Securing ProtonMail: Building a Web App that Doesn’t Trust the Server

Daniel Huigens
What do we want to achieve?

- Allow you to trust that we can't read your email
- Without trusting the server
# How does our web app work?

<table>
<thead>
<tr>
<th>Normal web app</th>
<th>Our web app</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trust source code coming from the server</strong></td>
<td>?</td>
</tr>
<tr>
<td><strong>Send password to the server</strong></td>
<td>Use <strong>Secure Remote Password</strong> protocol</td>
</tr>
<tr>
<td><strong>Trust data coming from the server</strong></td>
<td>?</td>
</tr>
<tr>
<td><strong>Send data to the server unencrypted</strong></td>
<td>Send data to the server signed and encrypted using OpenPGP</td>
</tr>
</tbody>
</table>
The JavaScript trust problem (I)

- HTML, CSS and JavaScript are sent to the browser each time
- The browser does what the server says
- Server says: send me the password
The JavaScript trust problem (II)

- Could be hacked or rogue:
  - Employee
  - Hosting
  - Content Delivery Network (if used)
  - National Security Agencies
  - Corporate Network
OzCoin Hacked, Stolen Funds Seized and Returned by StrongCoin

Apr 24, 2013  4:55 PM  by Vitalik Buterin
“the funds were intercepted when the user made a payment”

“how did this happen?”
Source Code Transparency

- Hash the code at the source
- Publish it somewhere
- Verify that everyone gets the same code
Certificate Transparency

• Append-only log server
• Gives you Signed Certificate Timestamp
• Promises to publish the Certificate in the Log
Service Workers

• Sit “between web app and server”
• Can read and block responses
• Can even detect updates to the Service Worker itself
All together now

- Certificate goes in the Log Server
- Able to verify that there's only one certificate
- Hash goes in the certificate
- $\Rightarrow$ Everyone sees the same code
## How will our web app work?

<table>
<thead>
<tr>
<th>Normal web app</th>
<th>Our web app</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trust source code coming from the server</strong></td>
<td><strong>Verify source code coming from the server</strong></td>
</tr>
<tr>
<td><strong>Send password to the server</strong></td>
<td><strong>Use Secure Remote Password protocol</strong></td>
</tr>
<tr>
<td><strong>Trust data coming from the server</strong></td>
<td>?</td>
</tr>
<tr>
<td><strong>Send data to the server unencrypted</strong></td>
<td><strong>Send data to the server signed and encrypted using OpenPGP</strong></td>
</tr>
</tbody>
</table>
Key distribution solutions

• In-person exchange / verification
• Key Signing parties
• Web of Trust
Key Transparency

- Publish all keys
- Make sure that everyone sees the same keys
- Everyone checks their own key
- ⇒ All keys can be trusted
Merkle tree

Root Node
Hash(Node 0 + Node 1)

Node 0
Hash(0-0 + 0-1)

Node 1
Hash(1-0 + 1-1)

256 steps

Node 0-0-...-0
Hash(Empty Node)

Node 0-0-...-1
Hash(Fingerprint)

Node 1-1-...-0
Hash(Empty Node)

Node 1-1-...-1
Hash(Fingerprint)

[0-0-...-1, proof] == VerifiableRandomFunction(EmailAddress)
# How will our web app work?

<table>
<thead>
<tr>
<th>Normal web app</th>
<th>Our web app</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trust source code coming from the server</strong></td>
<td><strong>Verify source code coming from the server</strong></td>
</tr>
<tr>
<td><strong>Send password to the server</strong></td>
<td>**Use <strong>Secure Remote Password</strong> protocol</td>
</tr>
<tr>
<td><strong>Trust data coming from the server</strong></td>
<td><strong>Verify data coming from the server</strong></td>
</tr>
<tr>
<td><strong>Send data to the server unencrypted</strong></td>
<td><strong>Send data to the server signed and encrypted using OpenPGP</strong></td>
</tr>
</tbody>
</table>
Thanks! Questions?

Contact Us!

Daniel Huigens
Cryptography Engineer
d.huigens@protonmail.com
PGP Key ID: F7D8FA8EC9D526EC

protonmail.com
news.ycombinator.com/user?id=protonmail
reddit.com/r/ProtonMail