CAPTCHA-based Code Voting

Jörg Schwenk, Rolf Oppliger, Christoph Löhr
Motivation:
Why do we need/want CAPTCHA-based Code Voting?
Motivation

Simplified internet voting protocol
Motivation

Compromised internet voting protocol
Motivation

Secure internet voting protocol

Server → Encrypted Channel → Client PC → Interaction → User

- Vote = A
- Vote =?
- B =?

Secure server ↔ user communication

CAPTCHA-based Code Voting
Code Voting:
The basic approach
How does Code Voting work?

• Every candidate is assigned to a random number or string (=voting code) and compiled to a code sheet
• These code sheets are distributed out-of-band (e.g. by mail) to the voter
• The voter casts his ballot by entering the voting code of the desired candidate

A exemplary code sheet with voting codes

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Voting code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>236412</td>
</tr>
<tr>
<td>Bob</td>
<td>990234</td>
</tr>
<tr>
<td>Carol</td>
<td>141290</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Code Voting diagram

Secure server ↔ user communication

<table>
<thead>
<tr>
<th>Choice</th>
<th>Voting code</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>487</td>
</tr>
<tr>
<td>B</td>
<td>342</td>
</tr>
</tbody>
</table>

CAPTCHA-based Code Voting
## Advanced modes of code voting

### A exemplary code sheet with voting and verification codes

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Voting code</th>
<th>verification code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>236412</td>
<td>124355</td>
</tr>
<tr>
<td>Bob</td>
<td>990234</td>
<td>672345</td>
</tr>
<tr>
<td>Carol</td>
<td>141290</td>
<td>45686</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### A exemplary code sheet with voting, verification and confirmation codes

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Voting code</th>
<th>verification code</th>
<th>confirmation code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>236412</td>
<td>124355</td>
<td>252435</td>
</tr>
<tr>
<td>Bob</td>
<td>990234</td>
<td>672345</td>
<td>574546</td>
</tr>
<tr>
<td>Carol</td>
<td>141290</td>
<td>45686</td>
<td>124145</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Drawbacks of Code Voting

• **Usability**
  The concept of code voting may become confusing for the average user. The bigger the code sheets get, the harder it is to look up.

• **Scalability**
  The more choices are offered and the more people are participating to vote
  ⇒ the more data needs to be stored
  ⇒ the more data needs to be looked up
  ⇒ the more code sheets need to be created and distributed out-of-band
What are CAPTCHAs?
What are CAPTCHAs?

Completely
Automated
Public
Turing test to tell
Computers and
Humans
Apart
What are CAPTCHAs?

**CAPTCHA: The idea**

**Picture**

**Computer** → **Picture** → **Human** → **Dog**
What are CAPTCHAs?

CAPTCHA: The implementation

Picture

Computer

Secret

Human

Secret
What are CAPTCHAs?

CAPTCHA: The implementation

Optical character recognition (OCR) enables computers to read text from images ⇒ Distort the images
What are CAPTCHAs?

CAPTCHA: The implementation

smwm →...

smwm →...

smwm → Submit
What are CAPTCHAs?

Example of CAPTCHA-Implementations
What are CAPTCHAs?

Common use of CAPTCHAs

- Spam protection
- Denial of service attacks
- Automatic account creation
- Protection against brute force password cracking attacks
CAPTCHA-based Code Voting:
How does it work?
Combining CAPTCHAs with Code Voting
The names of the candidates are not send as a string, but as a CAPTCHA images with a random name. These images are hyperlinks.

<table>
<thead>
<tr>
<th>Alice</th>
<th>Dave</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>Eve</td>
</tr>
<tr>
<td>Carol</td>
<td></td>
</tr>
</tbody>
</table>
Combining CAPTCHAs with Code Voting

The CAPTCHA images are at runtime and differ in shape and position from voter to voter.
Combining CAPTCHAs with Code Voting

Every CAPTCHA of the candidates is associated with a random string, representing the voting code.

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Voting Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>686e5854524a344f383952714e4245</td>
</tr>
<tr>
<td>Bob</td>
<td>31416c513439636f766c6948596238</td>
</tr>
<tr>
<td>Carol</td>
<td>64336a30417937554946424366696c</td>
</tr>
<tr>
<td>Dave</td>
<td>4b6e754e352e374a6f637855307145</td>
</tr>
<tr>
<td>Eve</td>
<td>4c742e4441464a506e656138715368</td>
</tr>
</tbody>
</table>
Combining CAPTCHAAs with Code Voting

Just like the CAPTCHA images, the voting codes are created at runtime differ from voter to voter, too.

<table>
<thead>
<tr>
<th>Voter A</th>
<th>Voter B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Candidate</strong></td>
<td><strong>Voting Code</strong></td>
</tr>
<tr>
<td>Alice</td>
<td>686e58545...</td>
</tr>
<tr>
<td>Bob</td>
<td>31416c513...</td>
</tr>
<tr>
<td>Carol</td>
<td>64336a304...</td>
</tr>
<tr>
<td>Dave</td>
<td>4b6e754e3...</td>
</tr>
<tr>
<td>Eve</td>
<td>4c742e444...</td>
</tr>
</tbody>
</table>
The user interface
CAPTCHA-based Code Voting

What the Client PC/Malware is able to see

<table>
<thead>
<tr>
<th>HTML code for a candidate entry</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;a href=&quot;vote.php?voted=686e5854524a344f383952714e42454a&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;img src=&quot;/tmp/9689/686e5854524a344f383952714e42454a.gif&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/a&gt;</code></td>
</tr>
</tbody>
</table>

The file name of the CAPTCHA image is a random string, thus making it impossible for the computer to conclude the candidate‘s name from it.
CAPTCHA-based Code Voting protocol

Server

Encrypted Channel

Client PC

User

Secure server ↔ user communication

Images/Voting Codes

Voting code

| 487 |
| 342 |

Choice

| A   | B   |

"Click"
Benefits of CAPTCHA-based Code Voting

• **Ease of use**
  The voter can cast the ballot by simply clicking on a hyperlink.

• **Scalability**
  CAPTCHA images and voting codes are created at runtime. Small amounts of data only need to be stored while the voting procedure is in progress. Since the voting codes can be send in-band, there is no need for out-of-band distribution.
Preliminary Security Analysis
Preliminary Security Analysis

Threads of CAPTCHA-based Voting

• Certain CAPTCHAs can be solved by computers

Several CAPTCHA implementations are considered defeated.
⇒ Use up-to-date CAPTCHA implementations
Preliminary Security Analysis

- Use humans to solve CAPTCHAs

<table>
<thead>
<tr>
<th>Choice</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Voting code</th>
</tr>
</thead>
<tbody>
<tr>
<td>487</td>
</tr>
<tr>
<td>342</td>
</tr>
</tbody>
</table>

Malware

Images/Voting Codes

Secure server ↔ user communication

Images

Text
Using humans to solve CAPTCHAs

• **Pay people to solve CAPTCHAs**
  Low wages in certain countries makes it rather inexpensive for the attacker to hire people to let them solve CAPTCHAs.

• **Set up a free CAPTCHA provider**
  Websites, such as weblogs or forums, often implement CAPTCHAs from free CAPTCHA providers. An attacker might act as such provider and distribute the CAPTCHAs to unsuspicous visitors of these websites.

• **Set up a website to let the visitors solve the CAPTCHAs**
Cracking Google captchas with porn

Apparently Spammers are re-using an old but nonetheless efficient trick to defeat Google captchas. By offering free porn, they try to find humans who do the work for them and enter the distorted characters that are needed to fill out a Google registration request.

Request to enter the characters

Cracking captchas (Completely Automated Public Turing Test to Tell Computers and Humans Apart) is becoming increasingly important for spammers. The only thing standing between them and lucrative, free email accounts for example at Google is the fact, that their automated scripts are heaving a hard time to reliably decipher the images with distorted characters. As Thorsten Holz of Germany’s Honeynet Project now reports, spammers are now resorting to a method discussed four years ago – but one that has apparently not become less efficient since. On websites, they present porn images with captchas used at Google and ask users to enter the text to get the next image.

If you do so, you indeed are presented with a new image and another captcha. In the background, the spammers apparently simultaneously try to create a new Google account with the string you entered. So if you do not enter the captcha properly, you actually get an error message. Unlike the trojan from last year that awarded users with a strip show when they recognized a Yahoo captcha properly, this method does not install any software on the user's computer; it is a simple web application.

Google Mail is currently turning out to be a constant source of spam; in the past few months, the share of spam sent from Google's servers has almost quadrupled. And such tricks could indeed play a role in this increase. In March, there were reports of an automated method that detected Google captchas 20 to 30 percent of the time, though no details have been reported since.

Source: http://www.heise.de/english/newsticker/news/113336
Preliminary Security Analysis

Random voting

Secure server ↔ user communication

Client PC

Images/Voting Codes

342

Malware

"Click"

User

342=B

487=?

Voting code

B=

487

342

Choice

A

B

Images

CAPTCHA-based Code Voting
Conclusion
Conclusion

• CAPTCHA-based Code Voting is easy to use
• has a good scalability
• Protection against Malware voting fraud
• Security depends on the premise that computers cannot read the CAPTCHAs
• CAPTCHAs may be solved by humans
• Malware is able to cast random votes
Thank you!

Questions?