MILCOM 2016

SECURE COMMUNICATIONS AT THE SPEED OF CYBER

Application Auditing

Suresh C. Kothari
Richardson Professor
Department of Electrical and Computer Engineering

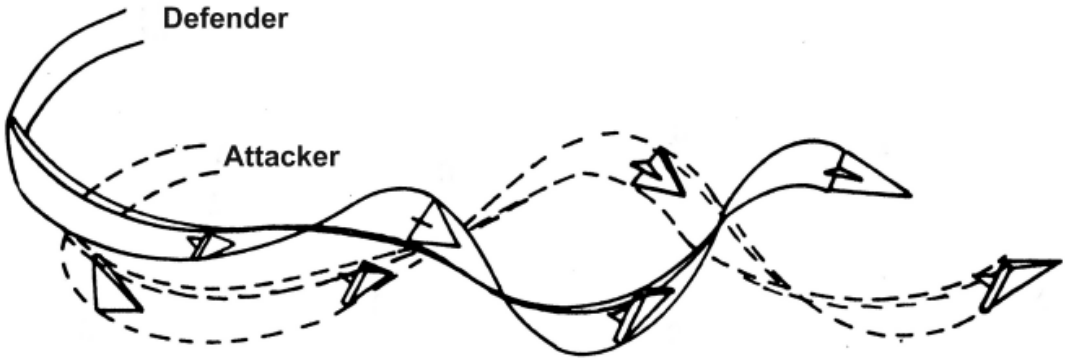
Ben Holland, Iowa State University

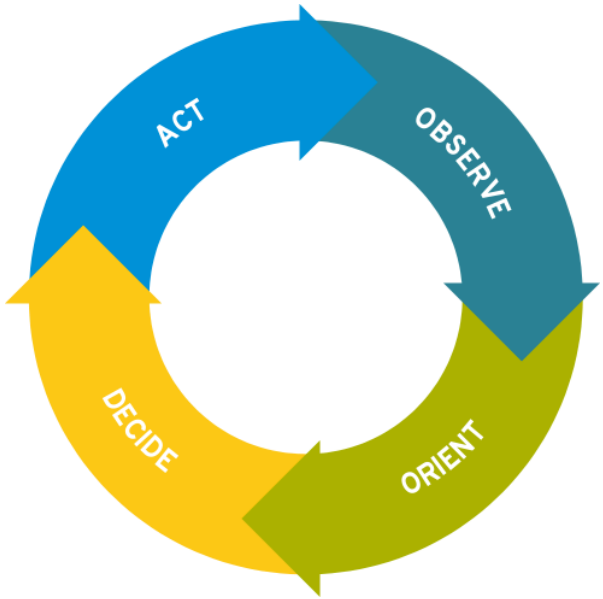
Acknowledgement: Team members at Iowa State University and EnSoft, DARPA contracts FA8750-12-2-0126 & FA8750-15-2-0080

BALTIMORE, MD • NOVEMBER 1–3, 2016

John Boyd's OODA Loop

"Security is a process, not a product" – Bruce Schneier





Our opponent

- Time!
- Our adversaries are looking too…

"…IA > AI, that is, that intelligence amplifying systems can, at any given level of available systems technology, beat AI systems. That is, a machine and a mind can beat a mind-imitating machine working by itself."
– Fred Brooks

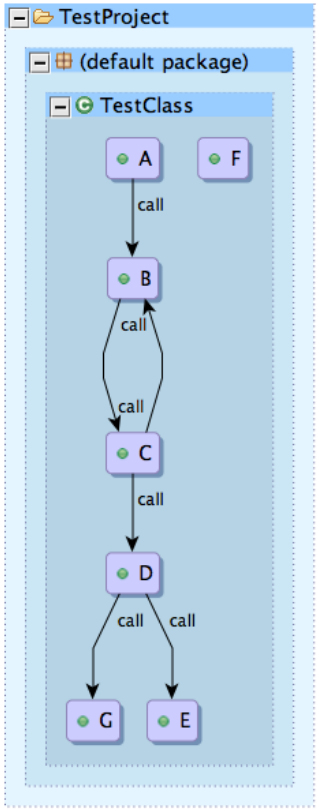
Speeding through OODA with Atlas

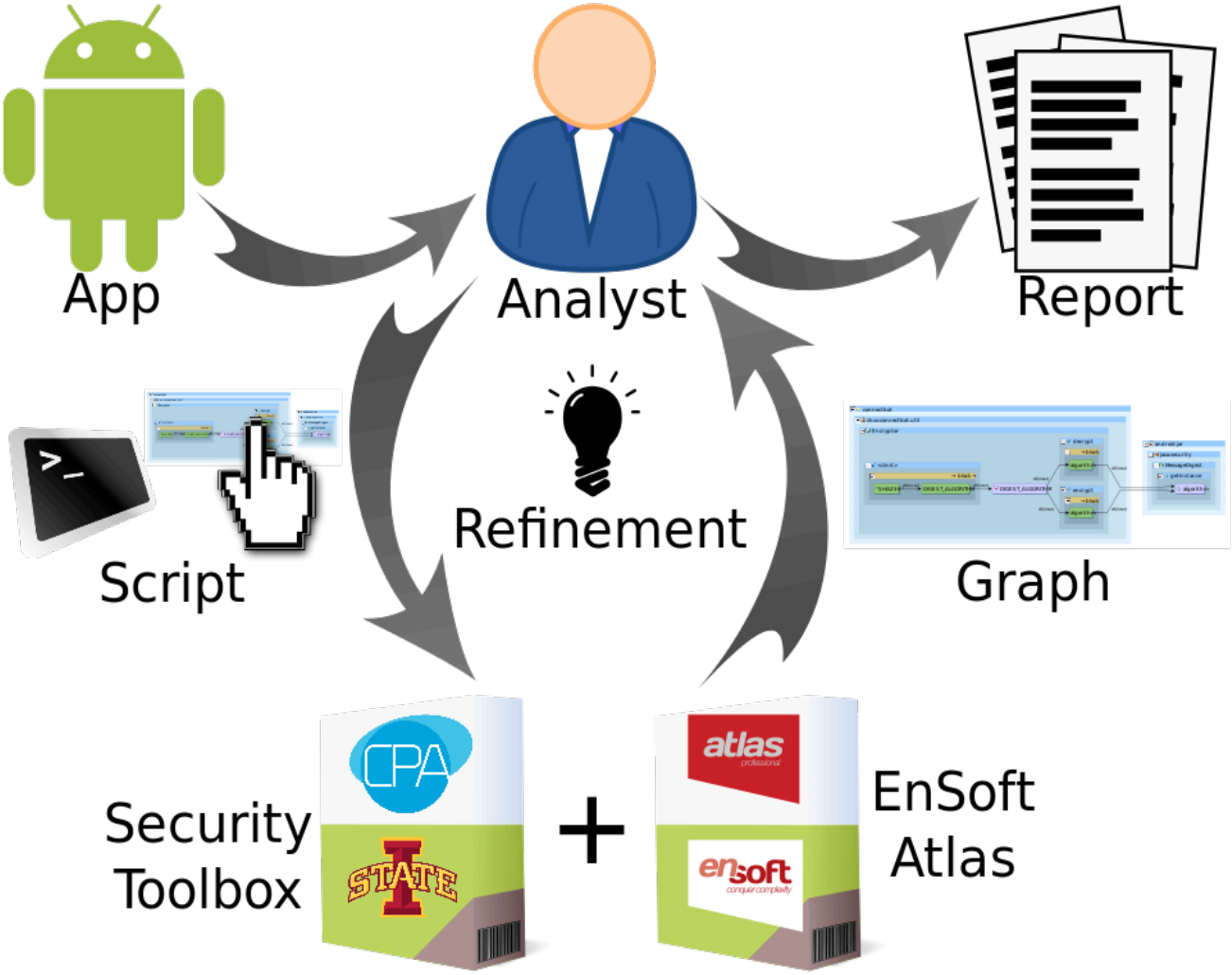
```
1  public class TestClass {
2
3      public void A() {
4          B();
5      }
6
7      public void B() {
8          C();
9      }
10
11     public void C() {
12         B();
13         D();
14     }
15
16     public void D() {
17         G();
18         E();
19     }
20
21     public void E() {
22
23     }
24
25     public void F() {
26
27     }
28
29     public void G(){
30
31     }
32
33 }
```

Program Declarations, Control Flow, and Data Flow

Queryable Graph Database

2-way Source Correspondence




App
Analyst
Report
Refinement
Script
Graph
Security Toolbox
CPA
STATE
+
atlas
ensoft
EnSoft Atlas

Lab: Auditing Web Application

- Can you find the information leakage?
- Compiled Tomcat JSP web application with SQL database
  - https://github.com/benjholla/LoginSideChannels
- Atlas WAR Decompiler Frontend
  - https://ben-holland.com/AtlasWBP/

Threat Assessment

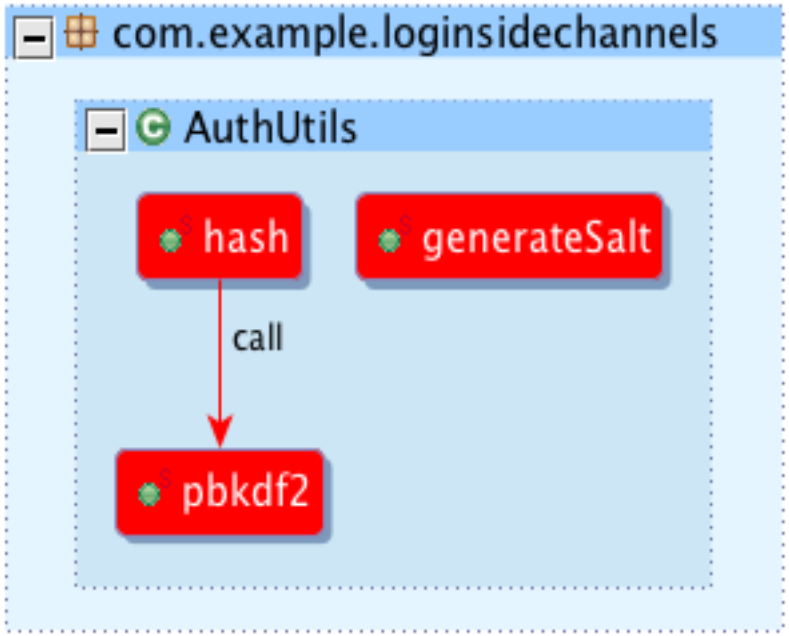
- What secrets must be protected?
- What observables does the attacker control?
- Is user enumeration a concern?
  - Consider: Ashley Madison breach

LoginSideChannels Vulnerability

- The existence of users can be inferred through timing differentials.
- More time is required to validate a password of a valid user than an invalid user.
- Attacker does not need to know any valid passwords and only has to guess at valid users.

Loop Analysis

- **Approximation:** Loops are expensive and nested loops are more expensive than non-nested loops
- **Loop Call Graph:** Recovers loops, induces call edges, highlights calls of loops called within loops.
- **Note:** Hashes are computed in a feedback loop of N rounds for improved resistance to brute force attacks.

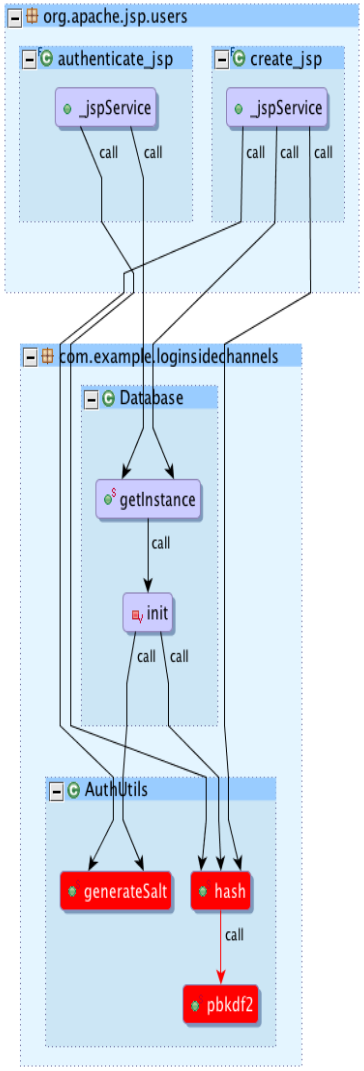


Loop Context

Question: Where are these loops used and why?

Analysis: Inspect call graph to get some context

Answer: Primarily used by two services: authenticate user and create user.

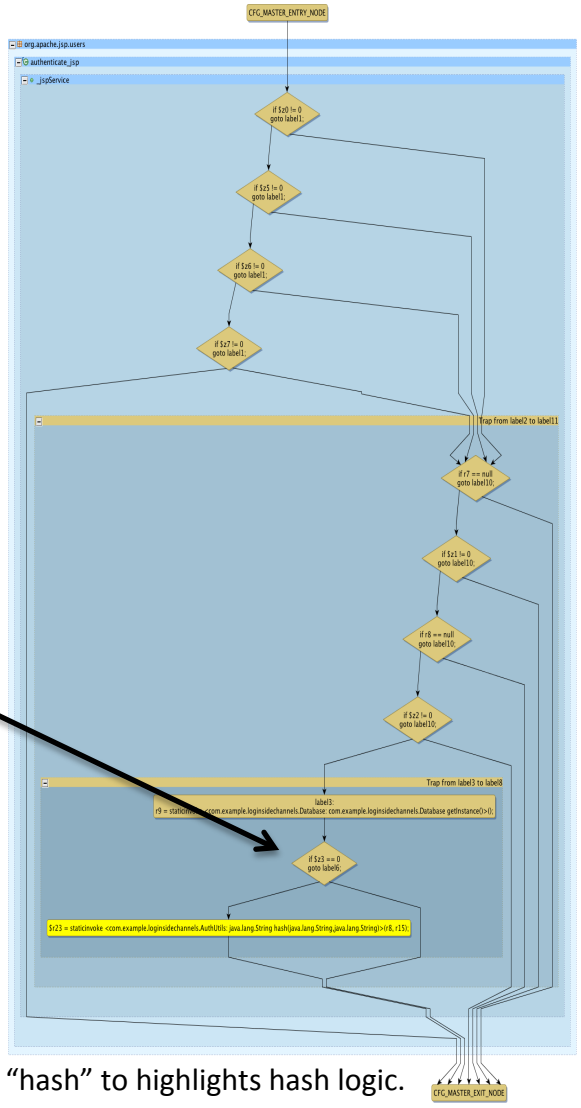


Find guard conditions

Question: Is the expensive logic used conditionally?

Analysis: Compute an Event Flow Graph (EFG, a compact graph containing only relevant conditions). Inspect “authenticate_jsp” method in a EFG.

Answer: EFG reveals a conditional guard on the hash. Analyst clicks to view code. Condition depends on result of SQL query.



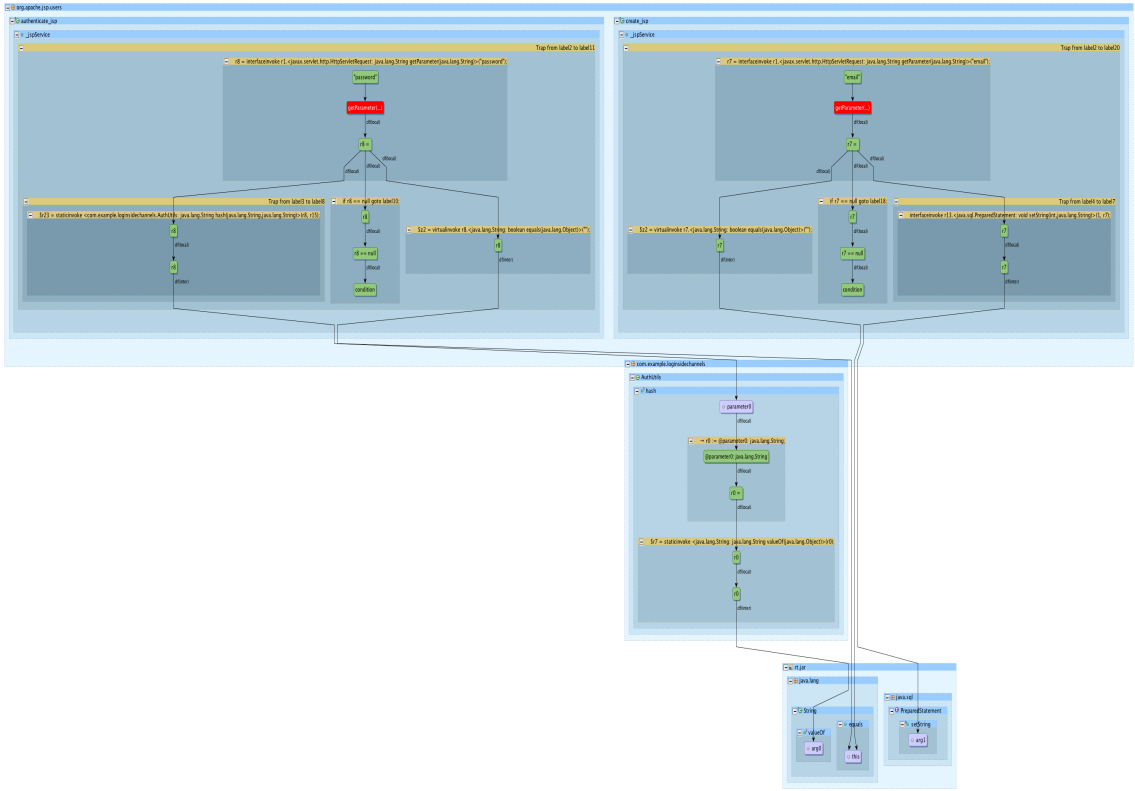
Control-F: Find “hash” to highlights hash logic.

Follow secret taint

Question: Can a secret be deduced by this potential timing difference?

Analysis: Follow data flow forward from secrets (email, password) to conditionals.

Observation: Password flows to hash; email flows into SQL.



```
r9 = staticinvoke <Database: Database getInstance()>();
r10 = virtualinvoke r9.<Database: Connection getConnection()>();
r11 = interfaceinvoke r10.<Connection: PreparedStatement prepareStatement(String)>(
      "SELECT * FROM webdb.users where Email=? LIMIT 1");
interfaceinvoke r11.<PreparedStatement: void setString(int,String)>(1, r7);
r12 = interfaceinvoke r11.<PreparedStatement: ResultSet executeQuery()>();
$z3 = interfaceinvoke r12.<ResultSet: boolean next()>();
if $z3 == 0 goto label06;
...
$r23 = staticinvoke <AuthUtils: String hash(String,String)>(r8, r15);
```

Observation: The SQL query controls the condition of interest.

Answer: Relatively expensive logic (hash) is invoked only if email exists in the database.

Demo: Proof of Concept Exploit

Video: user_enumeration_timing_attack.mp4