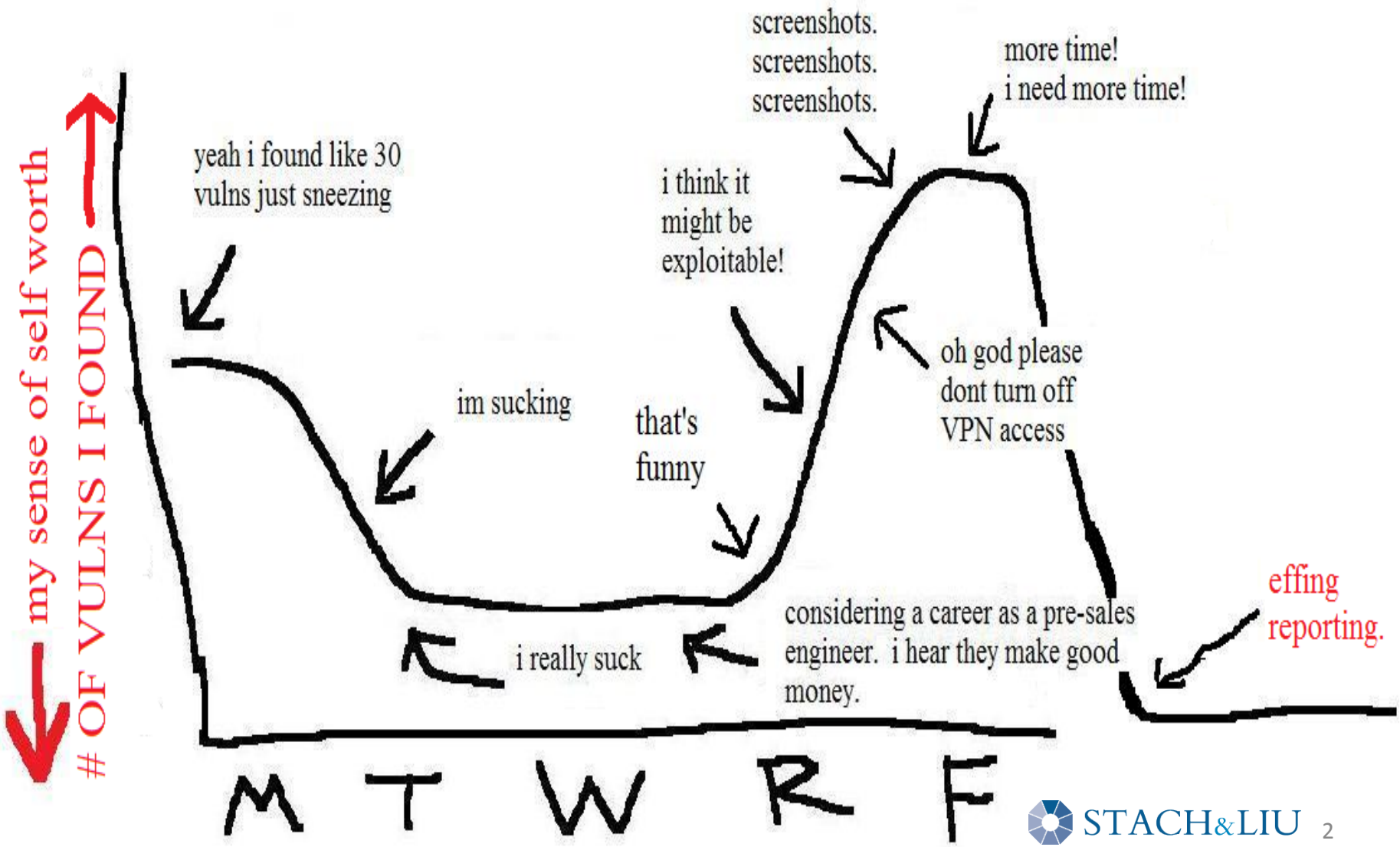
Real-world Code Review

SANS Penetration Testing and Web Application Attacks Summit

02 JUNE 2009

Presented by Vinnie Liu
Stach & Liu, LLC

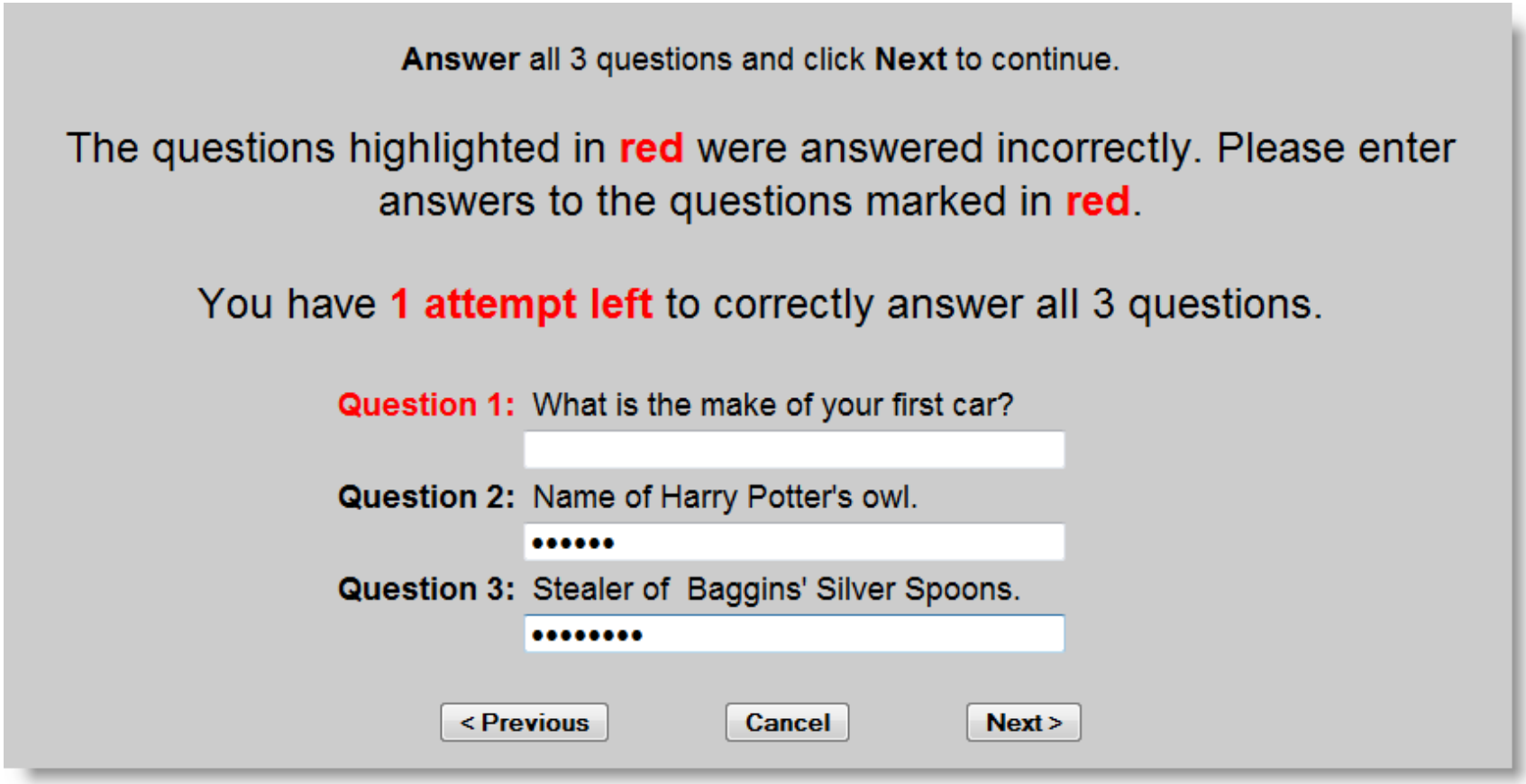
a typical week



topics

- Right Tool, Right Time, Right Place
- HQL Manipulation Attacks
- Slow State Manipulation Attacks

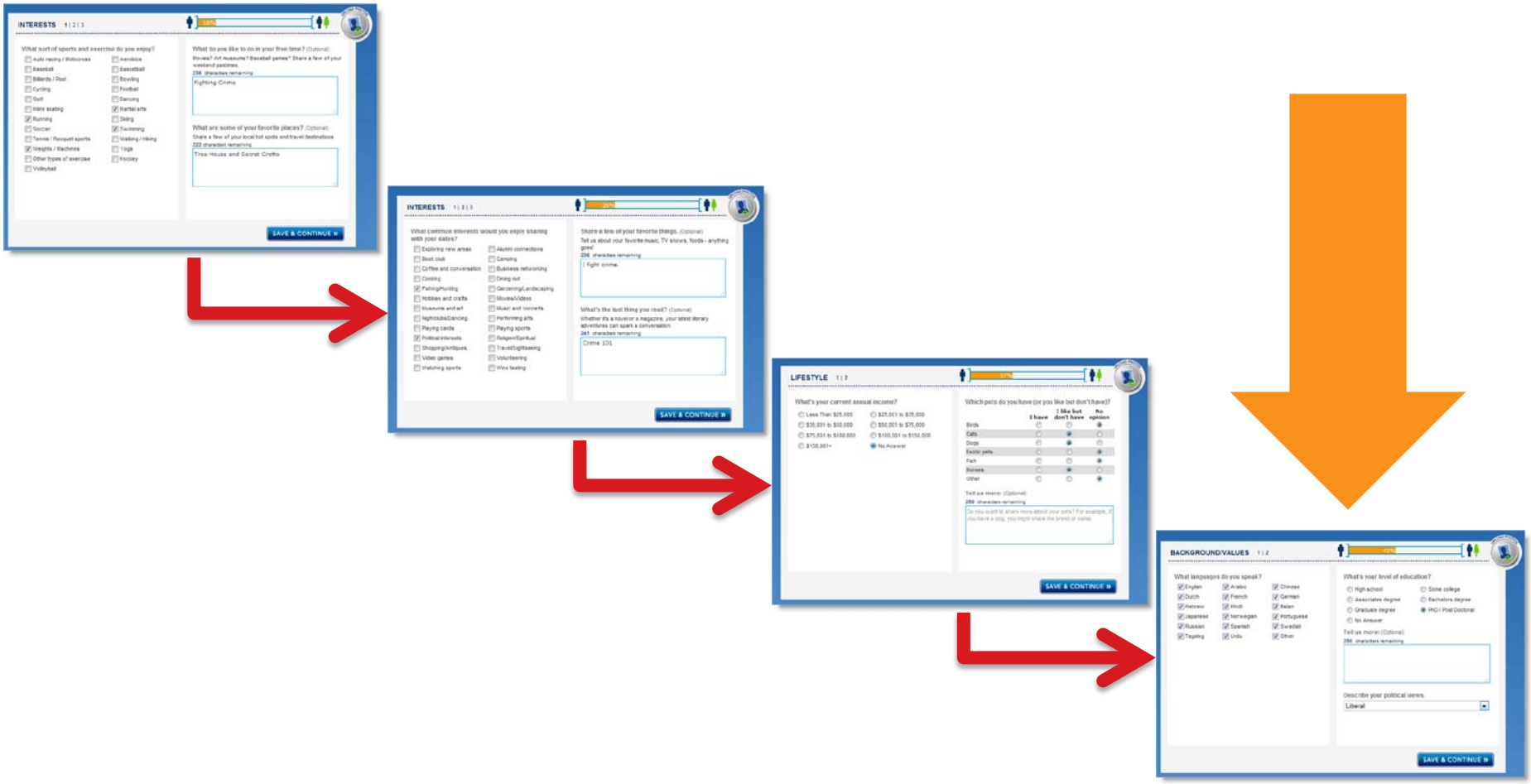
security questions



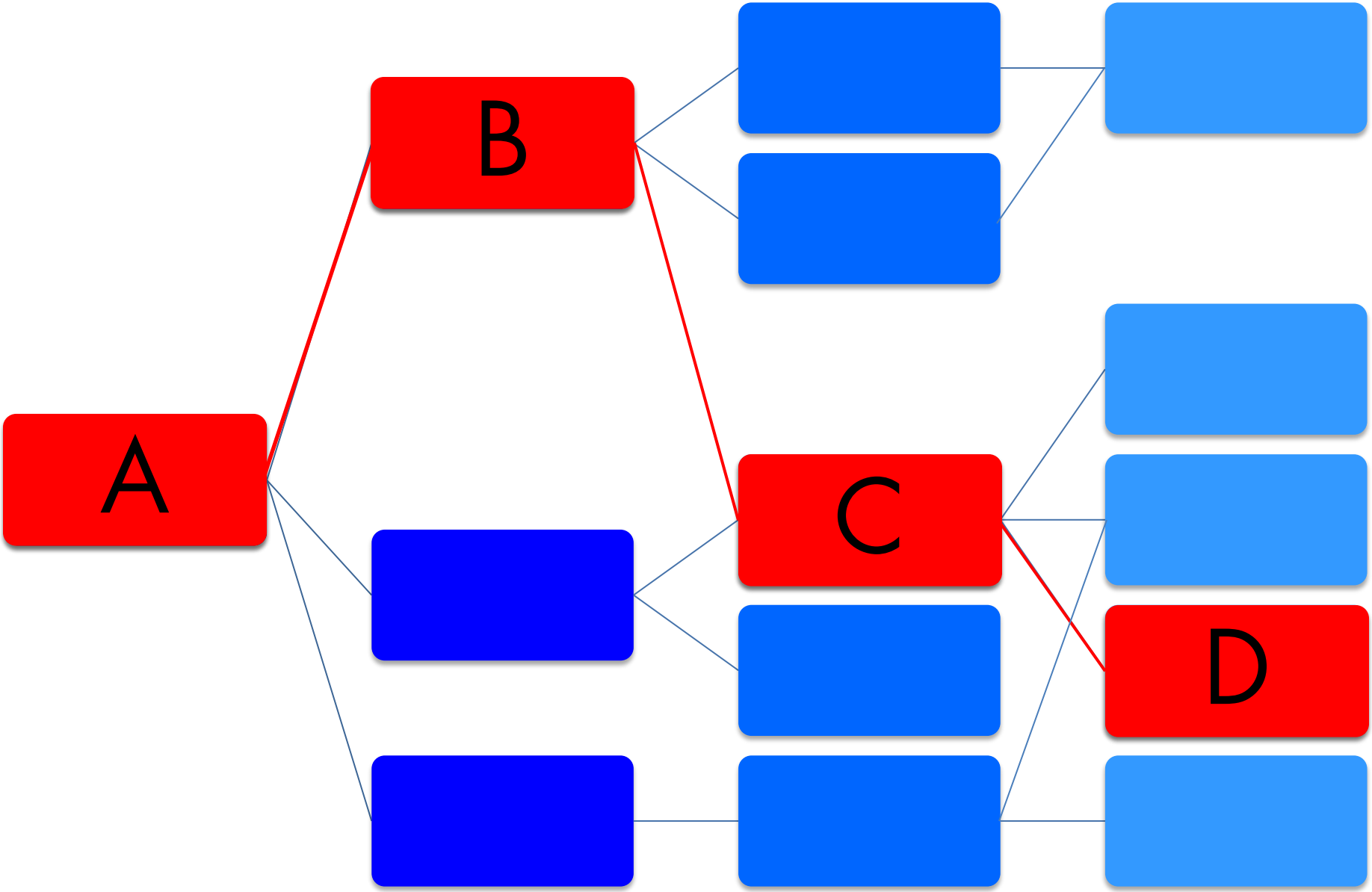
directory indexing

opt/	14-Nov-2007 18:39	-
pam.conf	31-Jul-2004 22:34	552
pam.d/	17-Sep-2008 19:17	-
papersize	04-Dec-2007 15:44	7
passwd	30-Jul-2008 15:12	4.8K
passwd-	25-Jul-2008 09:27	4.8K
passwd.bak	26-Nov-2007 23:53	1.4K
perl/	05-Feb-2008 17:12	-
postfix/	22-Aug-2008 20:49	-

workflow SQLi

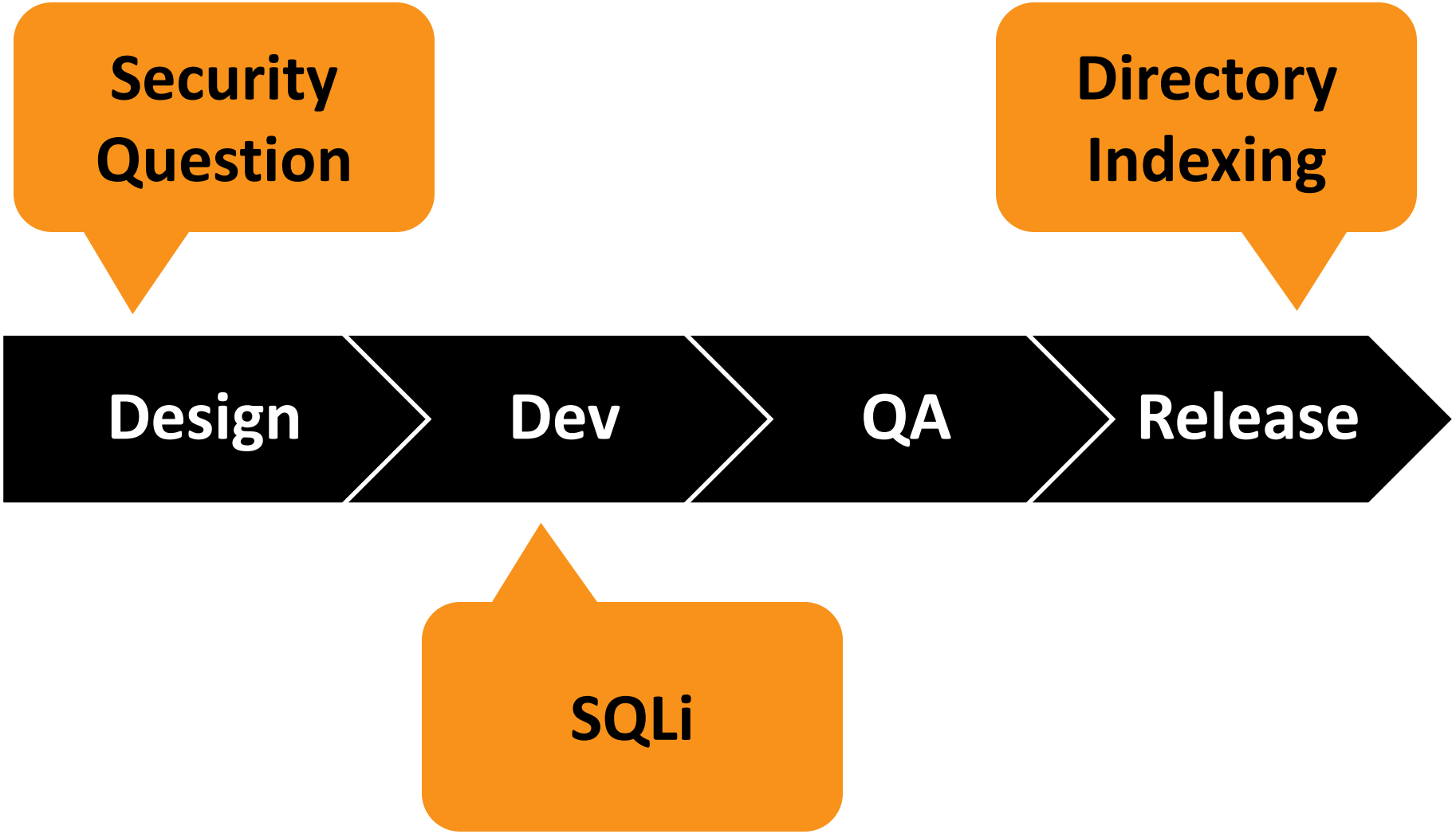


STACH&LIU



STACH&LIU

```
1  String sYum =
       request.getParameter("Treat");

2  if (!sYum.equals("ricekrispy"))
         return;

3  String sQuery = "
         SELECT *
         FROM TastySnacks
         WHERE Treat = '"+sYum+"'";

4  executeQuery(sQuery);
```

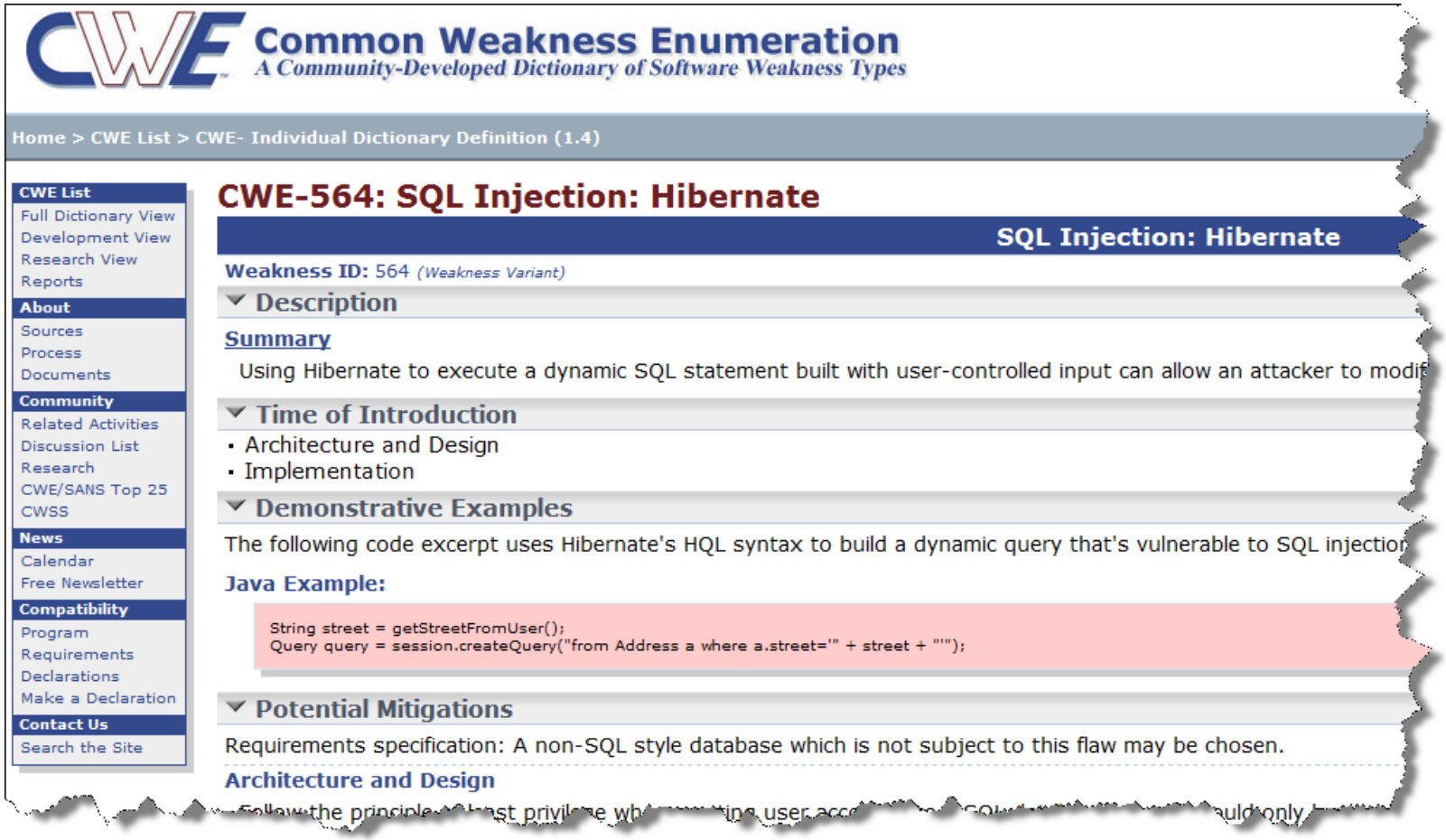

Security Question
Directory Indexing
Design
Dev
QA
Release
SQLi

topics

- Right Tool, Right Time, Right Place
- HQL Manipulation Attacks
- Slow State Manipulation Attacks

"consider using persistence layers such as **Hibernate** or Enterprise Java Beans, which can provide significant protection against SQL injection if used properly"[1]

HQL injection



Common Weakness Enumeration
A Community-Developed Dictionary of Software Weakness Types

Home > CWE List > CWE- Individual Dictionary Definition (1.4)

CWE List
Full Dictionary View
Development View
Research View
Reports
About
Sources
Process
Documents
Community
Related Activities
Discussion List
Research
CWE/SANS Top 25
CWSS
News
Calendar
Free Newsletter
Compatibility
Program
Requirements
Declarations
Make a Declaration
Contact Us
Search the Site

CWE-564: SQL Injection: Hibernate

SQL Injection: Hibernate

Weakness ID: 564 (Weakness Variant)

Description

Summary

Using Hibernate to execute a dynamic SQL statement built with user-controlled input can allow an attacker to modif

Time of Introduction

- Architecture and Design
- Implementation

Demonstrative Examples

The following code excerpt uses Hibernate's HQL syntax to build a dynamic query that's vulnerable to SQL injectio

Java Example:

```
String street = getStreetFromUser();
Query query = session.createQuery("from Address a where a.street='" + street + "'");
```

Potential Mitigations

Requirements specification: A non-SQL style database which is not subject to this flaw may be chosen.

Architecture and Design

Follow the principle of least privilege wh ing user acc GO ould only

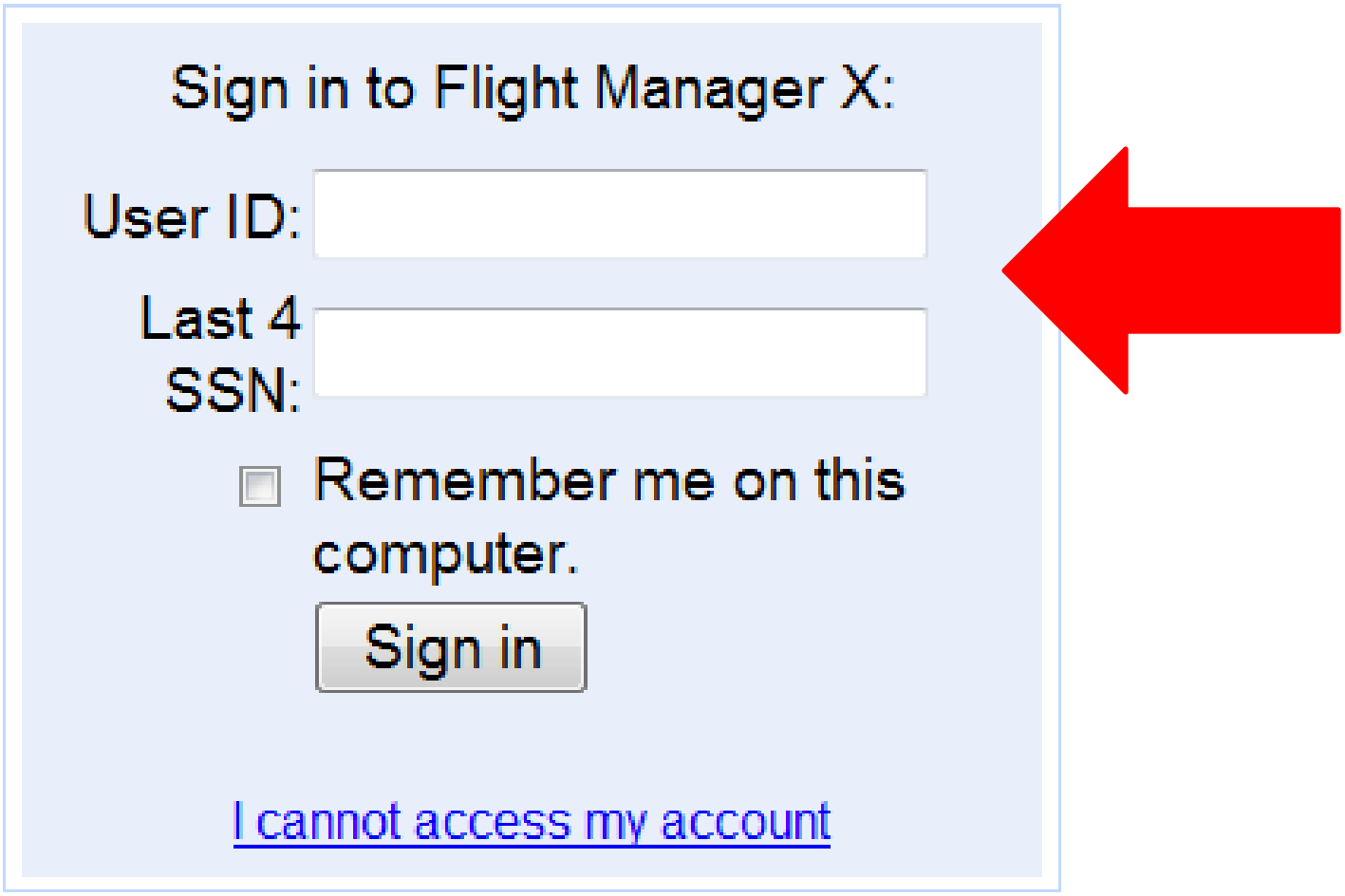
```
String query = “from Users u where
   u.ID = ‘” + uid + “’”;

session.find(query);
```

```
String query = “from Users u where
   u.ID = ?”;

session.find(query, uid,
   StringType);
```

unpublished vector

```
List users =

sess.createCriteria(User.class)
.add(

   Restrictions.sqlRestriction(
     “Users.ID = ‘” + uid + “’”)

.list();
```

HQL manipulation



```
1  String uid = r.getParameter("uid");
   String ssn = r.getParameter("ssn");

2  String query = "
            from Users u
            where u.ID = :id
            and u.SSN like :last4";

3  String[] p = {"id", "last4"};
   String[] v = {uid, '%' + ssn};

4  session.find(query, p, v);
```

simple bypass

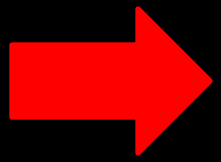
```
% 0 1 2 3 4 5 6 7 8 9                    5
```

```
from Users u
where u.ID = :id
and u.SSN like %5
```

SSN enumeration

%	0	1	2	3	4	**5**	6	7	8	9	**5**
	0	1	**2**	3	4	5	6	7	8	9	**25**
	0	1	2	3	4	5	**6**	7	8	9	**625**
	0	1	2	3	4	5	6	7	8	9	**0625**
	0	**1**	2	3	4	5	6	7	8	9	**10625**
	0	1	2	3	4	5	6	7	**8**	9	**810625**
	0	1	2	3	4	**5**	6	7	8	9	**5810625**
	0	1	2	3	**4**	5	6	7	8	9	**45810625**
	0	1	**2**	3	4	5	6	7	8	9	**245810625**

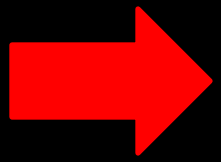
direct Criteria



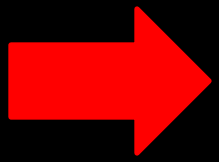
```
List users =
   sess.createCriteria(Users.class)
      .add(
           Restrictions.sqlRestriction(
                "Users.SSN = ?",
                '%' + ssn,
                Hibernate.STRING)
           )
   .list();
```

logical Criteria

```
List users =
   sess.createCriteria(Users.class)
   .add(
   Restrictions.and(
     Restrictions.eq(“ID", uid),
     Restrictions.like(“SSN", ‘%’+ssn)
     )
   )
   .list();
```



Example criteria



```
Example ex = Example.create(user)
    .ignoreCase()
    .enableLike();

List results =
    session.createCriteria(User.class)
      .add(ex)
      .list();
```

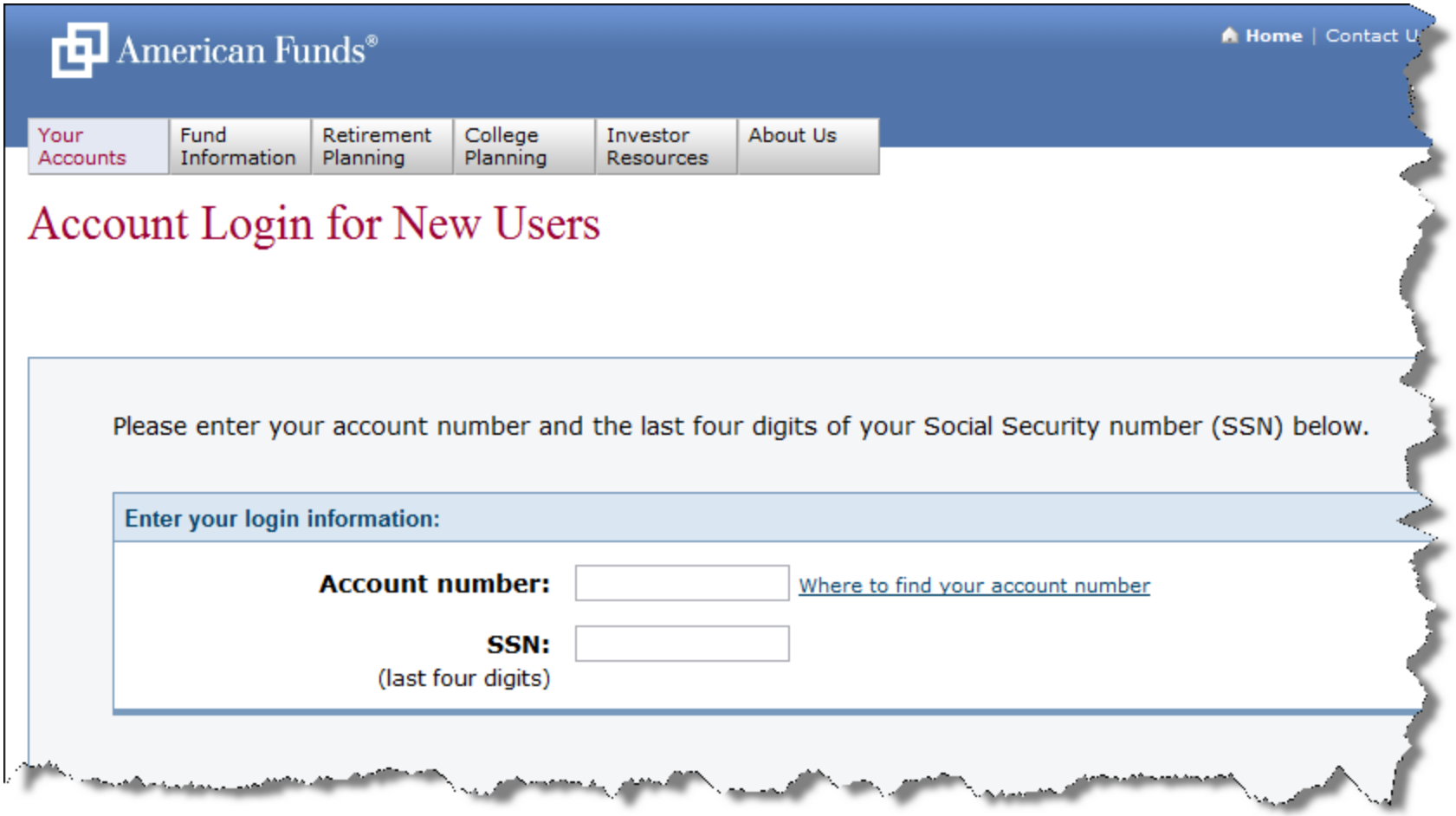
affects ASP.NET too

```
SQLi =   “SELECT *
         FROM Users
         WHERE SSN
         LIKE '%‘ + ssn"

1 sql =  “SELECT *
         FROM Users
         WHERE SSN
         LIKE '%' + @ssn“
```

2
```
cmd.Parameters.Add(
   New SQLParameter("@ssn",
     "%“ & ssn & "%"))
```

3
```
sql =    “SELECT * FROM Users
         WHERE SSN LIKE @ssn“

cmd.Parameters.Add(
   New SQLParameter("@ssn", ssn))
```

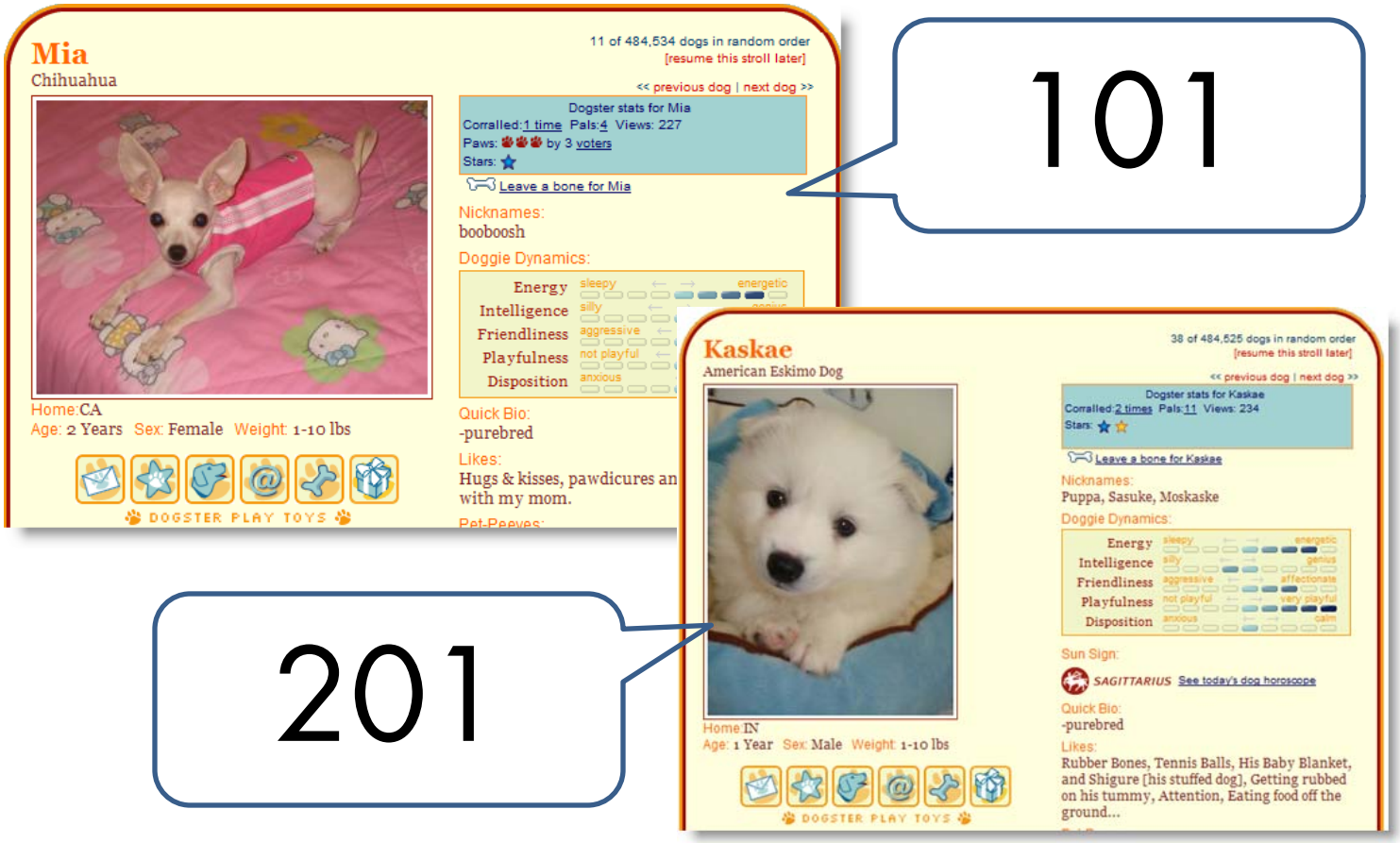
for real


American Funds®
Home | Contact U
Your Accounts
Fund Information
Retirement Planning
College Planning
Investor Resources
About Us
Account Login for New Users
Please enter your account number and the last four digits of your Social Security number (SSN) below.
Enter your login information:
Account number:
Where to find your account number
SSN:
(last four digits)

topics

- Right Tool, Right Time, Right Place
- HQL Manipulation Attacks
- Slow State Manipulation Attacks

slow state manipulation



http://dog.com/edit.jsp?dogID=101

STACH&LIU

edit check

1. `String dogID = request.getParameter(“dogID”);`

2. `request.getSession().setAttribute(“editDogID”, dogID);`

3. `if (ownsDog() != true) return;`

   `{...}`

4. `request.getSession().removeAttribute(“editDogID”);`

access denied

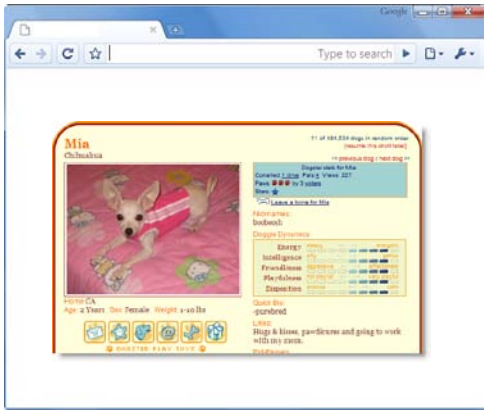
http://dog.com/edit.jsp?dogID=201

submit check

```
1  if (getDogID.equals(
2        request.getSession()
       .getAttribute("editDogID"))
   {
3        // submission allowed
   }
```

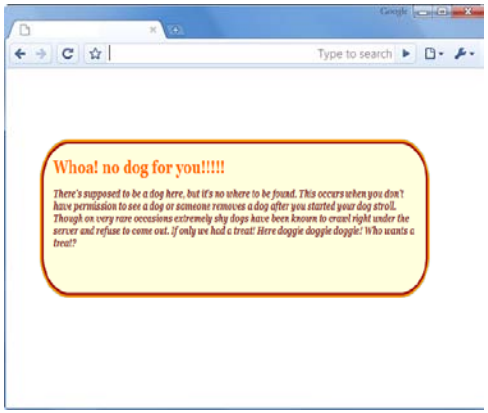
dog theft

1



dogID=101

2



dogID=201

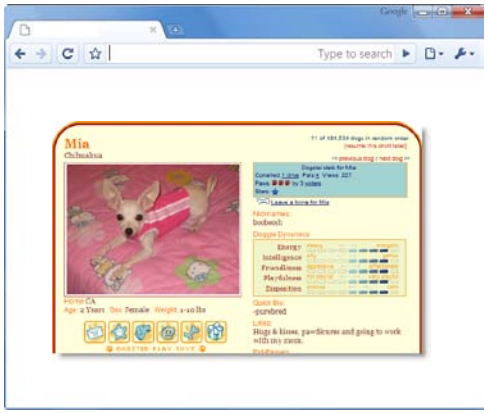
editDogID=201

edit check

```
String dogID =
   request.getParameter(“dogID”);

request.getSession()
   .setAttribute(“editDogID”, dogID);

if (ownsDog() != true) return;

{...}

request.getSession()
   .removeAttribute(“editDogID”);
```

dog theft

3



submit changes
w/dogID=201

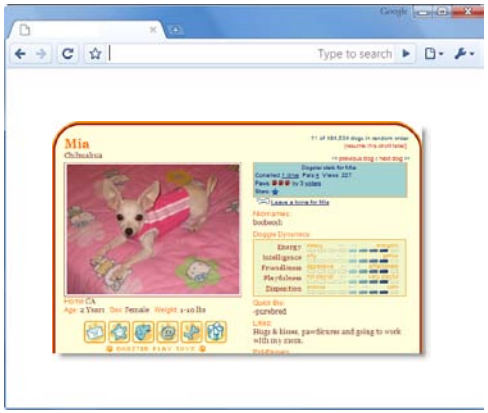
editDogID=201

submit check

```
if (getDogID.equals(
      request.getSession()
      .getAttribute("editDogID"))
{
          // submission allowed
}
```

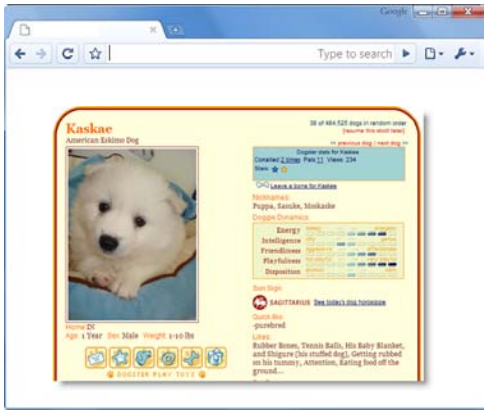
dog theft*

3



submit changes
w/dogID=201

4



dogID=201

*No Dogs Were Physically Stolen For This Presentation

summary

- Right Tool, Right Time, Right Place
- HQL Manipulation Attacks
- Slow State Manipulation Attacks

thanks!