Side Channel Vulnerabilities on the Web - Detection and Prevention

Sebastian Schinzel
Virtual Forge GmbH
University of Mannheim
ssc@seecurity.org

Copyright 2007 © The OWASP Foundation Permission is granted to copy, distribute and/or modify this document under the terms of the OWASP License.
Who am I?

- 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)

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

Density

Min: 34
Max: 150
Avg: 39
Med: 37
Timing Side Channels

Detection and Attack

![Graph showing response time distributions for s=0 and s=1 with minimum, maximum, average, and median values.]

- Min: 34
- Max: 150
- Avg: 39
- 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 AND
Password correct AND
User not locked AND
User not expired?
```

```
<table>
<thead>
<tr>
<th>User exists?</th>
<th>Error page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
```

```
User locked?
```

```
<table>
<thead>
<tr>
<th>User locked?</th>
<th>Error page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
```

```
User expired?
```

```
<table>
<thead>
<tr>
<th>User expired?</th>
<th>Error page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
```

```
Password correct?
```

```
<table>
<thead>
<tr>
<th>Password correct?</th>
<th>Error page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
```
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
Timing Side Channels

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

<table>
<thead>
<tr>
<th>Density</th>
<th>Response Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>s=0</td>
</tr>
<tr>
<td></td>
<td>s=1</td>
</tr>
<tr>
<td>Min:</td>
<td>34</td>
</tr>
<tr>
<td>Max:</td>
<td>150</td>
</tr>
<tr>
<td>Avg:</td>
<td>39</td>
</tr>
<tr>
<td>Med:</td>
<td>37</td>
</tr>
</tbody>
</table>

Min: 150  Max: 150  Avg: 150  Med: 150
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

\[
\begin{align*}
\text{Step 1:} & \quad A_1, A_2, \ldots, A_n \quad B_1, B_2, \ldots, B_n \\
\text{Step 2:} & \quad \text{LCS} \quad \text{LCS} \\
\text{Step 3:} & \quad X_A \quad X_B \\
& \quad \Delta \\
& \quad E
\end{align*}
\]
Storage Side Channels

Results (1/2)

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

Non-existent user name (s=0)

HTTP/1.1 200 OK
Date: Mon, 25 Jan 2010 11:47:55 GMT
Server: Apache/2.2.9 (Debian) PHP/5.2.6-1+lenny4 with Suhosin-Patch
X-Powered-By: PHP/5.2.6-1+lenny4
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Last-Modified: Mon, 25 Jan 2010 11:47:55 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Type: text/html;charset=iso-8859-1
Content-Length: 5472

Existing user name (s=1)

HTTP/1.1 200 OK
Date: Mon, 25 Jan 2010 11:47:45 GMT
Server: Apache/2.2.9 (Debian) PHP/5.2.6-1+lenny4 with Suhosin-Patch
X-Powered-By: PHP/5.2.6-1+lenny4
Expires: 0
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Last-Modified: Mon, 25 Jan 2010 11:47:45 GMT
Vary: Accept-Encoding
Content-Type: text/html;charset=iso-8859-1
Content-Length: 5472
Storage Side Channels

Results (2/2)

Online gallery leaks the amount of private pictures:

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

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