Side Channel Vulnerabilities on the Web - Detection and Prevention

Sebastian Schinzel
Virtual Forge GmbH
University of Mannheim
ssc@seecurity.org
Who am I?

- PHD Student at University of Mannheim (soon University of Erlangen)
  - Research topic: side-channel vulnerabilities in Web Applications

- Security Consultant at Virtual Forge GmbH
  - Expert at SAP-Software-Security
  - Co-author of “Sichere ABAP-Programmierung” at SAP-Press (http://sap-press.de/2037)
Agenda

- Background
- Side channel vulnerabilities on the Web
- Timing Side Channels
  - Detection
  - Attack
  - Prevention
- Storage Side Channels
  - Detection
  - Attack
  - Prevention
- Conclusion
Background

- Active, intrusive attacks against software systems well researched
- Vulnerabilities in real systems appear if developers don’t apply countermeasures

- Let’s assume an application with none of the top Web vulnerabilities (OWASP Top10, SANS Top25, ...)
- What can attackers still do..?
Background

- Side channel vulnerabilities allow attackers to infer potentially sensitive information just by observing normal behavior of software system
- Attacker is a passive observer

- Apply Paul Watzlawick to software applications
  - “One Cannot Not Communicate (Man kann nicht nicht kommunizieren)”
Background

Mind reading? Not as esoteric as you may think...

- Which thought do you currently think?
  1. Think about how your last pizza looked like
  2. Think about how a pink elephant with wings looks like
  3. Think about the melody of your favorite song
  4. Think about the noise of the pink elephant’s wings

- Your eyes may leak this information [6]...
Background

Mind reading? Not as esoteric as you may think...

When we can read human minds: can we also read the mind of software applications?
Agenda

- Background
- Side channel vulnerabilities on the Web
- Timing Side Channels
  - Detection
  - Attack
  - Prevention
- Storage Side Channels
  - Detection
  - Attack
  - Prevention
- Conclusion
Side channel vulnerabilities on the Web

- Learn what a user types by observing
  - reflections of monitor picture [1]
  - inter-packet timing in encrypted SSH session [2]

- Learn about the action a user performs on a Web application by observing packet sizes in encrypted Web traffic [3]
Side channel vulnerabilities on the Web

- Learn existence of user name from
  - response time of Web application [4]
  - error messages in Web page

- Timing related
  - Learn private key of SSL server [5]
  - Learn amount of hidden images in Gallery [4]
Agenda

- Background
- Side channel vulnerabilities on the Web
- Timing Side Channels
  - Detection
  - Attack
  - Prevention
- Storage Side Channels
  - Detection
  - Attack
  - Prevention
- Conclusion
Timing Side Channels

Example control flow of login form

Control flow have different length and therefore different execution time

Can we measure the time difference between control flow 1 and 2?
Timing Side Channels

Detection and Attack

Response Time (ms) vs Density

- Min: 34
- Max: 150
- Avg: 39
- Med: 37

s=0
Timing Side Channels

Detection and Attack

\[
\text{Min: } 34 \\
\text{Max: } 150 \\
\text{Avg: } 39 \\
\text{Med: } 37
\]
Timing Side Channels

Detection and Attack

- **Statistical analysis of response times difficult**
  - Highly skewed distribution, sometimes with multiple modi, depending on network conditions and measurement hardware [7]
  - Thus, parametric hypothesis tests (e.g. t-test) useless
  - Detection and attack requires custom hypothesis tests

- Detection and attack may require many thousand probes (potentially high effort)
Timing Side Channels

Preventing timing side channels (white box)

- Join control paths, e.g.
  - Pack all db queries in one SQL statement

User exists? → Yes
  → User locked? → No
    → User expired? → No
      → Password correct? → No
        → Error page
      → Yes
    → Yes
  → Yes
  → Error page

User exists AND Password correct AND User not locked AND User not expired?
Timing Side Channels

Preventing timing side channels (black box)

- Change control flow so that paths have same execution time, e.g.
  - Delay short control paths

![Flowchart]

- User exists?
  - Yes
  - User locked?
    - Yes
      - Error page
    - No
      - User expired?
        - Yes
          - Error page
        - No
          - Password correct?
            - No
              - Delay
            - Yes
              - Yes
Timing Side Channels

Mitigation: fix response time to Worst Case Execution Time (WCET)

\[ s=0 \quad s=1 \]

\[
\begin{array}{c|c|c|c}
\hline
& \text{Min:} & \text{Max:} & \text{Avg:} \\
\hline
s=0 & 34 & 150 & 39 \\
\hline
s=1 & 150 & 150 & 150 \\
\hline
\end{array}
\]

Density

Response Time (ms)

WCET
Timing Side Channels

Preventing timing side channels (black box)

- Mitigation: fix response time to worst case execution time

- **Pro:**
  - No differences in response times
  - Perfect mitigation for timing vulnerabilities

- **Con:**
  - Serious performance impact!

- More performant strategies are currently researched
Agenda

- Background
- Side channel vulnerabilities on the Web
- Timing Side Channels
  - Detection
  - Attack
  - Prevention
- Storage Side Channels
  - Detection
  - Attack
  - Prevention
- Conclusion
Storage Side Channels

Example for obvious storage side channel: Error messages of login forms
Storage Side Channels

Example for obvious storage side channel: Error messages of login forms
Storage Side Channels

Example for obvious storage side channel: Error messages of login forms

- “Invalid user name” → user name does not exist
- “Invalid password” → user name exists
Storage Side Channels

- **Hidden** storage side channel: Secret-dependent differences that are invisible to “normal user”
  - HTTP headers and values
  - HTML meta data
  - ...

Storage Side Channels

- Noise is a problem for measurements
  - lots of dynamic content in HTTP/HTML

$ diff responses/1.content responses/3.content
2c2
  < Date: Tue, 22 Jun 2010 17:20:31 GMT
  ---
  > Date: Tue, 22 Jun 2010 17:20:37 GMT
  8c8
  < Last-Modified: Tue, 22 Jun 2010 17:20:34 GMT
  ---
  > Last-Modified: Tue, 22 Jun 2010 17:20:38 GMT
122c122
  <  <input type="hidden" name="challenge" value="35018d1af7184bad10944cb617677c99" />
  ---
  >  <input type="hidden" name="challenge" value="b50cbc351f525fcad0cb0fc97e080b29" />

Time dependent difference

Randomly generated difference
Storage Side Channels

- New method to detect storage side channels (to be published S. Schinzel and F. Freiling)
  - Factor out all irrelevant differences
  - Works on binary data

```
A_1, A_2, ..., A_n
\downarrow
LCS
\downarrow
X_A
\downarrow
\Lambda
\downarrow
E

B_1, B_2, ..., B_n
\downarrow
LCS
\downarrow
X_B
```

- Step 1:
- Step 2:
- Step 3:
Storage Side Channels

- New method to detect storage side channels (to be published S. Schinzel and F. Freiling)
  - Factor out all irrelevant differences
  - Works on binary data

A1: <input type="hidden" name="challenge" value="35018d1af7184bad10944cb617677c99" />
A2: <input type="hidden" name="challenge" value="b50cfc351f525fa0db0fc97e080b29" />
A3: <input type="hidden" name="challenge" value="d4636195f85aa97be8536b762040a92" />
A4: <input type="hidden" name="challenge" value="0b33ab736e3e30b6ed194a76cf214dac" />
A5: <input type="hidden" name="challenge" value="07c16ff3fd5beb4ef5db0fcb343315b4" />
A6: <input type="hidden" name="challenge" value="62406154e66b7f85f85aa97be8536b762040a92" />
A7: <input type="hidden" name="challenge" value="cb9a053a26136efa02e8dace91ee33691" />
A8: <input type="hidden" name="challenge" value="952fd6b974853c6d51d4651ef26e2ba" />
A9: <input type="hidden" name="challenge" value="f78306c61402fd28f7d252d679e0bba" />
A10: <input type="hidden" name="challenge" value="65dd8e9169270a9f4556187d53edf81e" />

x1: "35018d1af7184bad10944cb617677c99"
x2: "501fad0c99"
x3: "1fab9"
x4: "ab9"
x5: "b"
x6: "b"
x7: "b"
x8: "b"
x9: "b"
x10: ""
Storage Side Channels

Results (1/3)

- Widely used Content Management System leaks information by HTTP header ordering
  - Does user account exist?

![Secret-independent differences and Secret-dependent difference graphs]
## Storage Side Channels

### Results (1/3)

- **Typo3 leaks information by HTTP header ordering**
  - Does user account exist?

<table>
<thead>
<tr>
<th>Non-existent user name (s=0)</th>
<th>Existing user name (s=1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HTTP/1.1 200 OK</strong></td>
<td><strong>HTTP/1.1 200 OK</strong></td>
</tr>
<tr>
<td>Date: Mon, 25 Jan 2010 11:47:55 GMT</td>
<td>Date: Mon, 25 Jan 2010 11:47:45 GMT</td>
</tr>
<tr>
<td>Server: Apache/2.2.9 (Debian) PHP/5.2.6-1+lenny4 with Suhosin-Patch</td>
<td>Server: Apache/2.2.9 (Debian) PHP/5.2.6-1+lenny4 with Suhosin-Patch</td>
</tr>
<tr>
<td>X-Powered-By: PHP/5.2.6-1+lenny4</td>
<td>X-Powered-By: PHP/5.2.6-1+lenny4</td>
</tr>
<tr>
<td>Expires: Thu, 19 Nov 1981 08:52:00 GMT</td>
<td>Expires: Thu, 19 Nov 1981 08:52:00 GMT</td>
</tr>
<tr>
<td>Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0</td>
<td>Cache-Control: no-cache, must-revalidate, post-check=0, pre-check=0</td>
</tr>
<tr>
<td>Pragma: no-cache</td>
<td>Cache-Control: no-cache, must-revalidate, post-check=0, pre-check=0</td>
</tr>
<tr>
<td>Vary: Accept-Encoding</td>
<td>Pragma: no-cache, must-revalidate, post-check=0, pre-check=0</td>
</tr>
<tr>
<td>Content-Type: text/html; charset=iso-8859-1</td>
<td>Vary: Accept-Encoding</td>
</tr>
<tr>
<td>Content-Length: 5472</td>
<td>Content-Type: text/html; charset=iso-8859-1</td>
</tr>
<tr>
<td></td>
<td>Content-Length: 5472</td>
</tr>
</tbody>
</table>
Storage Side Channels

Results (2/3)

Postfix Admin: user name exists?

```
[...]
Content-Length: 3633
[...]
<td><input class="flat" type="text" name="fUsername" value="admin@admin.de" /></td>
[...]
```
Online gallery leaks the amount of private pictures:

- Secret-independent differences
- Secret-dependent difference
Storage Side Channels

Results (3/3)

- Online gallery leaks the amount of private pictures:

  7 public images, 0 private image ($s=0$)

```html
<div style='float:left'>Pictures -
  <a href='display.php?t=bycat&amp;q=4&amp;nr=7&amp;st=0&amp;upto=12&amp;p=1'>
  <span style='color:#fff'>Other</span>
  ↑
  </a>
</div>

7 public images, 1 private image ($s=1$)

```html
<div style='float:left'>Pictures -
  <a href='display.php?t=bycat&amp;q=4&amp;nr=8&amp;st=0&amp;upto=12&amp;p=1'>
  <span style='color:#fff'>Other</span>
  ↑
  </a>
</div>
Agenda

- Background
- Side channel vulnerabilities on the Web
- Timing Side Channels
  - Detection
  - Attack
  - Prevention
- Storage Side Channels
  - Detection
  - Attack
  - Prevention
- Conclusion
Conclusion

- Side channel vulnerabilities pose a serious threat for Web applications with high security requirements
- Timing side channels may require substantial measurement and analysis effort
  - Depending on timing difference
  - Depending on network noise
- Hidden storage side channels can be found with around a dozen requests
  - Independent of the size of secret-depended changes
  - Independent of network noise
Conclusion

- Side channels can appear in various ways
  - Detection is difficult
- Side channel attacks are passive
  - Attacks are feasible for a skilled attacker
- Prevention strategies may have a negative impact on system performance
  - Prevention is difficult
Call for participation!

- Academia
  - Joint research
  - Lots of promising topics for theses (Bachelor, Master, Diploma)

- Business, Organizations
  - Applying our academic tools to real-world applications
  - Get tomorrow’s security analysis now

Get in touch!
Bibliography


[4]: Andrew Bortz and Dan Boneh, Exposing private information by timing web applications, WWW, pp. 621-628, ACM, 2007


Thank you for your attention!

Feedback, discussion?

Contact:
Sebastian Schinzel
ssc@seecurity.org