Interactive Code Reviews

Use ‘Manual IAST’ for Effective CR

Tamir Shavro
Head of Seeker R&D, Synopsys

OWASP Israel, September 2016
• Head of Seeker R&D at Synopsys
• 18 years of XP in the Software & Security Fields
• Hands-on Pen-Tester XP
• IDF Intelligence Corps, Tech Unit
Agenda

• Background & Motivation
• Core Idea of the Solution
• Implementation Steps
• Live Demo
• Pros & Cons
• Q&A
Background & Motivation

What’s wrong with current review process?
What’s wrong with current review process?

- Many ‘Too-s’
  - Too much code, too short timeframe
  - Too many attack vectors
  - Too many entry points / pages / parameters
  - Too many new frameworks / third party components
  - Too often, too complex to follow and understand
Motivation

What if I could tell you where to look...

• Don’t spend time chasing ghosts
  HINT: no LDAP activity → LDAP Injection goes off the list
• Make new frameworks transparent by looking at the provider level
• Focus only on relevant code sections
• Order of magnitude improvement of value for $$$
Core Idea of the Solution

Empowering the Reviewer with Runtime Technology
Core Idea of the Solution

What is Runtime/Interactive Technology?

- Runtime information could be monitored easily by using existing technologies
  - Live debugging techniques at provider level
  - On-the-fly instrumentation/profiling techniques
Core Idea of the Solution

Basic Byte-Code Debugging Explained

JAVA BYTE CODE

```
1  ILOAD    // i = j + k
2  ILOAD k
3  IADD
4  ISTORE i
5  ILOAD i  // if (i < 3)
6  BIPUSH 3
7  IF_ICMPEQ 1
8  ILOAD j  // j = j - 1
9  BIPUSH 1
10  ISUB
11  ISTORE j
12  GOTO L2
13  L1: BIPUSH 0  // k = 0
14  ISTORE k
15  L2:...
```
Core Idea of the Solution

Debugging at Provider Level Explained

Tested Application

```java
public void saveCustomerOrder(CustomerOrder order) {
    String query = "Insert into orders " +
    "(order_number, users_id_fk, sales_tax, credit_card, total, bank_account) " +
    "values (?, ?, ?, ?, ?, ?);" +
    "order.getOrderByNumber() = "+
    "order.getUser().getId() = " +
    "order.getSalesTax() = " +
    "order.getCreditCardNumber() = " +
    "order.getTotal() = " +
    "order.getBankAccountNumber() = ";"

    JdbcTemplate jt = new JdbcTemplate(getDataSource());
    jt.execute(query);
}
```

Runtime Data

```sql
insert into orders
(order_number, users_id_fk, sales_tax, credit_card, total, bank_account) values (...)
```
Core Idea of the Solution

On-the-fly Instrumentation Explained

```
1  i = j + k;
2       ILOAD j  // i = j + k
3           0  0x15 0x02
4   if (i <= 3) 
5       ILOAD k
6           2  0x15 0x03
7          iADD
8           4  0x60
9     else
10        ISTORE i
11           5  0x36 0x01
12       if (i < 3) 
13        ILOAD i
14           7  0x15 0x01
15          BIPUSH 3
16           9  0x10 0x03
17     if.1
18        IF_ICMPEQ L1
19           11 0x9F 0x00 0x0D
20     end
21        ILOAD j  // j = j - 1
22           14  0x15 0x02
23          BIPUSH 1
24           16  0x10 0x01
25     l1:
26       lSUB
27           18  0x64
28       lSTORE j
29           19  0x36 0x02
30      L2:
31       lSTORE k
32           26  0x36 0x03
33      L3:
34       RETURN
35           28
```
Core Idea of the Solution

So...What’s in it for us?

HTTP Request

Runtime Data

Line of Code

```java
Statement stmt = t.createStatement();
ResultSet rs = stmt.executeQuery(SqlString);
if(rs.next()) {
    out.println("hello " + rs.getString(1));
} else {
    out.println("login failed");
}
```
Implementation Steps

Empowering the Reviewer with Runtime Technology
Step 1 - Allow Remote Debugging

- Alter JVM arguments to allow debugging (same as increasing \(-Xmx\))
- Implemented by adding one line to the startup script of the app
  
  `JAVA_OPTS="$JAVA_OPTS -agentlib:jdwp=transport=dt_socket,server=y,address=8000,suspend=n"`
Step II – Attach to Target App

• Using your favorite IDE (e.g. eclipse), create a remote connection and import breakpoints file at provider level
Step III – Use Runtime Data to Focus Your Attacks

Implementation Steps

1. Send Request
2. View Runtime Data
3. Prioritize Attack Vectors
Live Demo

Use Runtime Data During Manual PT/CR
Pros & Cons
Pros & Cons

Pros
• More value for $$$
• Makes the PT/CR more effective
• We’re not chasing ghosts anymore
• Simple to use

Cons
• Access to tested environment needed
• Need to have the app up & running
• Might not be possible when testing on production
Pros & Cons

Debugger vs. Profiler

**Debugger**
- Simple to use
- Great at identifying entry points
- Might be limited when with heavy traffic apps

**Profiler**
- Harder to fine tune to get relevant data
- A bit more complex to use
- Faster than debugger, can handle heavy traffic
Thank You!

Questions?

*Email to get BP pack: tamir.shavro at synopsys (dot) com
mail title should be: OWASP BP PACK