Static and Dynamic Analyses

CS520/620
College of Information and Computer Sciences
Hee-Tae Jung
Today’s Agenda

• Homework 1 Grading

• Examples: Apple’s ‘goto fail’ bug, Ariane 5 Failure

• Replication

• Tools: Checkstyle and JUnit in Eclipse.

• Concepts: Static and Dynamic Analyses
Homework 1 Grading

• As long as the student specified the Patient, the Therapist, the Session entities and defined some relations among them, he/she received 20 points.

• If the improper JAVA implementation was due to the improper UML design, then the student received 20 points. The student lost points only when the JAVA implementation did not represent the corresponding UML design.

• Hence, basically, most of the students started with 40 points.

• Students did not loose any points for minor errors and typos.

• Students did not loose points if the specified cardinality was larger than the necessary cardinality. Also, if the specified data structure was flexible enough to contain more than necessary values, students did not loose even if they didn't articulate the cardinality information. Some example data structures include arrays, array lists, vectors and etc.

• Students did not loose points if the specified data type had more representative power. For instance, the student was allowed to use floating points instead of ints simply because any integer values can be stored in floating point attributes.

• Student lost points when highlighted entities, attributes, operators were missing or ill-defined. They also lost the points for not articulating the corresponding data types. If the student used "Object" data type without defining Object, then the student lost points.
Homework 1 Grading

![Histogram showing the distribution of grades for Homework 1]
Homework 1 Grading

- Have any questions or want to appeal?
- Friday 2-3pm, LGRC A307
Apple’s ‘goto fail’ Bug

OSStatus err;
...
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
    goto fail;
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &hashOut)) != 0)
    goto fail;
...
err = sslRawVerify(ctx, ctx->peerPubKey, dataToSign, dataToSignLen, signature, signatureLen);

if (err) {
    sslErrorLog("SSLDecodeSignedServerKeyExchange: sslRawVerify returned %d\n", (int)err);
    goto fail;
}

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
Apple’s ‘goto fail’ Bug

OSStatus err;
...
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
    goto fail;
    goto fail;
If ((err = SSLHashSHA1.update(&hashCtx, &hashOut)) != 0)
    goto fail;
...
err = sslRawVerify(ctx, ctx->peerPubKey, dataToSign, dataToSignLen, signature, signatureLen);

if (err) {
    sslErrorLog("SSLDecodeSignedServerKeyExchange: sslRawVerify returned %d\n", (int)err);
        goto fail;
    }

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
Apple’s ‘goto fail’ Bug

OSStatus err;
...
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
goto fail;

goto fail;

If ((err = SSLHashSHA1.update(&hashCtx, &hashOut)) != 0)
goto fail;
...
err = sslRawVerify(ctx, ctx->peerPubKey, dataToSign, dataToSignLen, signature, signatureLen);

if (err) {
    sslErrorLog("SSLDecodeSignedServerKeyExchange: sslRawVerify returned %d\n", (int)err);
    goto fail;
}

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;

Fails to validate the authenticity of the connection.

What’s stored in err?
Apple’s ‘goto fail’ Bug

OSStatus err;
...
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
  goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
  goto fail;
  goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &hashOut)) != 0)
  goto fail;
...
err = sslRawVerify(ctx, ctx->peerPubKey, dataToSign, dataToSignLen, signature, signatureLen);

if (err) {
  sslErrorLog("SSLDecodeSignedServerKeyExchange: sslRawVerify returned %d\n", (int)err);

  goto fail;
}

fail:
  SSLFreeBuffer(&signedHashes);
  SSLFreeBuffer(&hashCtx);
  return err;
Apple’s ‘goto fail’ Bug
Ariane 5 Failure
Ariane 5 Failure
Ariane 5 Failure

• A code from the Ariane 4 was reused for the Ariane 5.

• A 64-bit floating point value, that was evaluated runtime, is programmed to be stored in a 16-bit signed int variable (= JAVA short data type).

• Out of range exception is triggered.

• The operation of PC was ceased and self-destruction was executed.
Ariane 5 Failure
Replication

- Eclipse Luna
- Checkstyle
- JUnit
Static Analysis

• Non-runtime environment
• Inspect program code
• Abstraction required
• Within your imagination
• Rule-based
• Conservative
• Sound (but many false positives)
Dynamic Analysis

- Run-time environment
- Valid only for test input combinations
- Work better on large complicated code
- Complete (if something appears wrong, it’s actually wrong)
## Soundness and Completeness

<table>
<thead>
<tr>
<th></th>
<th>actual error</th>
<th>actual non-error</th>
</tr>
</thead>
<tbody>
<tr>
<td>estimated error</td>
<td>true positive</td>
<td>false positive</td>
</tr>
<tr>
<td>estimated non-error</td>
<td>false negative</td>
<td>true negative</td>
</tr>
</tbody>
</table>
# Soundness and Completeness

<table>
<thead>
<tr>
<th></th>
<th>actual error</th>
<th>actual non-error</th>
</tr>
</thead>
<tbody>
<tr>
<td>estimated error</td>
<td>true positive</td>
<td>false positive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Static Analysis (sound)</td>
</tr>
<tr>
<td>estimated non-error</td>
<td>false negative</td>
<td>true negative</td>
</tr>
</tbody>
</table>
## Soundness and Completeness

<table>
<thead>
<tr>
<th></th>
<th>Actual Error</th>
<th>Actual Non-Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated Error</td>
<td>True Positive</td>
<td>False Positive</td>
</tr>
<tr>
<td>Estimated Non-Error</td>
<td>False Negative</td>
<td>True Negative</td>
</tr>
</tbody>
</table>