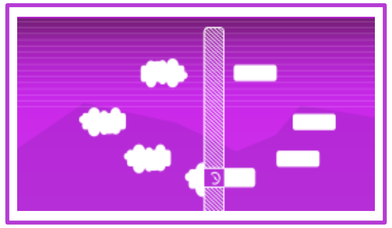
Building Security Analytics using Native XML Databases

Mansi Sheth

Email: msheth@veracode.com
Twitter: @1MansiS

Binary Static Analysis (SAST)



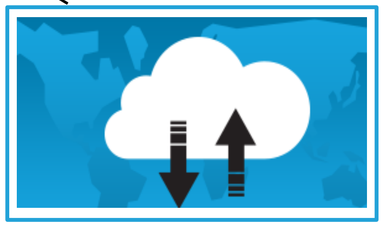
Vendor Application Security

Web Apps Perimeter Monitoring

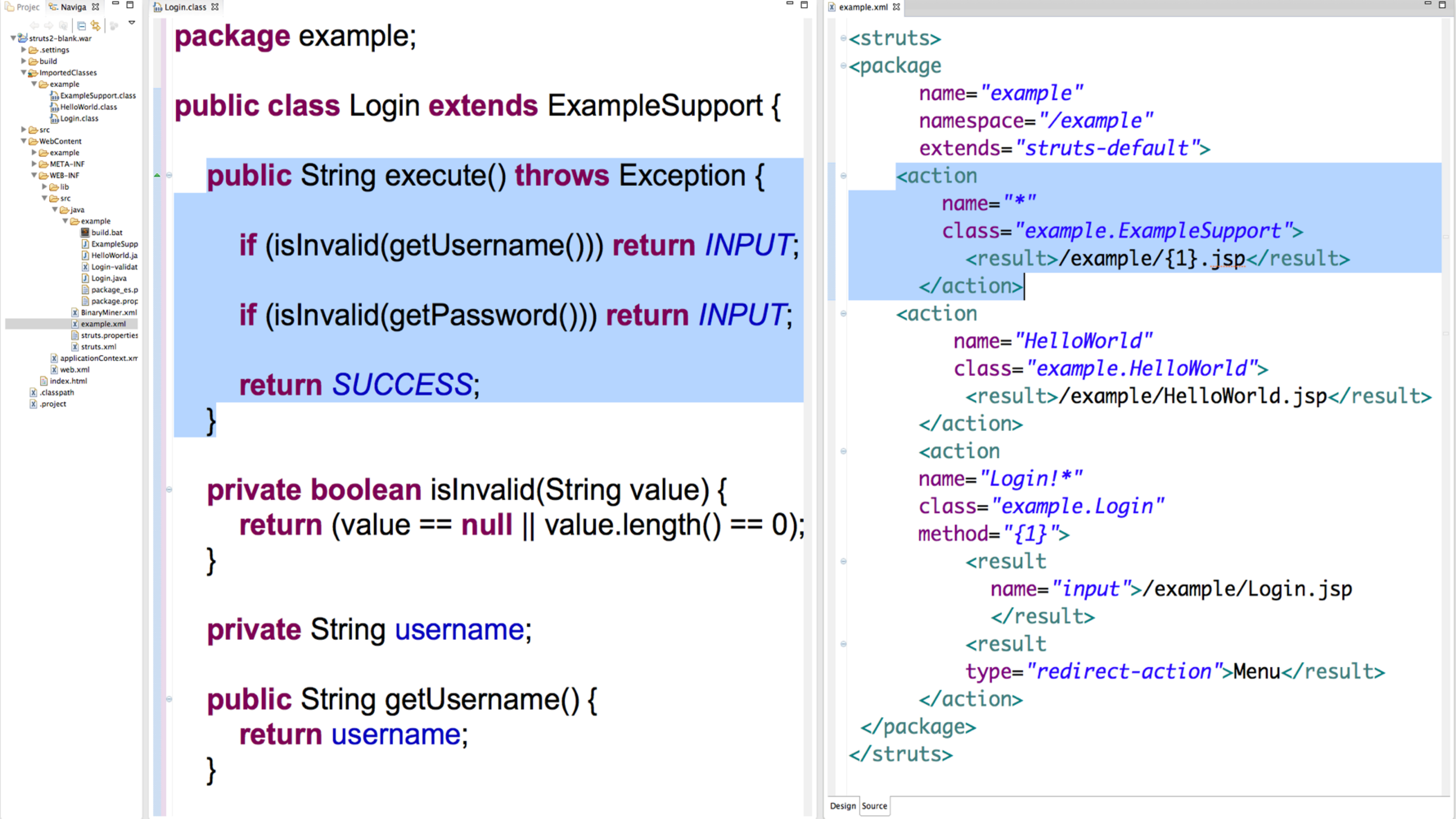
Big Data

Dynamic Analysis (DAST)

Mobile Application Security



Single Cloud-Based Platform


Project
Navigator
struts2-blank.war
.settings
build
ImportedClasses
example
ExampleSupport.class
HelloWorld.class
Login.class
src
WebContent
example
META-INF
WEB-INF
lib
src
java
example
build.bat
ExampleSupp
HelloWorld.ja
Login-validat
Login.java
package_es.p
package.prop
BinaryMiner.xml
example.xml
struts.properties
struts.xml
applicationContext.xm
web.xml
index.html
.classpath
.project
Login.class
package example;
public class Login extends ExampleSupport {
public String execute() throws Exception {
if (isInvalid(getUsername())) return INPUT;
if (isInvalid(getPassword())) return INPUT;
return SUCCESS;
}
private boolean isInvalid(String value) {
return (value == null || value.length() == 0);
}
private String username;
public String getUsername() {
return username;
}
example.xml
<struts>
<package
name="example"
namespace="/example"
extends="struts-default">
<action
name="*"
class="example.ExampleSupport">
<result>/example/{1}.jsp</result>
</action>
<action
name="HelloWorld"
class="example.HelloWorld">
<result>/example/HelloWorld.jsp</result>
</action>
<action
name="Login!*"
class="example.Login"
method="{1}">
<result
name="input">/example/Login.jsp
</result>
<result
type="redirect-action">Menu</result>
</action>
</package>
</struts>
Design
Source

#XML Metadata

Login.class

```
package example;

public class Login extends ExampleSupport {

    public String execute() throws Exception {

        if (isInvalid(getUsername())) return INPUT;

        if (isInvalid(getPassword())) return INPUT;

        return SUCCESS;
    }

    private boolean isInvalid(String value) {
        return (value == null || value.length() == 0);
    }
```

```
<class name="example.Login">
    <inheritance
        name="example.ExampleSupport"/>
    <function name="isInvalid"/>
    <function name="&lt;init&gt;">
        <apiCalls
            name="example.ExampleSupport:&lt;init&gt;"/>
    </function>
    <function name="execute">
        <apiCalls
            name="example.Login:getUsername"/>
        <apiCalls
            name="example.Login:isInvalid"/>
        <apiCalls
            name="example.Login:getPassword"/>
        <apiCalls
            name="example.Login:isInvalid"/>
    </function>
</class>
```

#XML Metadata

```
<struts>
  <package
      name="example"
      namespace="/example"
      extends="struts-default">
        <action
          name="*"
          class="example.ExampleSupport">
            <result>/example/{1}.jsp</result>
        </action>
        <action
            name="HelloWorld"
            class="example.HelloWorld">
             <result>/example/HelloWorld.jsp</
        </action>
```

```
<class name="WEB-INF/example.xml">
    <function name="action/@name">
        <apiCalls
            name="HelloWorld"/>
        <apiCalls
            name="Login!*"/>
        <apiCalls
            name="*"/>
    </function>
</class>
```

#XML Metadata

```
<Archives>
    <archive name="struts2-blank-2.0.1.war">
        <class name="example.Login">
            <inheritance name="example.ExampleSupport"/>
            <function name="isInvalid"/>
            <function name="&lt;init&gt;">
                <apiCalls name="example.ExampleSupport:&lt;init&gt;"/>
            </function>
            <function name="execute">
                <apiCalls name="example.Login:getUsername"/>
                <apiCalls name="example.Login:isInvalid"/>
                <apiCalls name="example.Login:getPassword"/>
                <apiCalls name="example.Login:isInvalid"/>
            </function>
        </class>
        <class name="WEB-INF/src/java.example.xml">
            <function name="action/@name">
                <apiCalls name="HelloWorld"/>
                <apiCalls name="Login!*"/>
                <apiCalls name="*"/>
            </function>
        </class>
        <class name="WEB-INF.web.xml">
            <function name="filter-class">
                <apiCalls name="org.apache.struts2.dispatcher.FilterDispatcher"/>
            </function>
        </class>
    </archive>
</Archives>
```

10MB * 300 ~= 3GB

3GB * 5 * 52 ~= 1TB

1TB * 7 ~= 7TB

Clients using insecure api: org.insecured:f_insecure()

Top 10 most prevalent frameworks across all our clients ?

Most inherited entry points from framework f ?

Approaches Considered

Bash Automated Script

Relational Database

Hadoop + Mahout

Hadoop + AVRO + HIVE

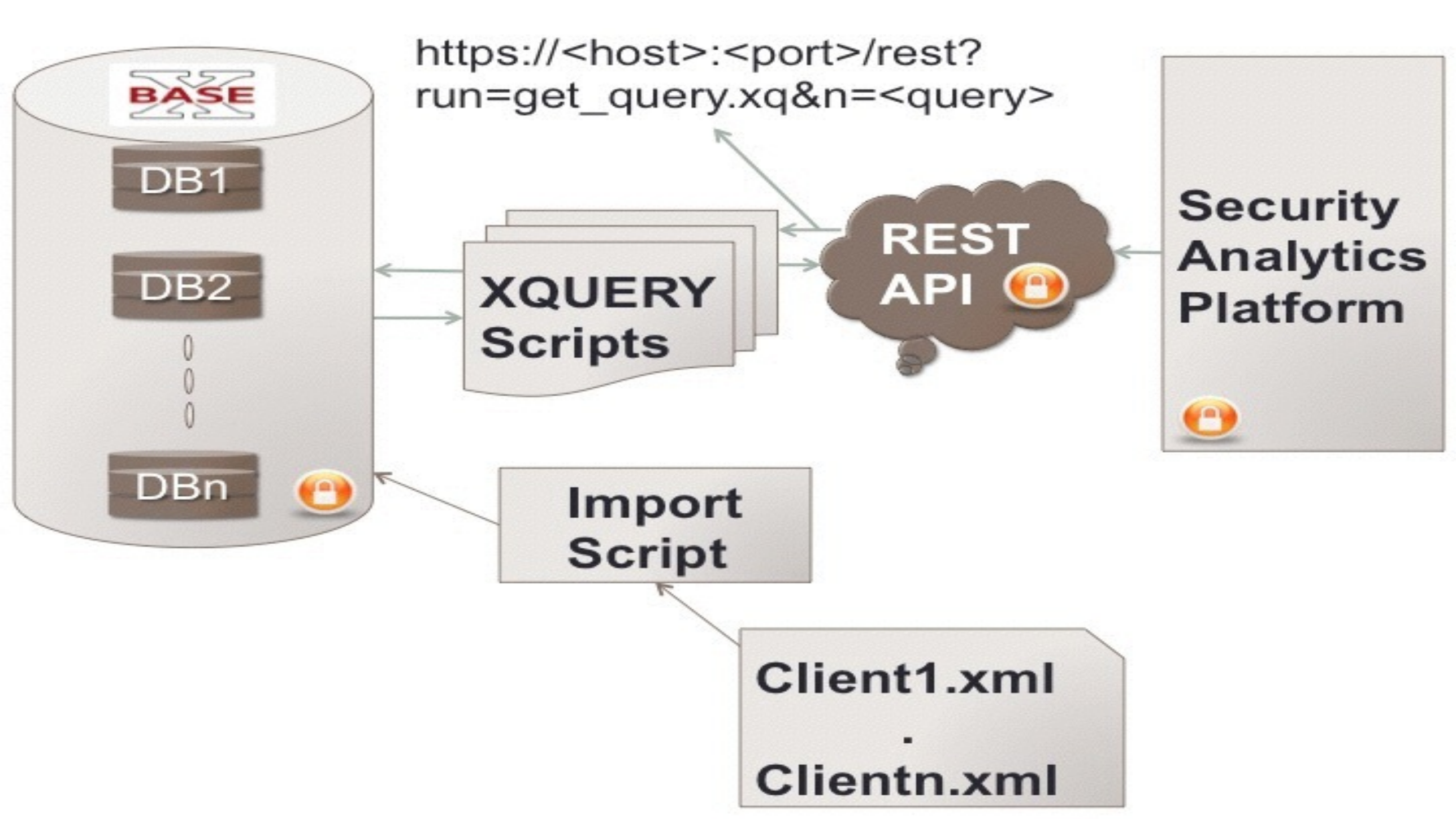
NoSQL

Wish List

Store Parsed XML

Use XML Query technologies

Results in reasonable timeframe

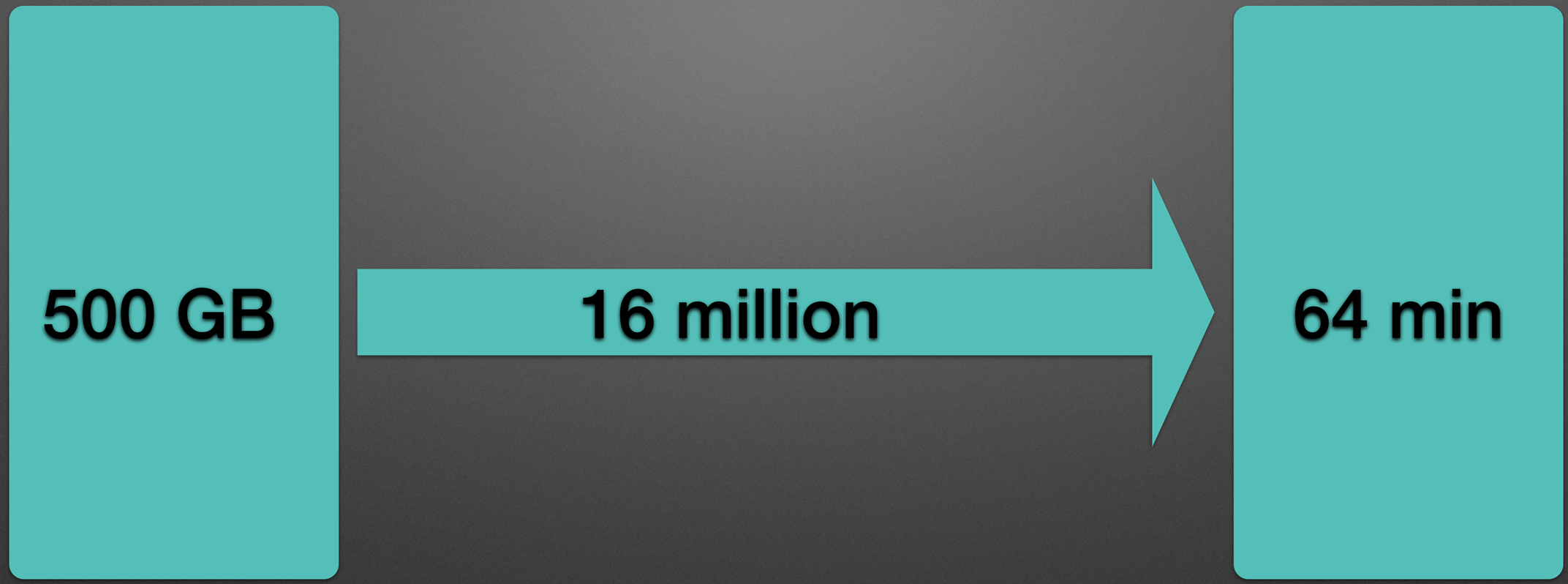

https://<host>:<port>/rest?
run=get_query.xq&n=<query>
BASE
DB1
DB2
DBn
XQUERY
Scripts
REST
API
Security
Analytics
Platform
Import
Script
Client1.xml
.
Clientn.xml

```
for $db in db:list()
    let $query :=
    "declare variable $db external; "
        ||  "db:open($db)"
        || $n
    return
        xquery:eval($query,
            map { 'db': $db, 'query': $n
            })
```

Exploting Struts 2 OGNL Vulnerability CVE-2013-2251

Languages

- English
- Espanol


500 GB
16 million
64 min

BaseX Wish List

Security at rest

Credits

Chris Eng
Veracode Management
BaseX Community
XMLPrague Committee
Photo Credits: Kenneth Rougeau

Building Security Analytics using Native XML Databases

Mansi Sheth

Email: msheth@veracode.com
Twitter: @1MansiS
Blog: http://www.veracode.com/blog
Github: https://github.com/1MansiS/BuildingSecurityAnalyticsUsingNXD