

Detecting States of Authentication With Protected Images ha.ckers.org web application security lab

Detecting States of Authentication With Protected Images ha.ckers.org web application security lab - Author not stated, ha.ckers.org.

- Published: date not stated
- Original: <<http://ha.ckers.org/blog/20061108/detecting-states-of-authentication-with-protected-images/>>
- Preserved from: <http://ha.ckers.org/blog/20061108/detecting-states-of-authentication-with-protected-images/> (stored) on 2026-08-16
- Capture timestamp: 20090625161358
- Licence: unknown

Rights remain with the original author and publisher. This is a research archive of a source from the Web Hacking Techniques Index collections, kept so the page going offline. To read the original, follow the link above.

Content

UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

Detecting States of Authentication With Protected Images ha.ckers.org web application security lab

[[image: image]](<http://ha.ckers.org/blog/20071014/web-application-scanning-depth-statistics/>)

Paid Advertising [[image: web application security lab]](<http://ha.ckers.org/>)

Detecting States of Authentication With Protected Images

[Jeremiah Grossman](#) and I got to talking today and he reminded me of an old conversation we had had months ago around a way to detect the state of a user who is authenticated on a site. At the time it felt very academic and I didn't really feel like following through with it, but certain events have made me realize this is slightly more prevalent than either of us had originally thought. You can use files on sites to detect the state of a user.

The sample code is simple enough:

```
<IMG SRC="http://somesite.com/members/protected.jpg" onerror="alert('not authenticated')">
```

Let's assume you have an image that's inside the members directory as seen above. If the user is authenticated they can see the photo, if not, they can't and are redirected to a page where they must authenticate. If that's the case you can automatically detect if the user is logged in. The same holds true if the image changes to say something like "Hello, Bob!" once the user logs in. You can detect the size and use that to verify that the user is logged in.

You can take it further by looking for scripts that are hidden behind protected directories. Admittedly I've never seen anything like that, except in basic auth situations but I'm sure there are examples out there. But here's where the story ends. Neither Jeremiah or I could think of anything off the tops of our heads that would allow this technique to be more prevalent. Ideas?

This entry was posted on Wednesday, November 8th, 2006 at 3:30 pm and is filed under [Webapps ec](#). You can leave a response as well.

Respond here or Discuss [On the Forums](#)

Your Name *

E-Mail (will be hidden) *

URL

Archived from <http://ha.ckers.org/blog/20061108/detecting-states-of-authentication-with-protected-images/>. Rendered offline from the local Markdown copy.