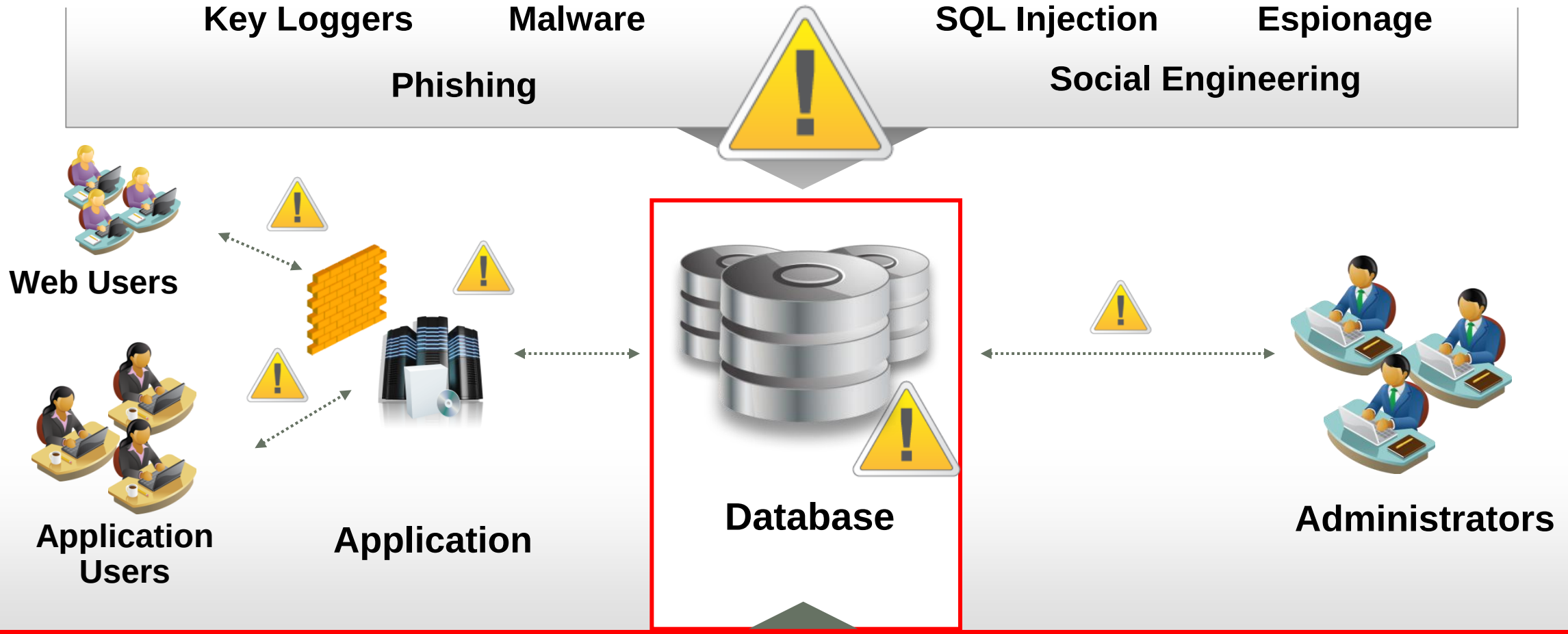
- How Do You Catch Them?
  - Must detect anomalous SQL
  - Need to ensure detection is accurate (i.e. do not have false positive nor false negative errors)
  - Must be performant (i.e. can be done in real-time)

- What Kind of Detection Engine Do I Need?
  - SQL Grammar versus Regular Expressions
  - White Lists versus Black Lists

Agenda

- Defining the problem?
- Do we need another firewall?
- The Oracle Database Firewall

Existing Security Solutions Not Enough



Data Must Be Protected in depth

ORACLE

What is a firewall?

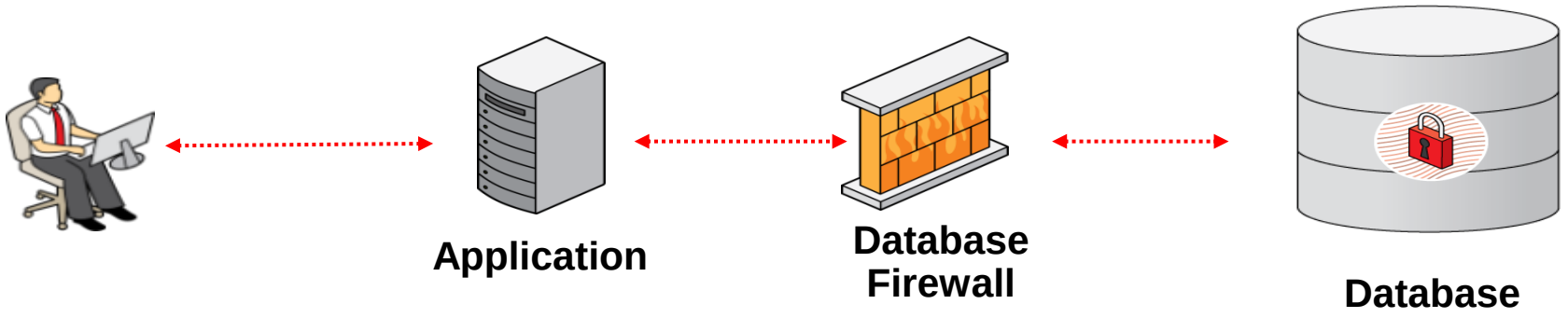
Webster dictionary

- A wall or partition designed to inhibit or prevent the spread of fire.
- Any barrier that is intended to thwart the spread of a destructive agent

Wikipedia

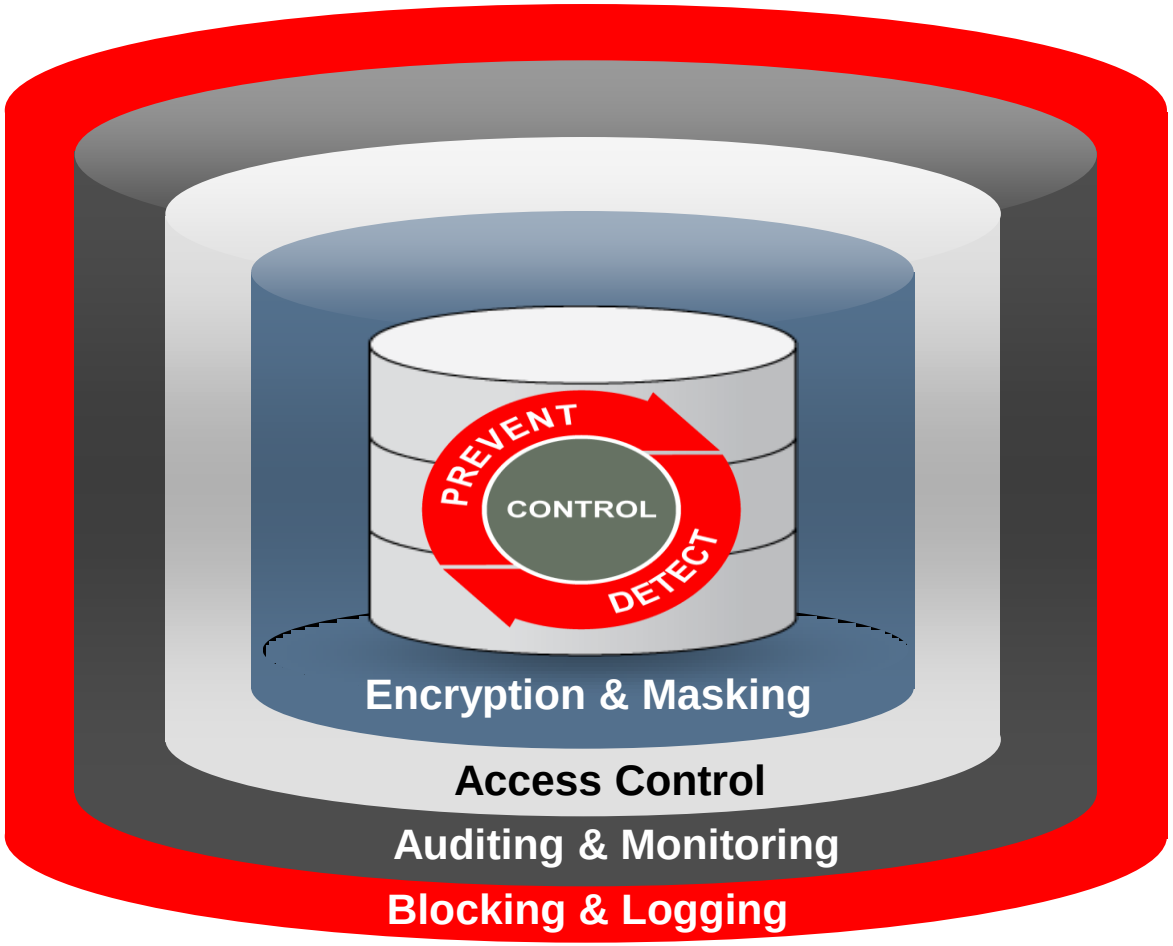
- A device or set of devices designed to permit or deny network transmissions based upon a set of rules and is frequently used to protect networks from unauthorised access while permitting legitimate communications to pass

Why not have a Database Firewall?



First line of defence to monitor and protect against existing and emerging threats

Database Defense-in-Depth



Encryption and Masking

- Oracle Advanced Security
- Oracle Secure Backup
- Oracle Data Masking

Access Control

- Oracle Database Vault
- Oracle Label Security

Auditing and Monitoring

- Oracle Audit Vault
- Oracle Configuration Management
- Oracle Total Recall

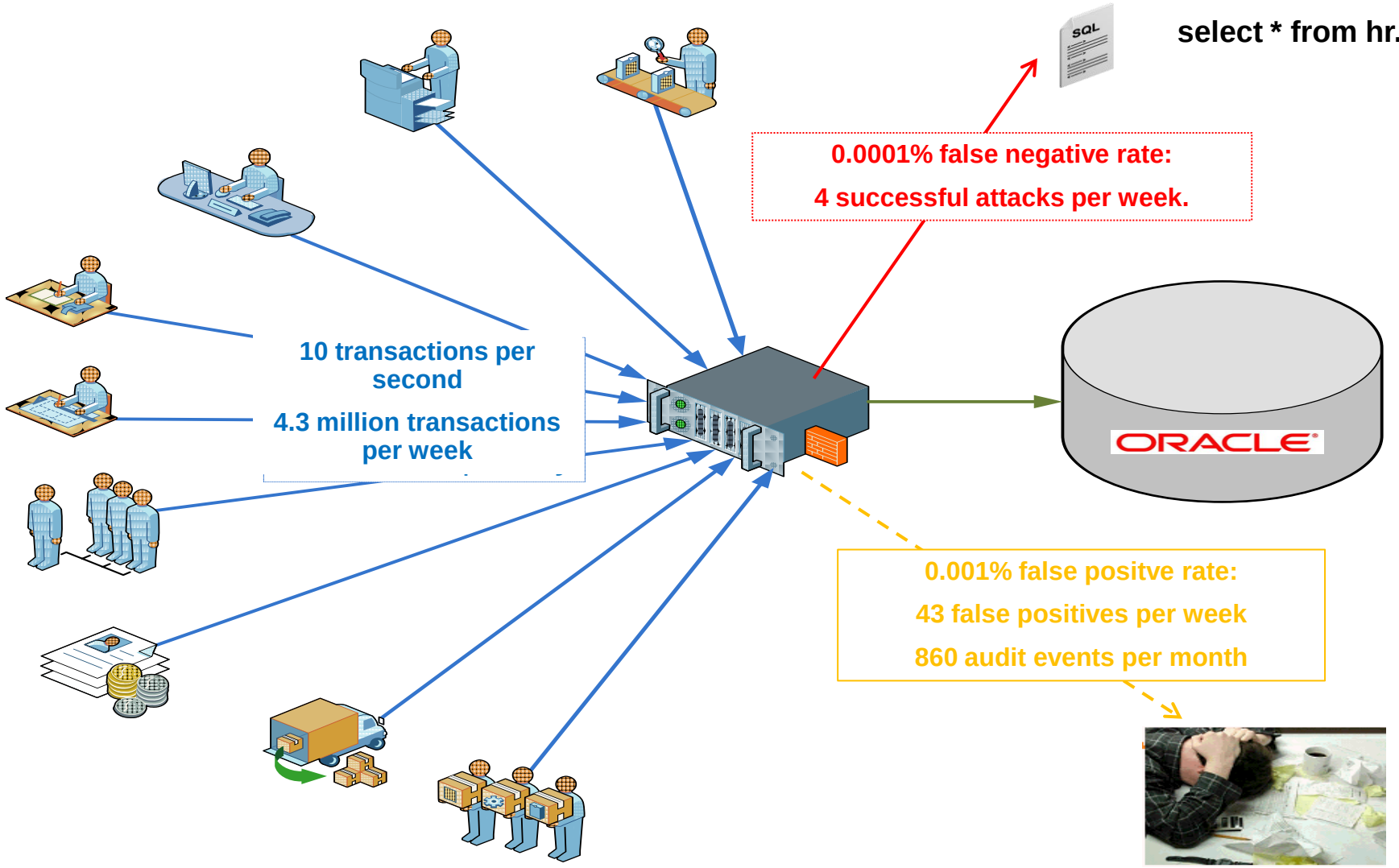
Blocking and Logging

- Oracle Database Firewall

Challenges facing a Database Firewall

- Building accurate policy profiles of good application behavior with changes over time
- Performance against your application as the number of transactions increase over the network
- Needing to throw more hardware on the network to handle your workload for scalability

The Cost of Inaccuracy



Detection Engines
Regular Expressions Based Approach

- Regex
  - A pattern describing a set of strings

- With regular expressions you can:
  - Find out whether a certain pattern occurs in the text
  - Locate strings matching a pattern and remove them or replace them with something else
  - Extract the strings matching the pattern

[Source: Andrei Zmievski, Yahoo! Inc.]

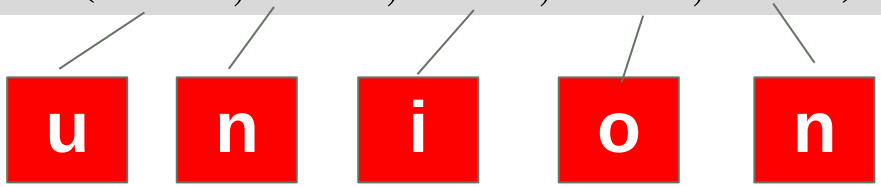
String Matching Keywords is Inadequate

- union is NOT universally bad when next to this select

```
SELECT lastname from boys
union
SELECT lastname from girls
```

- union without “saying it”

```
uni/* */on
u/* */nion
char(117,110,105,111,110)
```



Can you *'Tune'* Regular Expressions?

- union is bad when it appears ***near*** select

u(?:nion\b.{1,100}?\bselect

```
"(?:\b(?:(?:s(?:elect\b(?:.{1,100}?\b(?:(?:length|count|top)\b.{1,100}?\bfrom|from\b.{1,100}?\bwhere)|.*
?\b(?:d(?:ump\b.*\bfrom|ata_type)|(?:to_(?:numbe|cha)|inst)r))|p_(?:(?:addextendedpro|sqlexe)c|(?:oacrea
t|prepar)e|execute(?:sql)?|makewebtask)|ql_(?:longvarchar|variant))|xp_(?:reg(?:re(?:movemultistring|ad)
|delete(?:value|key)|enum(?:value|key)s|addmultistring|write)|e(?:xecresultset|numdsn)|(?:terminat|dirtr
e)e|availablemedia|loginconfig|cmdshell|filelist|makecab|ntsec)|u(?:nion\b.{1,100}?\bselect|tl_(?:file|h
ttp))|group\b.*\bby\b.{1,100}?\bhaving|d(?:elete\b\W*?\bfrom|bms_java)|load\b\W*?\bdata\b.*\binfile|(?:n
?varcha|tbcreato)r)\b|i(?:n(?:to\b\W*?\b(?:dump|out)file|sert\b\W*?\binto|ner\b\W*?\bjoin)\b|(?:f(?:\b\W
*?\(\W*?\bbenchmark|null\b)|snull\b)\W*?\()|a(?:nd\b ?(?:\d{1,10}|[\'\"][^=]{1,10}[\'\"])
?[=<>]+|utonomous_transaction\b)|o(?:r\b ?(?:\d{1,10}|[\'\"][^=]{1,10}[\'\"])
?[=<>]+|pen(?:rowset|query)\b)|having\b ?(?:\d{1,10}|[\'\"][^=]{1,10}[\'\"])
?[=<>]+|print\b\W*?\@\@|cast\b\W*?\()|(?:;\W*?\b(?:shutdown|drop)|\@\@version)\b|'(?:s(?:qloledb|a)|msda
sql|dbo)')"
```

[Source: ModSecurity, Web Application Firewall]

Issues with Regular Expressions

- Fails to understand meaning, motives and intentions of SQL when you just use strings and text
- Good Statement

```
SELECT * from stock where catalog-no = 'PHE8131' and
location = 1
```

- Bad Statement – SQL injection

```
SELECT * from stock where catalog-no = '' union
select cardNo, customerId, 0 from Orders --' and
location = 1
```

Understanding SQL

- SQL is a language with about 400 key words and a strict grammar structure

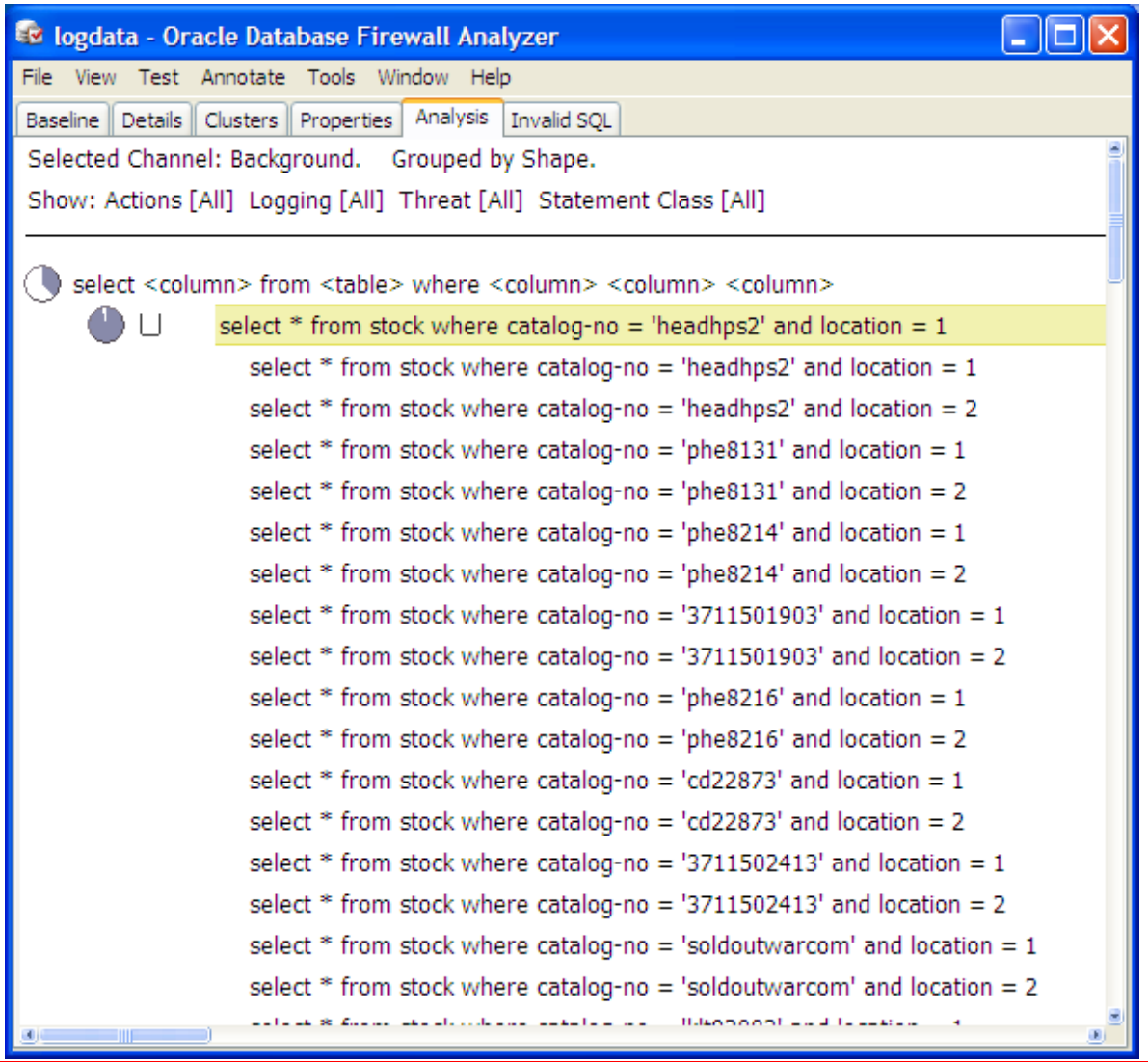
```
UPDATE tbl_users SET comments = 'The user has asked for
another account no, and wishes to be billed for services
between 1/2/2009 and 2/2/2009, and wants to know where the
invoice should be sent to. She will select the new service
level agreement to run from 3/7/2009 next month' WHERE id =
                        'A15431029';
```

KEY WORDS	SCHEMA	DATA	OPERATORS

- Grammatical context necessary to prevent false categorisation

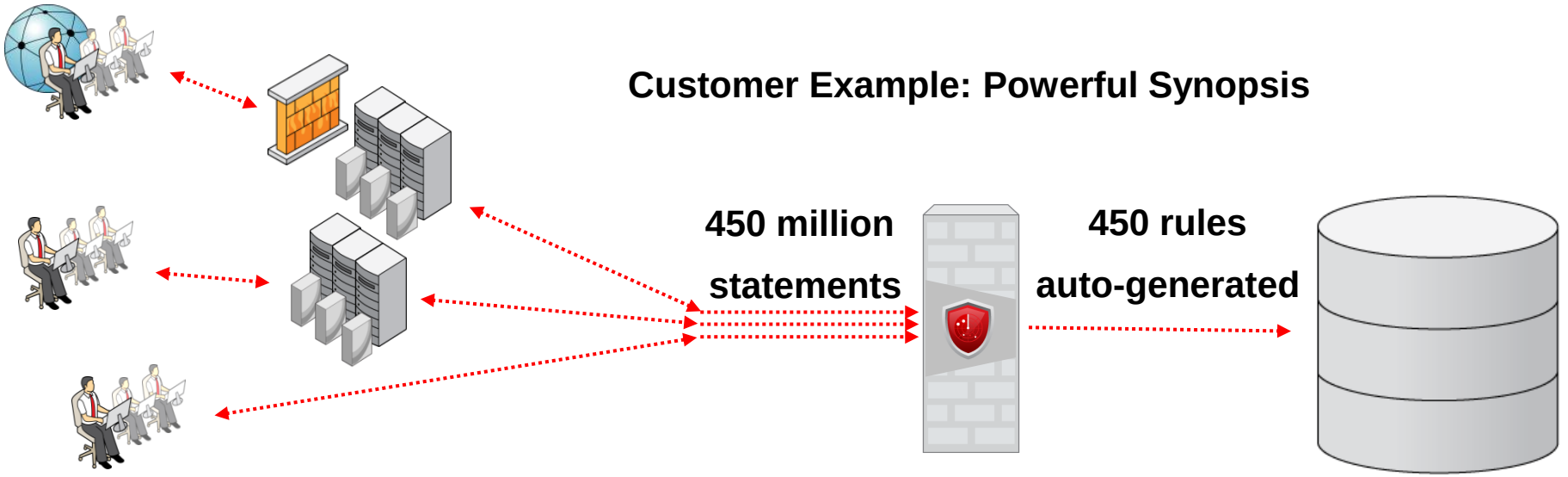
Grammar-based Clustering

- Thousands of individual statements transformed into manageable clusters
- Anomalous statements produce a new cluster and are instantly detected (e.g. blocked)



Grammar-Based Clustering

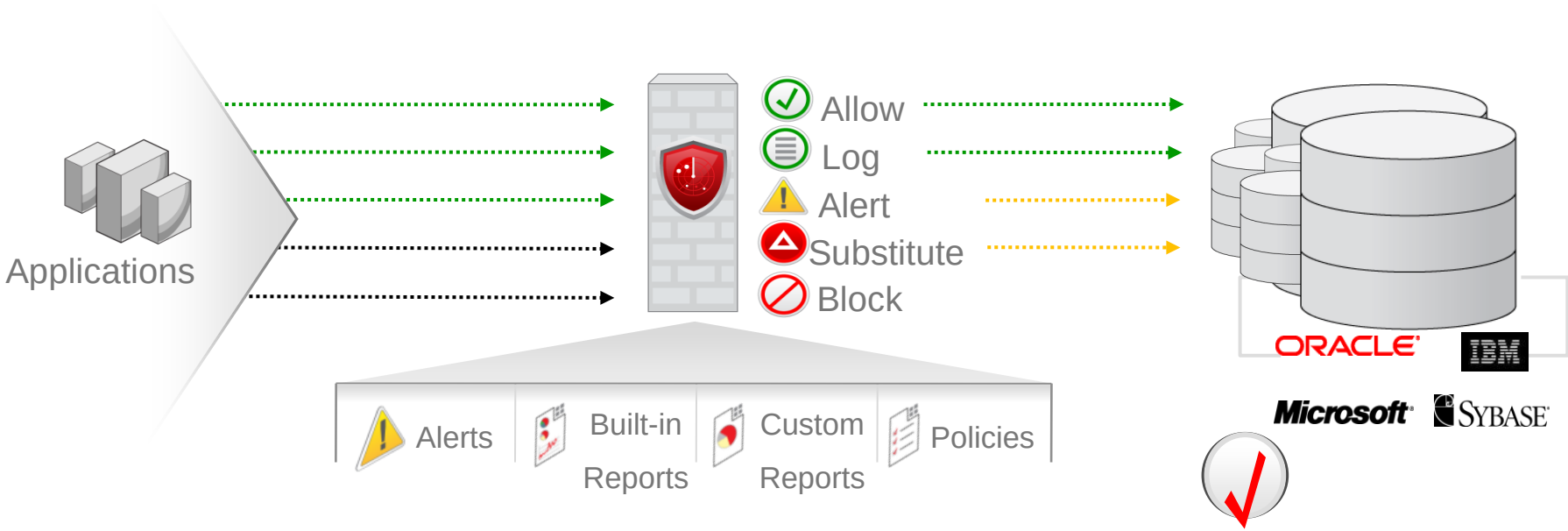
- Accuracy and Understanding
  - Use the same level of language power to detect SQL as to process it
- Performance
  - Speed of lookup is constant in the number of SQL rules in the policy
- Security
  - SQL injection and other dangerous SQL detected as anomalies



Agenda

- Defining the problem?
- Do we need another firewall?
- The Oracle Database Firewall

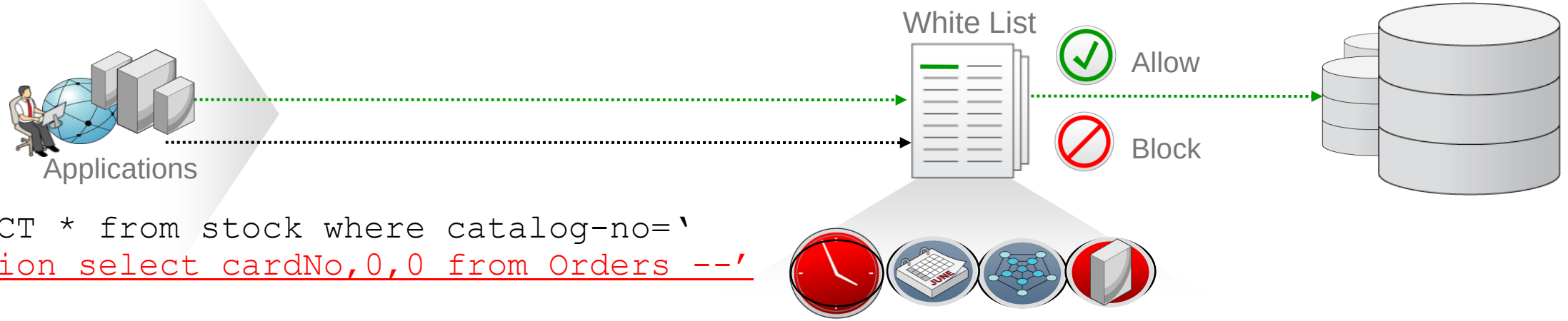
First line of defense



- Monitor and control database activity on the network
- Highly accurate SQL grammar based analysis to enforce normal activity
- Capture and log database interactions for forensic analysis and compliance reporting

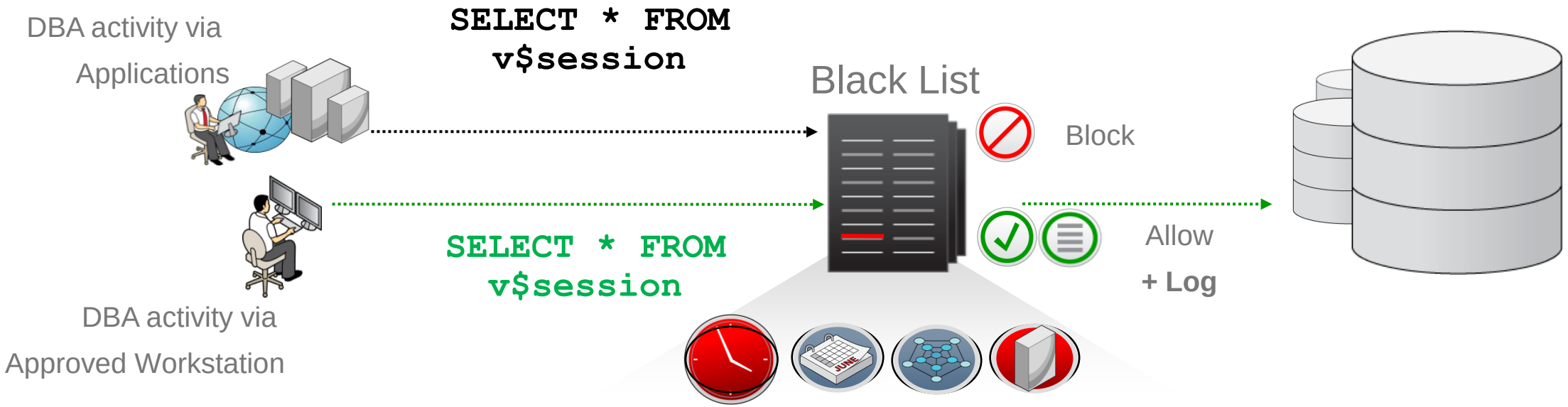
Positive Security Model

```
SELECT * from stock where catalog-no='PHE8131'
```



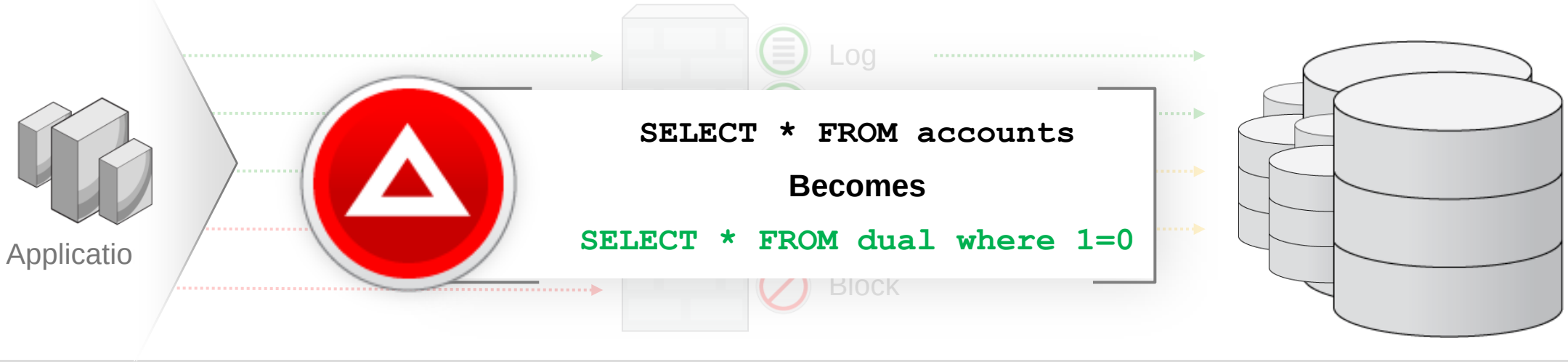
- "Allowed" behavior can be defined for any user or application
- Automated whitelist generation for any application
- Many factors to define policy (e.g. network, application, etc)
- Out-of-policy Database network interactions instantly blocked

Negative Security Model



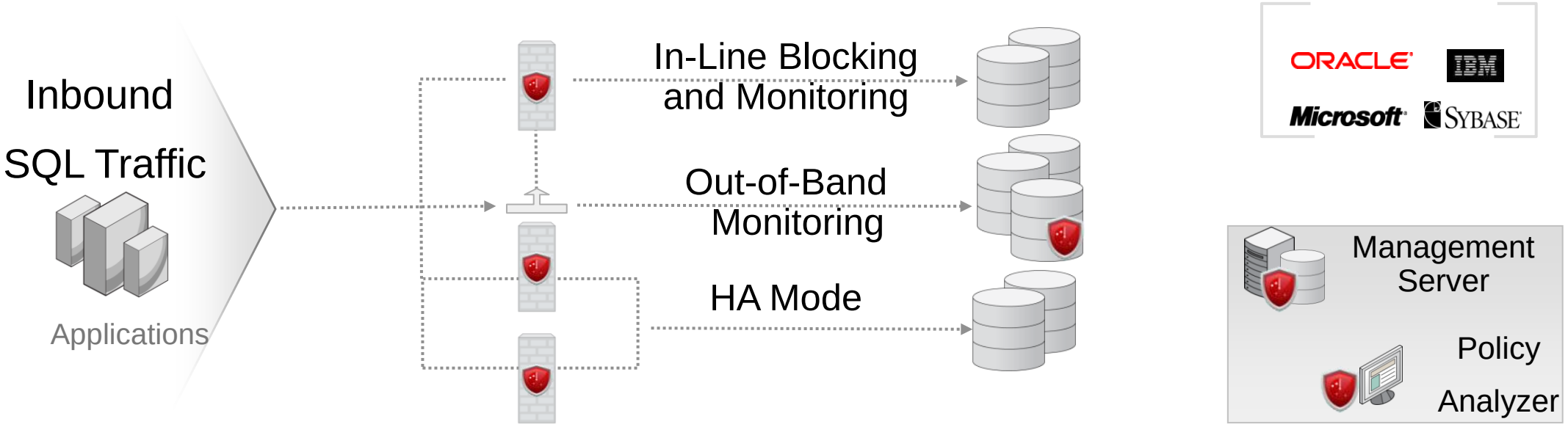
- Stop specific unwanted SQL interactions, user or schema access
- Ensures database interactions originate from appropriate sources
- Blacklist can take into account built-in factors such as time of day, day of week, network, application, etc
- Provide flexibility to authorized DBAs to do their job whilst monitoring usage

But what happens next?



- Application – Database connection pools cannot be disconnected
- Unique graceful blocking achieved by substituting out-of-policy statement with predefined benign statement
- Database guaranteed protected from inappropriate statement whilst the network session remains open.

Deployment Architecture



- Application and database vendor agnostic
- Deployment modes:
  - Inline, Out-of-Band, Optional Host Based Agents
  - Policy enforcement separated from policy management and reporting
  - High Availability
- Software solution based on hardened OE Linux and Intel for flexibility and scalability

Flexible Deployment Model

- Deploy on existing or new hardware:
  - Runs Oracle Enterprise Linux base operating system
  - Firewall blocking mode requires certified NIC card
- Scales vertically
  - Add CPU, disk, and memory to the servers versus adding more and more appliances
- Co-residence of Database Firewall and Database Firewall Management Server

User Role Reporting

- Entitlement Reports
  - User names
  - User roles and privileges
  - Last changed, changed by whom and when
- Automated and transparent
  - User role reporting can be run ad-hoc or scheduled
  - Report on user roles and privileges
  - Deltas since the last report

Stored Procedure Reporting

- Stored procedure contents

  Its not enough to know a procedure was run, it is important to know what SQL was executed when the procedure is called.

- Stored procedure reports

  Name

  Content

  Threat rating (injection risk, system tables etc).

  Stored procedure type (DML, DDL, DCL, SELECT etc)

  Last changed, changed by whom and when

- Automated and transparent

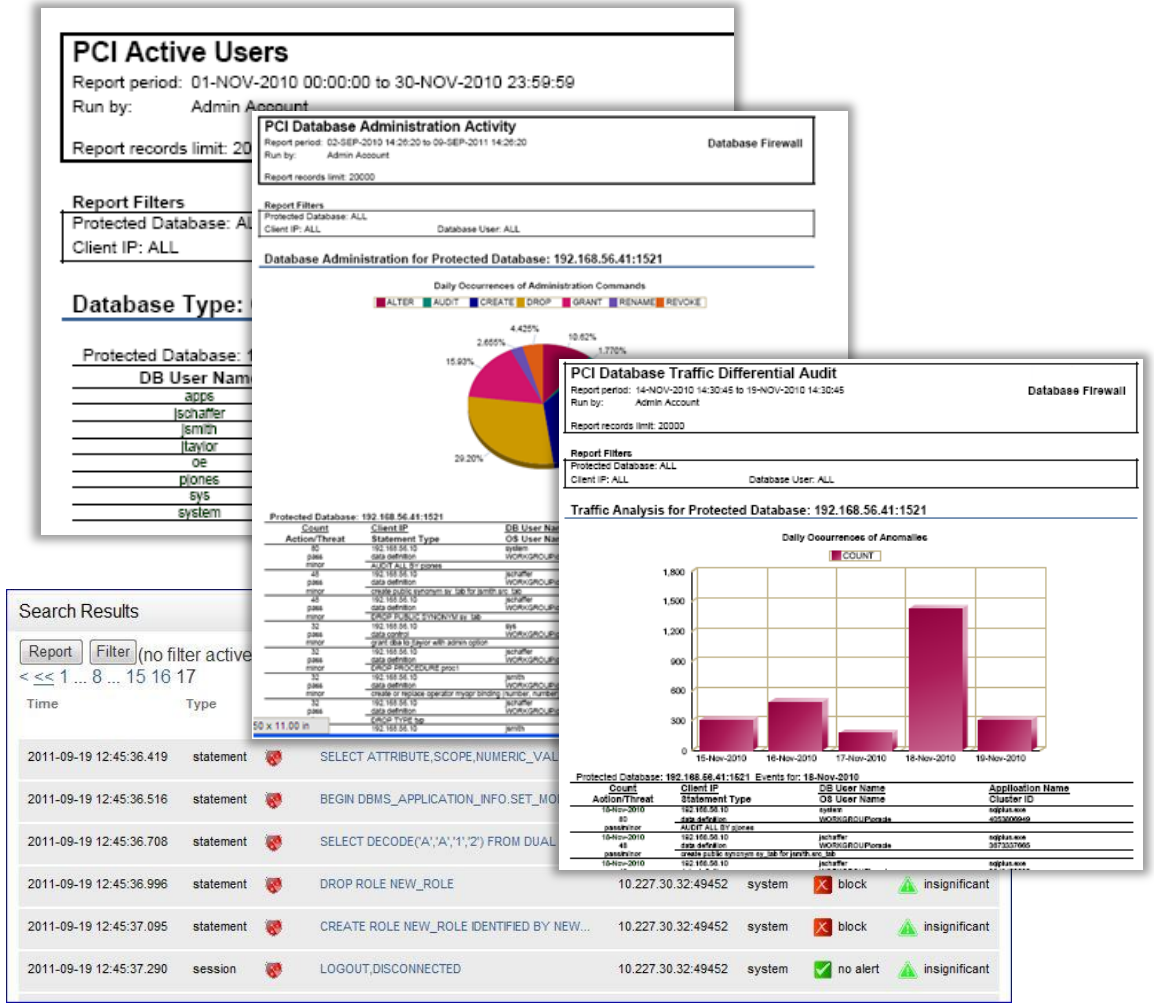
  Stored procedure reporting can be run adhoc or scheduled

Database Monitoring and Reporting

Extensive built in Security and Compliance reports

- Can be modified and customized
- Database activity monitoring
- Blocked and alerted statements
- Supports demonstrating controls for PCI, SOX, HIPAA, etc.

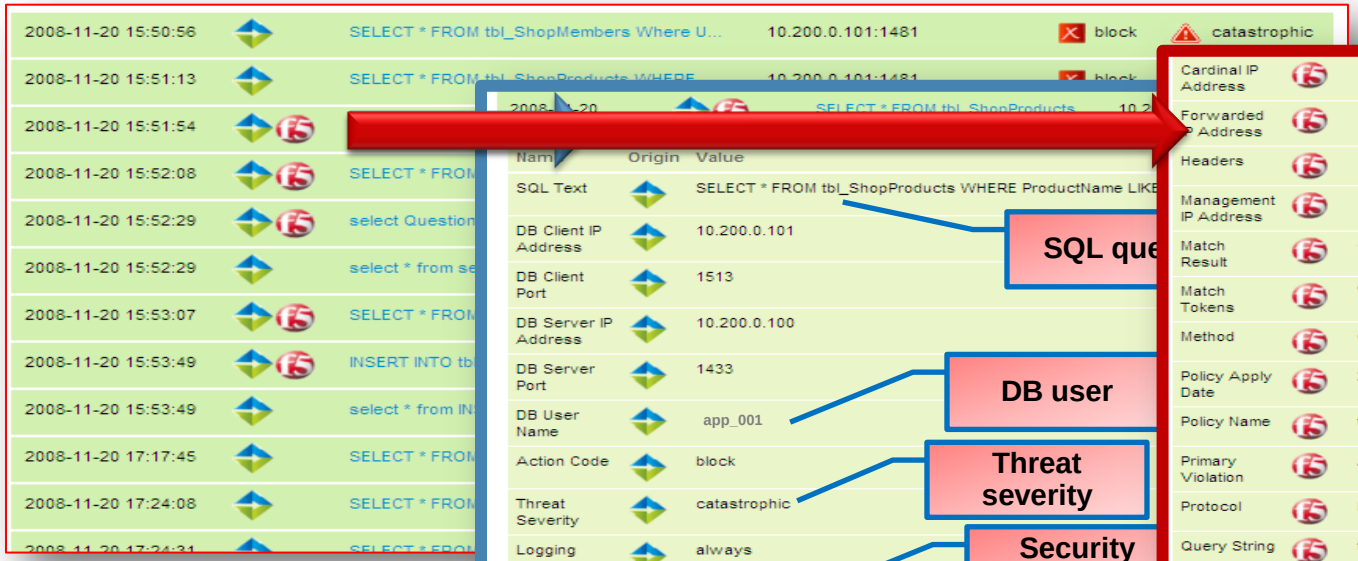
Logged SQL statements can be sanitized of sensitive PII data



F5 Application Security Manager

- World-class best-of-breed Web Application Firewall
- Provides comprehensive protection for all web application vulnerabilities including OWASP top 10
- Provides out of the box security
- Logs and reports all application traffic
- Provides L2->L7 protection
- Positive and negative security models

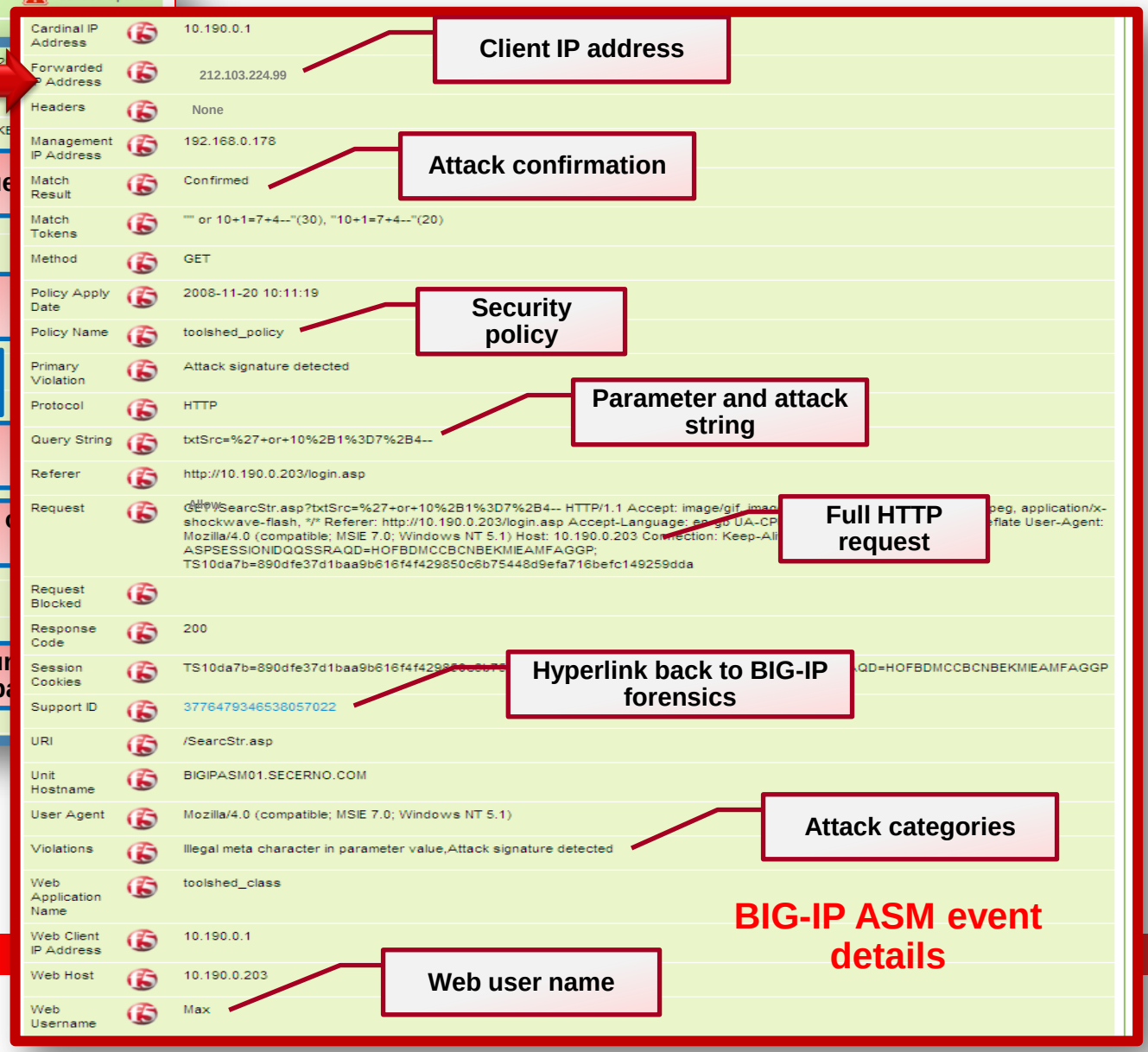
Oracle Database Firewall and F5 ASM reporting



Oracle Database Firewall event summary

showing database and Web events

Oracle Database Firewall event details



Client IP address

Attack confirmation

DB user

Security policy

Threat severity

Parameter and attack string

Security policy

Category of attack

Full HTTP request

Hyperlink back to BIG-IP forensics

Attack categories

BIG-IP ASM event details

Web user name

Summary

- Statistically your data is at risk due to poorly written applications
- Apply your defense in-depth strategy to databases
- Oracle DB Firewall provides :
  - Accuracy, scalability, deployment flexibility, heterogeneous support

For More Information

or

oracle.com/database/security

or

james.sadler@oracle.com