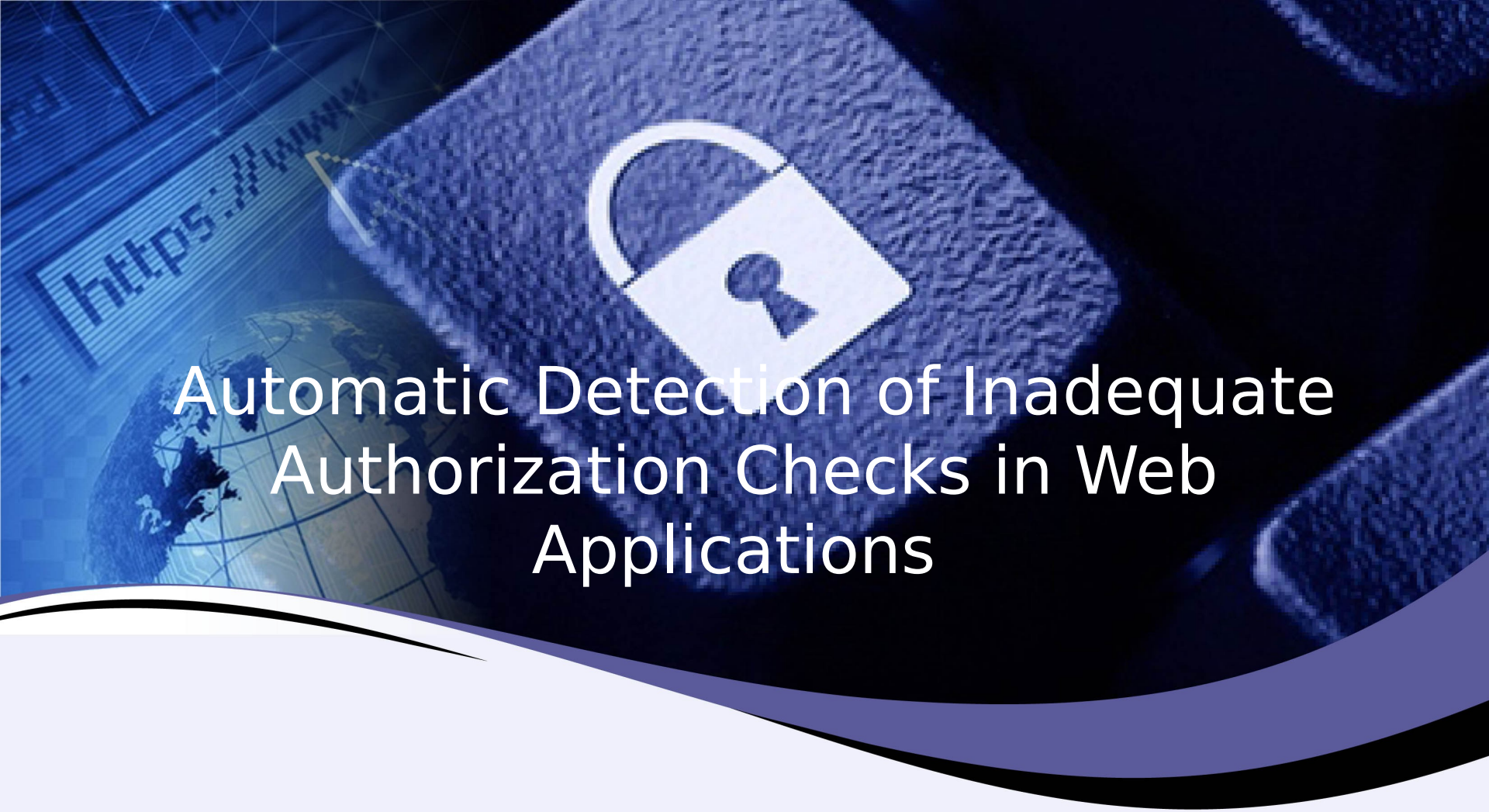


Automatic Detection of Inadequate Authorization Checks in Web Applications

OWASP

The Open Web Application Security Project

About Me

OWASP
The Open Web Application Security Project

Name	**Alvaro Muñoz**
Organization	HP Software Security Research
Currently	Researching the security impact of new technologies. Especially interested in Web, any language, any framework.
In previous episodes	Application Security Consultant Pentester
Other Stuff	CTF player, OSCP, GWAPT, CISSP ...
Location	Madrid, Spain
Contact	alvaro.munoz@hp.com @pwntester

About Me

OWASP
The Open Web Application Security Project

Name	**Dyvia Muthukumaran**
Organization	Imperial College
Currently	Postdoctoral researcher at Imperial College Working on security issues in Cloud Computing
In previous episodes	Ph.D at Penn State. Thesis work entailed automated authorization hook placement.
Location	London, UK
Contact	dmuthuku@imperial.ac.uk

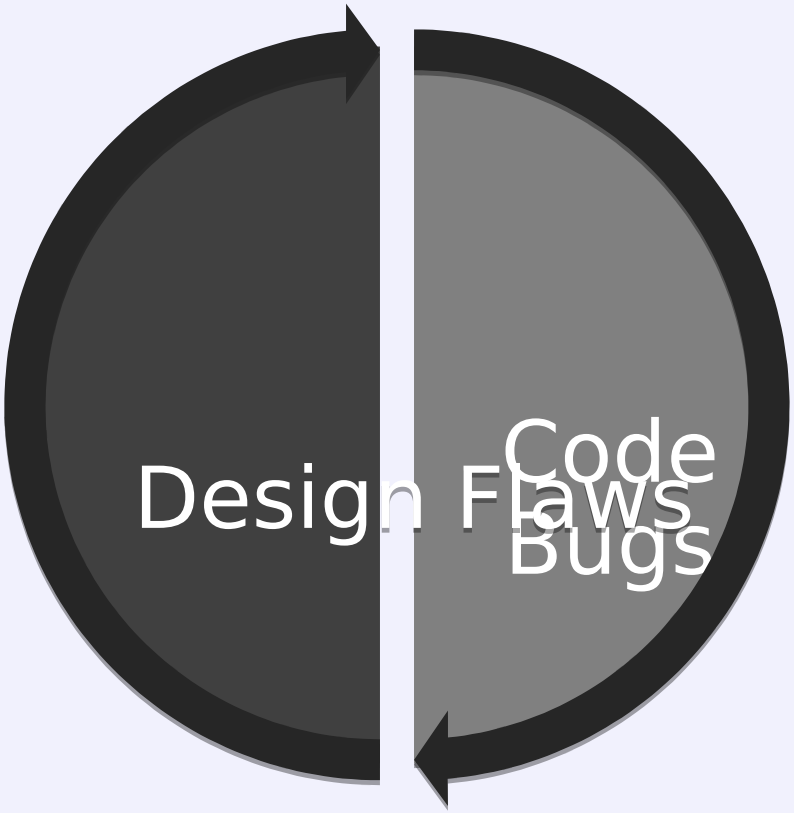
Imperial College London

Agenda

OWASP
The Open Web Application Security Project

- Agenda
  - Design Flaws vs Code Level Bugs
  - Why should we care?
  - Current detection techniques
  - Proposed solution

Application Security Duality

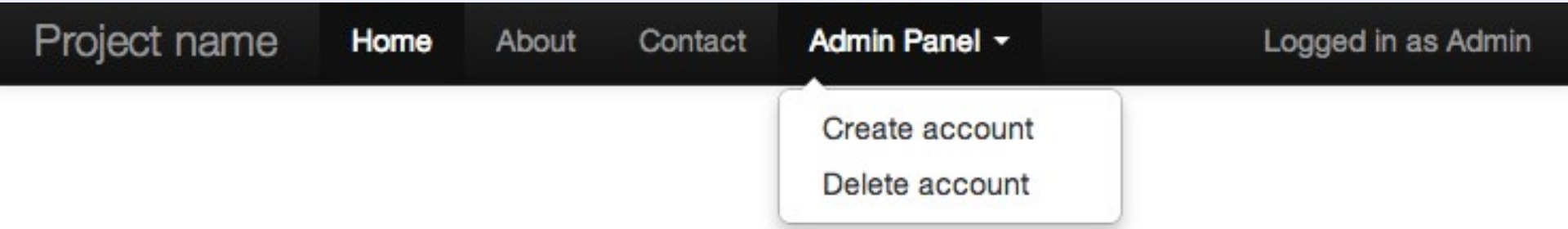



AA 550DT

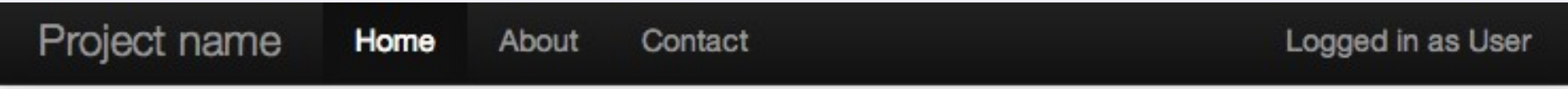
OWASP

The Open Web Application Security Project

Admin view



User view



OWASP

The Open Web Application Security Project

Is there anything wrong here?

```
@Repository
public class AccountDaoImpl implements AccountDAO  {

    @Autowired
    private SessionFactory sessionFactory;

    @PreAuthorize("hasRole('ROLE_ADMIN')")
    @Override
    public void createAccount(Account account) {
        this.sessionFactory.getCurrentSession().save(account);
    }

    @Override
    public void deleteAccount(Account account) {
        AccountEntity account = (AccountEntity) sessionFactory.getCurrentSession().load(AccountEntity.class, accountId);
        if (null != account) {
            this.sessionFactory.getCurrentSession().delete(account);
        }
    }

}
```

OWASP

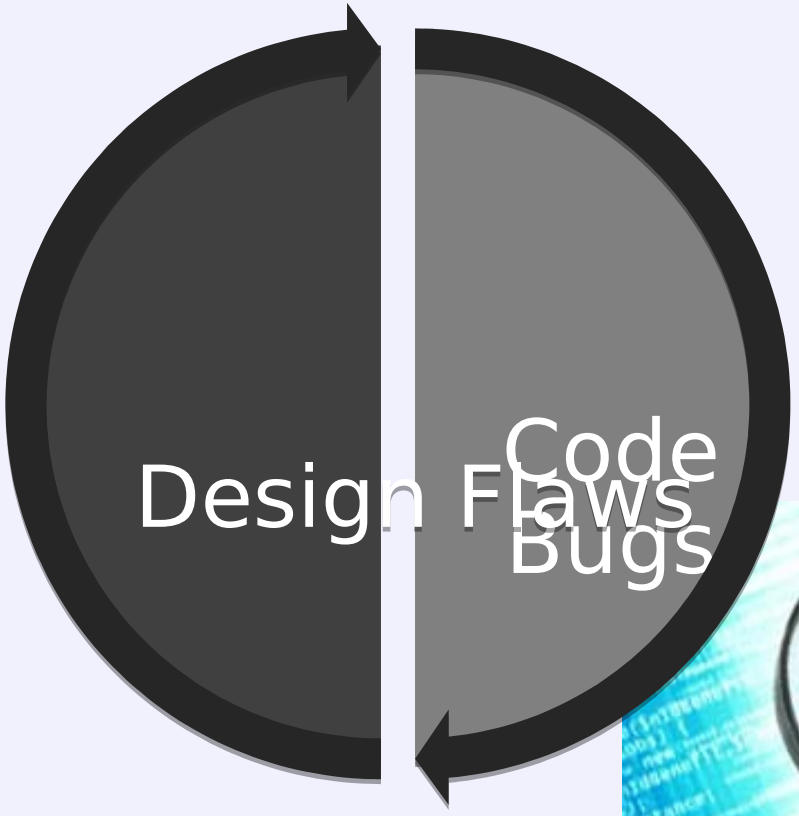
The Open Web Application Security Project

Any user can delete an account!!! Even if its not shown in UI ...

```
@Repository
public class AccountDaoImpl implements AccountDAO  {

    @Autowired
    private SessionFactory sessionFactory;

    @PreAuthorize("hasRole('ROLE_ADMIN')")
    @Override
    public void createAccount(Account account) {
        this.sessionFactory.getCurrentSession().save(account);
    }

    @Override
    public void deleteAccount(Account account) {
        AccountEntity account = (AccountEntity) sessionFactory.getCurrentSession().load(AccountEntity.class, accountId);
        if (null != account) {
            this.sessionFactory.getCurrentSession().delete(account);
        }
    }

}
```

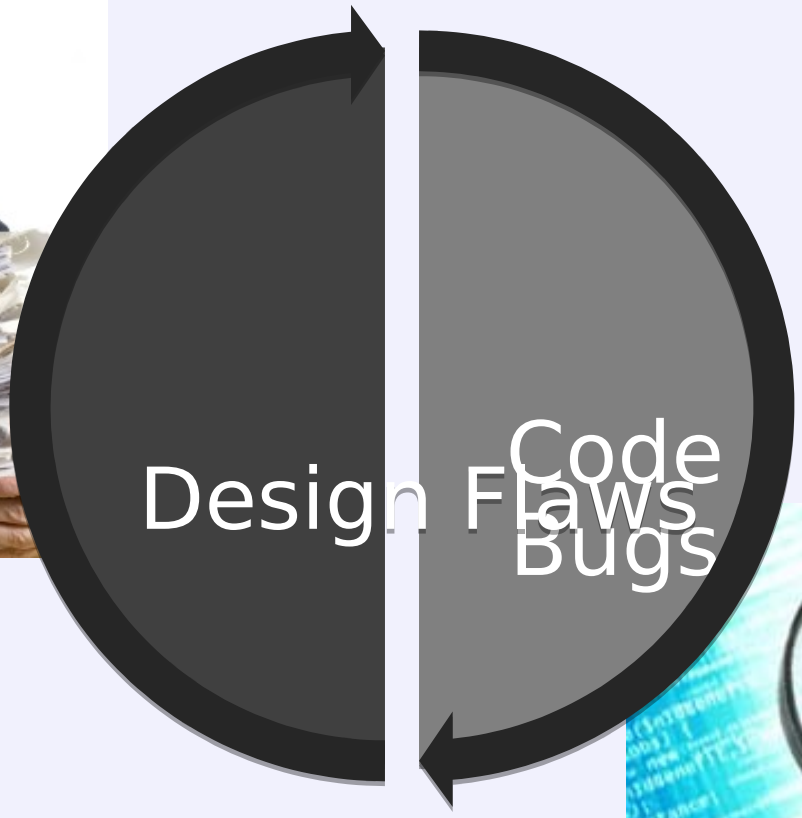
Detection

OWASP
The Open Web Application Security Project



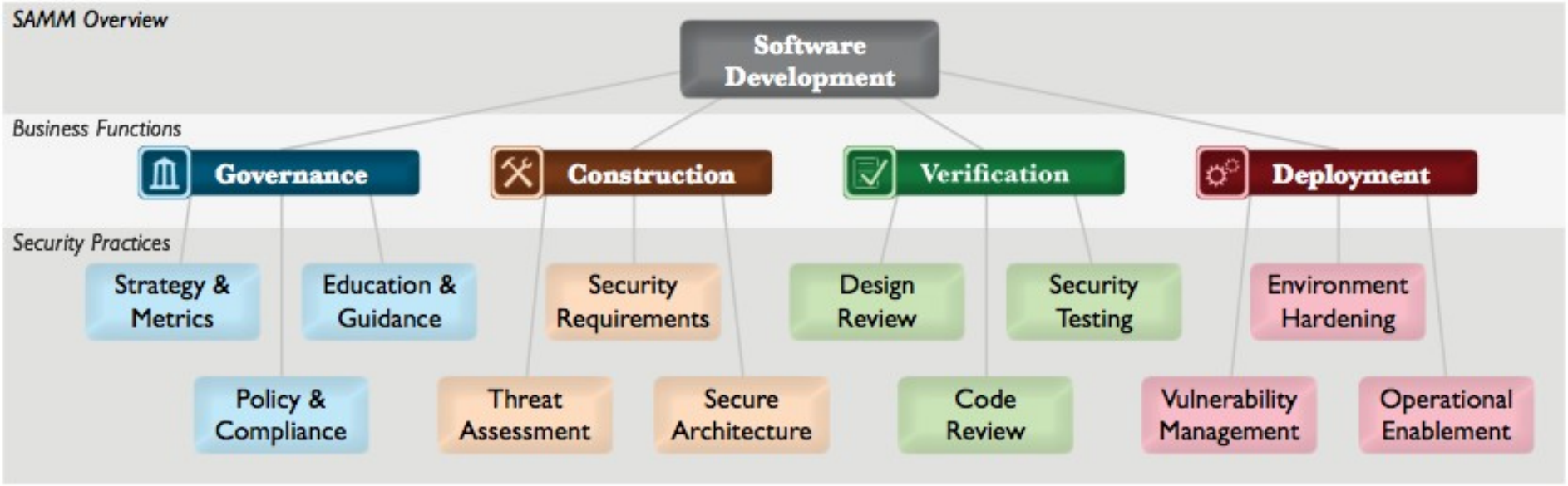
Detection

OWASP
The Open Web Application Security Project



OpenSAMM

OWASP
The Open Web Application Security Project



Source:
http://www.opensamm.org/

Code Review

OWASP
The Open Web Application Security Project

Code Review

...more on page 62

	CR 1	CR 2	CR 3
Objective	**Opportunistically find basic code-level vulnerabilities and other high-risk security issues**	**Make code review during development more accurate and efficient through automation**	**Mandate comprehensive code review process to discover language-level and application-specific risks**
Activities	A. Create review checklists from known security requirements B. Perform point-review of high-risk code	A. Utilize automated code analysis tools B. Integrate code analysis into development process	A. Customize code analysis for application-specific concerns B. Establish release gates for code review

Source:
http://www.opensamm.org/

Design Review

OWASP
The Open Web Application Security Project

Design Review

...more on page 58

	DR 1	DR 2	DR 3
Objective	**Support ad hoc reviews of software design to ensure baseline mitigations for known risks**	**Offer assessment services to review software design against comprehensive best practices for security**	**Require assessments and validate artifacts to develop detailed understanding of protection mechanisms**
Activities	A. Identify software attack surface B. Analyze design against known security requirements	A. Inspect for complete provision of security mechanisms B. Deploy design review service for project teams	A. Develop data-flow diagrams for sensitive resources B. Establish release gates for design review

Source:
http://www.opensamm.org/

OWASP Top Ten

OWASP
The Open Web Application Security Project

	2004	2007	2010	2013
A1	Unvalidated Input	Cross Site Scripting	Injection	Injection
A2	Broken Access Control	Injection Flaws	Cross Site Scripting	Broken Authentication and Session Management
A3	Broken Authentication and Session Management	Malicious File Execution	Broken Authentication and Session Management	Cross-Site Scripting
A4	Cross Site Scripting	Insecure Direct Object Reference	Insecure Direct Object References	Insecure Direct Object References
A5	Buffer Overflow	Cross Site Request Forgery	Cross Site Request Forgery	Security Misconfiguration
A6	Injection Flaws	Information Leakage and Improper Error Handling	Security Misconfiguration	Sensitive Data Exposure
A7	Improper Error Handling	Broken Authentication and Session Management	Insecure Cryptographic Storage	Missing Function Level Access Control
A8	Insecure Storage	Insecure Cryptographic Storage	Failure to Restrict URL Access	Cross-Site Request Forgery
A9	Application Denial of Service	Insecure Communications	Insufficient Transport Layer Protection	Using Known Vulnerable Components
A10	Insecure Configuration Management	Failure to Restrict URL Access	Unvalidated Redirects and Forwards	Unvalidated Redirects and Forwards

Source: http://h30499.www3.hp.com/t5/HP-Security-Research-Blog/OWASP-Top-Ten-2013/ba-p/6046369

OWASP
The Open Web Application Security Project

A4-Insecure Direct Object References

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data. Many web applications check URL access rights before rendering protected links and buttons. However, applications need to perform similar access control checks each time these pages are accessed, or attackers will be able to forge URLs to access these hidden pages anyway.

Source:
https://www.owasp.org/index.php/Top_10_2013-

OWASP
The Open Web Application Security Project

A7-Missing Function Level Access Control

Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.

Source: https://www.owasp.org/index.php/Top_10_2013-

Is this the *real life*? Is this just fantasy?

OWASP
The Open Web Application Security Project

Bloomberg grabs NetApp's earnings early, second case in a week

By Dominic Jones *on November 18, 2010*

NETAPP Inc. (NASDAQ:NTAP) has become the second company in less than a week to have earnings information leaked from an unsecured area of its corporate website.

Bloomberg confirmed to Dow Jones that it retrieved unpublished financial data from the company's website **more than an hour** before its scheduled release.

NASDAQ officials **halted** NetApp's stock at 3:11 pm ET after the stock had dropped 6.5% on the leaked news, which **hit trading desks around 2:45 pm ET**.

The storage and data management company hurriedly issued its earnings release via Market Wire at 3:31 pm and filed the same information in an 8-K on EDGAR at 3:34 pm.

Source: http://irwebreport.com/20101118/bloomberg-grabs-netapps-earnings-early-second-case-in-a-week/

Is this the *real life*? Is this just fantasy?

OWASP
The Open Web Application Security Project

The Register®

Data Centre | Software | Networks | **Security** | Policy | Business | Hardware | Science | Bootnotes | Columnists

Citigroup hack exploited easy-to-detect web flaw

Brute force attack exposes 200,000 accounts

By Dan Goodin, 14 Jun 2011

18

RELATED STORIES

Fund manager withdraws legal threat over security vuln

Financial company heavies researcher for reporting vulnerability

Hackers penetrate website for Nokia developers

Citigroup hit with *another* data leak

Adaptable System Recovery (ASR) for Linux virtual machines

Hackers who stole bank account details for 200,000 Citigroup customers infiltrated the company's system by exploiting a garden-variety security hole in the company's website for credit card users, according to a report citing an unnamed security investigator.

The New York Times reported that the technique allowed the hackers to leapfrog from account to account on the Citi website by changing the numbers in the URLs that appeared after customers had entered valid usernames and passwords. The hackers wrote a script that automatically repeated the exercise tens of thousands of times, the *NYT* said in an article published Monday.

"Think of it as a mansion with a high-tech security system – that the front door wasn't locked tight," reporters Nelson D. Schwartz and Eric Dash wrote.

The underlying vulnerability, known as an insecure direct object reference, is so common that it's included in the Top 10 Risks list compiled by the Open Web Application Security Project. It results when developers expose direct references to confidential account numbers instead of using substitute characters to ensure the account numbers are kept private.

Source: http://www.theregister.co.uk/2011/06/14/citigroup_website_hack_simple/

Current Approaches

OWASP
The Open Web Application Security Project

- Dynamic approach
  - Scan application with two or more roles
  - Compare results
  - Limitations:
    - Dynamic
      - Set up
    - Can only detect missing checks
    - High false positive rate

A better solution

OWASP
The Open Web Application Security Project

- Requirements
  - Works on Web Applications
  - Static
  - Finds both missing and inconsistent checks
  - **Does not require any training or preparation**

Identifying inadequate access checks

OWASP
The Open Web Application Security Project

1. Create Specification

What is currently being access checked?

2. Identify Anomalies

What similar operations are not been checked or checked differently?

3. Suggest Remediation

What checks should be added or modified?

OWASP
The Open Web Application Security Project

1. Create Specification

OWASP
The Open Web Application Security Project

Identify access checked methods

- Configuration Files:
  - URLs
  - PointCuts
- Source Code:
  - Annotations
  - if-else checks
- Consider super classes and call traces

Intercepting URLs

OWASP
The Open Web Application Security Project

```
<http use-expressions="true">
    <intercept-url
      pattern="/admin*"
      access="hasRole('admin')
    />
    <intercept-url
      pattern=”/index*"
      access=”isAuthenticated()”
    />
  </http>
```

Pointcut-based

OWASP
The Open Web Application Security Project

```
<global-method-security>
    <protect-pointcut expression="execution(*
com.*Service.*(..))"
        access="ROLE_USER"/>
  </global-method-security>
```

Pointcut-based

OWASP
The Open Web Application Security Project

```
<bean id="methodInterceptor"
class="_MethodSecurityInterceptor">
  <property name="authenticationManager"
ref="authManager"/>
  <property name="securityMetadataSource">
        <value>
      org.demo.AccountService.createAccount=ROLE_USER
      org.demo.AccountService.delete*=ROLE_ADMIN
        </value>
  </property>
</bean>
<bean id="accountService"
class="ord.demo.AccountServiceImpl">
  <sec:intercept-methods>
      <sec:protect access="ROLE_USER"
method="createAccount">
      <sec:protect access="ROLE_ADMIN" method="delete*">
  </property>
</bean>
```

Annotations

OWASP
The Open Web Application Security Project

```
public interface BankService {
    @RequiresRole(“teller”)
    public Account post(Account account, double amount);
    @RequiresPermission(“account:create”)
    public Account[] findAccounts();
 }
```

```
public interface BankService {
    @Secured("IS_AUTHENTICATED_ANONYMOUSLY")
    public Account readAccount(Long id);

    @PreAuthorize(isAuthenticated() and
hasRole(“ROLE_USER”))
    @PreFilter(hasPermission(filterObject,’read’))
    public Account[] findAccounts();
 }
```

Hiding/Disabling functionality

OWASP
The Open Web Application Security Project

```
<p>Hi <shiro:guest>Guest</shiro:guest>
<shiro:user><c:out value="$
{account.givenName}"/></shiro:user>!
```

```
<security:authorize ifAnyGranted="ROLE_ADMIN">
      <input type="submit" value="<spring:message
code="label.add"/>"/>
</security:authorize>
<sec:accesscontrollist hasPermission="1,2"
domainObject="someObject">
    This will be shown if the user has either of the
permissions
    represented by the values "1" or "2" on the given object.
</sec:accesscontrollist>
```

Programmatic

OWASP
The Open Web Application Security Project

```
// get the current subject
Subject currentUser = SecurityUtils.getSubject();

If (currentUser.hasRole(“administrator”)) {
    do_something();
} else {
    do_something_different();
}
```

Gather all protected operations

- What objects, classes, methods are accessed within access checked methods?
- For each operation:
  - Function
  - Variable (instance object, argument)
- Discard:
  - Utility/Helper function calls
  - Common Function calls (Heuristically detected)

Example

OWASP
The Open Web Application Security Project

```
public class CocktailServiceImpl implements CocktailService{

    List<Cocktail> cocktails=new ArrayList<Cocktail>();

   @PreAuthorize(hasRole('ADMIN'))
    public Cocktail getCocktail(int id) {
       return cocktails.get(int );
    }

    public Cocktail deleteCocktail(int id) {
      Cocktail cocktail = cocktails.get(int );
      cocktails.remove(int);
      return cocktail;
    }
}
```

Example

CocktailService.java

```
public interface CocktailService   {
{

@PreAuthorize(hasRole('ADMIN')
)
    Cocktail getCocktail(int );
     ....
}
```

@PreAuthorize(hasRole('ADMIN'))

CocktailServiceImpl.java

```
@RequestMapping("/admin/getDrink)
public getCocktail(int id) {
   return lookupCocktail(id);
}
```

isAuthenticated()

URL -> Access Check

Security-config.xml

```
<http>
<intercept-url
  pattern="/admin/*"

access="isAuthenticate
d()"
/>
```

Method -> URL

- **Annotation**
  @RequestMapping
- **Configuration**
  PointCuts, …

Method -> URL -> Access Check

@PreAuthorize(hasRole('ADMIN')) && isAuthenticated()

CocktailServiceImpl.java

```
private lookupCocktail()  {
    Cocktail c =
cocktails.get();
     return c;
}
```

OWASP
The Open Web Application Security Project

2. Identify Anomalies

Identify Anomalies

OWASP
The Open Web Application Security Project

- Given a class *c* where some of methods have access checks, are there methods that do not?
- If so, are they performing any of the same types of accesses as the ones that do?

Identify Anomalies

OWASP
The Open Web Application Security Project

- For each non-access controlled method examine all operations performed within function scope:
  - Identify Function
  - Identify Variable
    - Instance object, argument
- Fine Tune operations:
  - Map operations with CRUD actions

Identify Anomalies

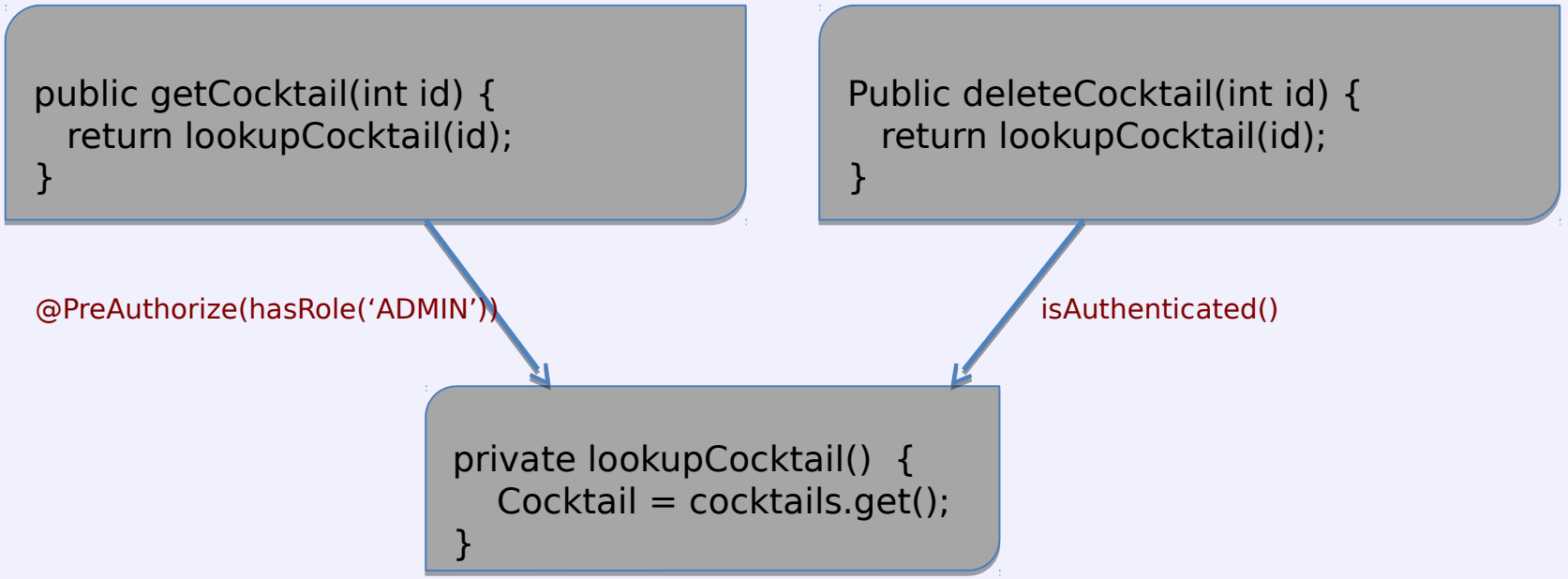
OWASP
The Open Web Application Security Project

Same variable and same method	[SAMEVAR+SAMEMETHOD]
Same variable and different method	[SAMEVAR+DIFFMETHOD]
Same variable type and same method	[SAMETYPE+SAMEMETHOD]
Same variable type and different method	[SAMETYPE+DIFFMETHOD]

- ***getCocktail()***
  - Operations:
    - this.cocktails + get()
  - Access Checks:
    - PreAuthorize(hasRole('Admin')).

- ***deleteCocktail()***
  - Operations:
    - this.cocktails + get()

Identify Anomalies: Inconsistency



OWASP
The Open Web Application Security Project

3. Suggest Remediation

Suggest Remediation

OWASP
The Open Web Application Security Project

- Present the developer with precise details of the anomalies found and the evidences supporting the finding
- If several evidences are found, present the most similar in terms of operations performed
- Provide developers with a set of access control checks based on evidences

Summary

- **getCocktail()**
  - CocktailServiceImpl + this.cocktails + get()
  - Access Checks:
    - PreAuthorize(hasRole('Admin'))

Create Specific ation

- **deleteCocktail()**
  - Operations: CocktailServiceImpl + this.cocktails + get()

Identify Anomali es

- **deleteCocktail()**
  - Add Check:
    - PreAuthorize(hasRole('Admin'))
    - Evidence: getCocktail()

Remedi ation

OWASP
The Open Web Application Security Project

IN THE WILD

FOR THE CODE IS DARK
AND FULL OF FLAWS

In the wild

OWASP
The Open Web Application Security Project

- Mifos:
  - Very large open source microfinance application.
  - 323,007 Java LOC
  - 122224 XML LOC
  - It uses spring annotations in addition to custom checks
  - 78 anomalies were found

In the wild: mifos

OWASP
The Open Web Application Security Project

Anomaly	**Location:** ClientServiceFacadeWebTier:859 **Function Signature:** public String transferClientToBranch(String, Short)
Suspicious operations	**[SAMEVAR+SAMEMETHOD]** this.customerService + transferClientTo @ 842
Evidence	**Function Signature:** public String transferClientToGroup(Integer, String, Integer) **Protected Operations:** this.customerService+ transferClientTo @ 733
Recommended Annotation Set	@PreAuthorize("isFullyAuthenticated() and hasRole('ROLE_CAN_UPDATE_GROUP_MEMBERSHIP_OF_CLIENT')")

```java
@Override
public String transferClientToGroup(Integer groupId, String clientGlobalCustNum, Integer previousClientVersionNo) {
    MifosUser user = (MifosUser) SecurityContextHolder.getContext().getAuthentication().getPrincipal();
    UserContext userContext = toUserContext(user);

    ClientBO client;
    try {
        client = this.customerService.transferClientTo(userContext, groupId, clientGlobalCustNum, previousClientVersionNo);
        return client.getGlobalCustNum();
    } catch (CustomerException e) {
        throw new BusinessRuleException(e.getKey(), e);
    }
}

@Override
public String transferClientToBranch(String globalCustNum, Short officeId) {

    MifosUser user = (MifosUser) SecurityContextHolder.getContext().getAuthentication().getPrincipal();
    UserContext userContext = toUserContext(user);

    OfficeBO receivingBranch = this.officeDao.findOfficeById(officeId);
    ClientBO client = this.customerDao.findClientBySystemId(globalCustNum);
    client.updateDetails(userContext);

    this.customerService.transferClientTo(client, receivingBranch);

    return client.getGlobalCustNum();
}
```

Source: https://github.com/mifos/head/blob/0d9cdffeb07bcbeb75ffa4f9107272c6694a00a2/appdomain/src/main/java/org/mifos/application/servicefacade/ClientServiceFacadeWebTier.java

OWASP

The Open Web Application Security Project

```
@PreAuthorize("isFullyAuthenticated() and hasRole('ROLE_CAN_UPDATE_GROUP_MEMBERSHIP_OF_CLIENT')")
String transferClientToGroup(Integer parentGroupIdValue, String clientGlobalCustNum, Integer previousClientVersionNo);

@PreAuthorize("isFullyAuthenticated()")
List<SavingsDetailDto> retrieveSavingsInUseForClient(Integer clientId);

String transferClientToBranch(String globalCustNum, Short officeId);
```

Source: https://github.com/mifos/head/blob/0d9cdffeb07bcbeb75ffa4f9107272c6694a00a2/serviceInterfaces/src/main/java/org/mifos/application/servicefacade/ClientServiceFacade.java

In the wild: mifos

OWASP
The Open Web Application Security Project

Anomaly	**Location:** QuestionnaireServiceFacadeImpl:94 **Function Signature:** public QuestionGroupDetail createQuestionGroup (QuestionGroupDetail)
Suspicious operations	**[SAMEVAR+SAMEMETHOD]** this.questionnaireService + defineQuestionGroup @ 94
Evidence	**Function Signature:** public QuestionGroupDetail createActiveQuestionGroup(QuestionGroupDetail) **Protected Operations:** this.questionnaireService + defineQuestionGroup @ 100
Recommended Annotation Set	@PreAuthorize("isFullyAuthenticated() and hasRole('ROLE_CAN_ACTIVATE_QUESTION_GROUPS') ")

OWASP

The Open Web Application Security Project

```
@Override
public QuestionGroupDetail createQuestionGroup(QuestionGroupDetail questionGroupDetail) throws SystemException {
    questionGroupDetail.setActivityId(addActivityPermission(questionGroupDetail.getTitle(), questionGroupDetail.getId()));
    return questionnaireService.defineQuestionGroup(questionGroupDetail);
}

@Override
public QuestionGroupDetail createActiveQuestionGroup(QuestionGroupDetail questionGroupDetail) throws SystemException {
    questionGroupDetail.setActivityId(addActivityPermission(questionGroupDetail.getTitle(), questionGroupDetail.getId()));
    return questionnaireService.defineQuestionGroup(questionGroupDetail);
}
```

Source:
https://github.com/mifos/head/blob/aa1d2ac985bfbe7ea4e07a0eb7a22cef7ba92bc1/appdomain/src/main/java/org/mifos/platform/questionnaire/service/QuestionnaireServiceFacadeImpl.java

OWASP

The Open Web Application Security Project

```
QuestionGroupDetail createQuestionGroup(QuestionGroupDetail questionGroupDetail) throws SystemException;

void createQuestionLinks(List<QuestionLinkDetail> questionLinks);

void createSectionLinks(List<SectionLinkDetail> sectionLinks);

@PreAuthorize("isFullyAuthenticated() and hasRole('ROLE_CAN_ACTIVATE_QUESTION_GROUPS')")
QuestionGroupDetail createActiveQuestionGroup(QuestionGroupDetail questionGroupDetail) throws SystemException;
```

Source: https://github.com/mifos/head/blob/6fe9141e4491194181c7ec85ef0adc3773208dcd/serviceInterfaces/src/main/java/org/mifos/platform/questionnaire/service/QuestionnaireServiceFacade.java

In the wild: mifos

OWASP
The Open Web Application Security Project

Anomaly	**Location:** SystemInformationServiceFacadeWebTier:90 **Function Signature:** public String getServerInformation(ServletContext, Locale)
Suspicious operations	**[SAMETYPE+SAMEMETHOD]** org.mifos.application.admin.SystemInfo + SystemInfo @ 90
Evidence	**Function Signature:** public SystemInfor getSystemInformation(ServletContext, Locale) **Protected Operations:** org.mifos.application.admin.SystemInfo + SystemInfo @ 46
Recommended Annotation Set	PreAuthorize(isFullyAuthenticated() and hasRole('ROLE_VIEW_SYSTEM_INFO'))

```
@Override
public String getServerInformation(ServletContext context,
                Locale locale) {

        try {
        DatabaseMetaData metaData = StaticHibernateUtil.getSessionTL().connection().getMetaData();

        final SystemInfo systemInfo = new SystemInfo(metaData, context, locale, true);
        return  systemInfo.getApplicationServerInfo();

    } catch (HibernateException e) {
        throw new MifosRuntimeException(e);
    } catch (SQLException e) {
        throw new MifosRuntimeException(e);
    }
}
```

```
        final SystemInfo systemInfo = new SystemInfo(metaData, context, locale, true);
        systemInfo.setCustomReportsDir(BirtReportsUploadAction.getCustomReportStorageDirectory());

        return new SystemInformationDto(
                systemInfo.getApplicationServerInfo(),
                systemInfo.getApplicationVersion(),
                systemInfo.getBuildDate(),
                systemInfo.getBuildNumber(),
                systemInfo.getCommitIdentifier(),
                systemInfo.getCustomReportsDir(),
                systemInfo.getDatabaseName(),
                systemInfo.getDatabasePort(),
                systemInfo.getDatabaseServer(),
                systemInfo.getDatabaseUser(),
                systemInfo.getDatabaseVendor(),
                systemInfo.getDatabaseVersion(),
                systemInfo.getDriverName(),
```

Source:
https://github.com/mifos/head/blob/e271189b8ec71e5724ffcf189f0aef7249896e13/application/src/main/java/org/mifos/application/admin/servicefacade/SystemInformationServiceFacadeWebTier.java

```
public interface SystemInformationServiceFacade {

    @PreAuthorize("isFullyAuthenticated() and hasRole('ROLE_VIEW_SYSTEM_INFO')")
    SystemInformationDto getSystemInformation(ServletContext context, Locale locale);

        String getServerInformation(ServletContext context, Locale locale);
}
```

Source:
https://github.com/mifos/head/blob/0d9cdffeb07bcbeb75ffa4f9107272c6694a00a2/serviceInterfaces/src/main/java/org/mifos/application/admin/servicefacade/SystemInformationServiceFacade.java

In the wild

OWASP
The Open Web Application Security Project

- pgGallery
  - Small photo sharing application
    - 1897 Java LOC
    - 419 XML LOC
  - 8 anomalies identified

In the wild: pgGallery

OWASP
The Open Web Application Security Project

Anomaly	**Location:** AlbumService.java:42 **Function Signature:** public List<Album> getBreadcrumbById (BigDecimal)
Suspicious operations	**[SAMEVAR+DIFFMETHOD]** this.albumMapper + getBreadcrumbById @ 42
Evidence 1	**Function Signature:** public List<Album> getByParent (BigDecimal) **Protected Operations:** this.albumMapper + getByParent @ 30
Evidence 2	**Function Signature:** public List<Album> getRoot() **Protected Operations:** this.albumMapper + getRoot @ 25
Recommended Annotation Set	PostFilter (hasAnyRole('ROLE_USER','ROLE_ADMIN')) PostFilter (filterObject.isPublic() == true)

```java
@Service
public class AlbumService {
	@Autowired
	private AlbumMapper albumMapper = null;

	@PostFilter("hasAnyRole('ROLE_USER','ROLE_ADMIN') or filterObject.isPublic() == true")
	public List<Album> getRoot(){
		return albumMapper.getRoot();
	}

	@PostFilter("hasAnyRole('ROLE_USER','ROLE_ADMIN') or filterObject.isPublic() == true")
	public List<Album> getByParent(BigDecimal id){
		return albumMapper.getByParent(id);
	}

	@PostAuthorize("hasAnyRole('ROLE_USER','ROLE_ADMIN') or returnObject.isPublic() == true")
	public Album getById(BigDecimal id){
		return albumMapper.getById(id);
	}

	public List<Album> getBreadcrumbById(BigDecimal id){
		if(id == null) {
			return new ArrayList<Album>();
		}
		List<Album> crumbs = albumMapper.getBreadcrumbById(id);
		Collections.reverse(crumbs);
		return crumbs;
	}
```

Source: https://github.com/chotchki/pgGallery/blob/master/src/main/java/pgGallery/db/service/AlbumService.java

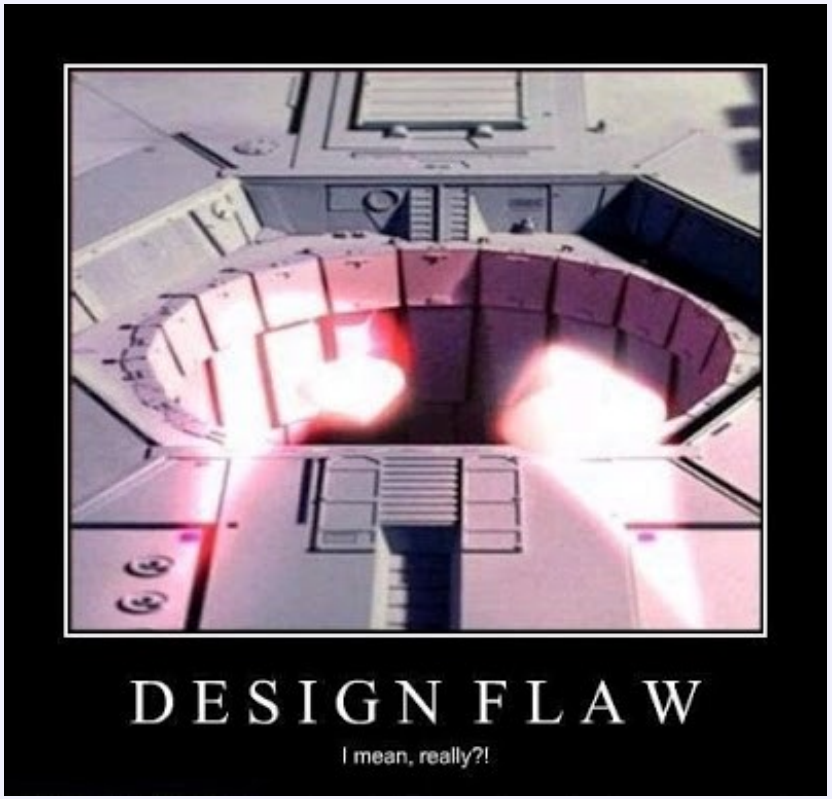
Next steps

OWASP
The Open Web Application Security Project

- Handle custom authorization checks.
- Reduce false positives:
  - Increase granularity of operations.
  - Map operations to CRUD actions.
- Extending to other frameworks/languages

OWASP
The Open Web Application Security Project

Thanks



alvaro.munoz@hp.com