# ha.ckers.org web application security lab

**ha.ckers.org web application security lab** - Author not stated, ha.ckers.org.

- Published: date not stated
- Original: <http://ha.ckers.org/blog/20070421/noisy-decloaking-methods/>
- Preserved from: http://ha.ckers.org/blog/20070421/noisy-decloaking-methods/ (stored) on 2026-08-17
- Capture timestamp: 20080112152834
- Licence: unknown



## 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.





## Noisy Decloaking Methods

Yesterday while I was helping Jeremiah with he forced basic auth cookie testing he asked a good question, which is how you can better de-anonymize users through alternative methods. Some of the initial thoughts he had wouldn't work, but the first thing that popped into my head was FTP and Gopher. Using out of bound methods to make TCP or UDP connections to a monitoring site are easy ways to correlate users (compared with time).

Now, having had a day to think about it, there are a ton of ways to do this exact same thing. Here are just a few ways I was thinking of:

**ftp://** FTP connections (port 21) - quietly connects to the port of the remote host. Many proxies don't forward FTP, meaning it will connect directly from the client to the server, bypassing any FTP servers.

**gopher://** Gopher connections (port 70) - quietly connects to the gopher port. Could be popped up in an iframe or anything similar.

**telnet://** Telnet connections (port 23) noisily opens the assigned telnet client. If you haven't already done this once, and authorized the application in Firefox it will warn you upfront about what is about to happen.

**file:///** Windows networking microsoft-ds and netbios-ssn (ports 445 and 139) although this can kind of grind your browser to a halt until it fails, it really can help identify the computer. In IE it will also cause a popup alert if it doesn't connect.

**scp://** WinSCP protocol (port 22) if WinSCP is installed the remote web server can ask you to connect to it. It will open the external application in a very obvious way.

The next question people are going to ask is, how do you do this if you have dozens of people hitting it at relatively the same time. This part of the technique is borrowed from a page out of HD Moore's decloak and some of Martin Johns' stuff. If you create a unique hostname per request, you can correlate that information back to the timestamp. However, because you aren't necessarily aware of the host (just the IP) this technique has to be modified slightly. Instead of hostname tricks you can use a number if IPs. Of course that means needing a lot of IP space. Setting up a packet sniffer in front of or on the host means you don't even have to keep any of those ports open.

Of course other protocols may be in place with external applications that are installed (similar to the scp example). Knowing them can be tricky and noisy if they fail, depending on how the fail. The point being there are a lot of alternative paths to getting a machine to connect directly to the machine in question by bypassing the normal hypertext transfer protocol completely.

This entry was posted on Saturday, April 21st, 2007 at 10:22 am and is filed under Webappsec. You can follow any responses to this entry through the RSS 2.0 feed. You can leave a response, or track back from your own site.

## Leave a Reply Or Discuss On the Forums

Name (required)

Mail (will not be published) (required)

Website

---