Digital Certificates

The issue:

A solution:

The contents of X.509, the most widely used digital certificate in the internet include:

- Serial Number
- Name of the certification authority
- Digital Signature of the issuer
- Valid-From/ Valid-To: time frame when the certificate is valid.
- Subject: person or name of the organization
- Key-usage: purpose of the public key (e.g. encipherment, signature, etc.)
- Public key
- Thumbprint Algorithm: the hash function used to hash the public key (e.g. SHA-256)
- Thumbprint: the hash itself

Typical usage:
Passwords

The use of passwords is the most common way a computer system authenticates the identity of a user. In general, here’s how it works:

Question: Why is the system not storing the password in plaintext? And why use a cryptographic hash function to store it?

Dictionary Attacks

Unfortunately, the “one-way” capability of cryptographic hash functions can be compromised because users tend to use passwords that are not only easy to remember but also easy to guess.

Suppose a system administrator insists that all passwords be 8 characters long. In the typical American keyboard, there are 94 different characters. How many different passwords are there?

If a computer can check the validity of one password in at least one microsecond, it will take at least 95 years on average to find the right password. But many users do not use the full range of characters that are available in a keyboard. In December 2009, a major password breach occurred at www.rockyou.com and 32 million passwords were released. The hacker then posted the passwords on the web but did not include any additional information. The Imperva Application Defense Center analyzed the passwords. Here are the top 20 most popular passwords:
<table>
<thead>
<tr>
<th>Rank</th>
<th>Password</th>
<th>Number of Users</th>
<th>Rank</th>
<th>Password</th>
<th>Number of Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>123456</td>
<td>290731</td>
<td>11</td>
<td>Nicole</td>
<td>17168</td>
</tr>
<tr>
<td>2</td>
<td>12345</td>
<td>79078</td>
<td>12</td>
<td>Daniel</td>
<td>16409</td>
</tr>
<tr>
<td>3</td>
<td>123456789</td>
<td>76790</td>
<td>13</td>
<td>babygirl</td>
<td>16094</td>
</tr>
<tr>
<td>4</td>
<td>Password</td>
<td>61958</td>
<td>14</td>
<td>monkey</td>
<td>15294</td>
</tr>
<tr>
<td>5</td>
<td>iloveyou</td>
<td>51622</td>
<td>15</td>
<td>Jessica</td>
<td>15162</td>
</tr>
<tr>
<td>6</td>
<td>princess</td>
<td>35231</td>
<td>16</td>
<td>Lovely</td>
<td>14950</td>
</tr>
<tr>
<td>7</td>
<td>rockyou</td>
<td>22588</td>
<td>17</td>
<td>michael</td>
<td>14898</td>
</tr>
<tr>
<td>8</td>
<td>1234567</td>
<td>21726</td>
<td>18</td>
<td>Ashley</td>
<td>14329</td>
</tr>
<tr>
<td>9</td>
<td>12345678</td>
<td>20553</td>
<td>19</td>
<td>654321</td>
<td>13984</td>
</tr>
<tr>
<td>10</td>
<td>abc123</td>
<td>17542</td>
<td>20</td>
<td>Qwerty</td>
<td>13856</td>
</tr>
</tbody>
</table>

Over the years, hackers and non-hackers alike have compiled *dictionaries* of the most common passwords. Their size is significantly smaller than $6 \times 10^{15}$. There are two ways hackers employ these kinds of dictionaries to guess passwords.

1. The “direct” way

2. The “indirect” way (See Demo)
Countermeasures: The goal here is to slow down the dictionary attachk.

1. Against direct guessing

2. Against indirect guessing
Access Control Models

One of the best ways to defend again an attack is to prevent them in the first place. Here’s a principle that is useful to follow:

*The Principle of Least Privilege:*

Different Access Control Models:

1. Access Control Matrix

```
<table>
<thead>
<tr>
<th></th>
<th>/etc/passwd</th>
<th>/usr/bin/</th>
<th>/u/roberto/</th>
<th>/admin/</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>read, write</td>
<td>read, write, exec</td>
<td>read, write, exec</td>
<td>read, write, exec</td>
</tr>
<tr>
<td>mike</td>
<td>read</td>
<td>read, exec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>roberto</td>
<td>read</td>
<td>read, exec</td>
<td>read, write, exec</td>
<td></td>
</tr>
<tr>
<td>backup</td>
<td>read</td>
<td>read, exec</td>
<td>read, exec</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
```

*Advantages:*

*Disadvantages:*

2. Access Control Lists (ACLs)

*Advantage:*

*Disadvantage:*
3. Capabilities

*Advantage:*

*Disadvantage:*

4. Role-based Access Control

*Advantage:*

*Disadvantage:*