Papers on Web-based Fraud and Identity Theft

Kevin Kane
kane@cs.utexas.edu

Design and Analysis of Secure Protocols
Fall 2004

"Web Spoofing: An Internet Con Game"

Edward W. Felten, Dirk Balfanz, Drew Dean, and Dan S. Wallach

- Let's be clear on the context: paper written in February 1997
- **Main point:** Given an attacker-created "shadow copy" of the World Wide Web, an attacker can:
  - monitor a user's activities including passwords and account numbers
  - Send false or misleading data in the victim's name
- *The attacker does not really copy the whole web, but interposes himself between the victim and the Web*

Spoofing Attacks

- Create a “misleading context,” which tricks the victim into making an inappropriate security-relevant decision
  - The decision would be appropriate, if the context were really what it claimed to be
  - *Example*: Bogus automated teller machines capture card numbers and inputted PINs for an attacker, then pretend to experience a fault so as to appear just as a malfunctioning machine

What is a "context" on the Web?

- Visual cues about a page's origin
  - Site name from URL in the Location bar to deduce source
  - File name from URL to deduce file type or function
  - Text and images on the page, such as a logo
  - Unique appearance of the page, such as an unconventional color scheme indicative of one particular source
  - *Temporal locality*: events occurring close in time are usually related
- *All of these cues can be unreliable!*

Web Spoofing

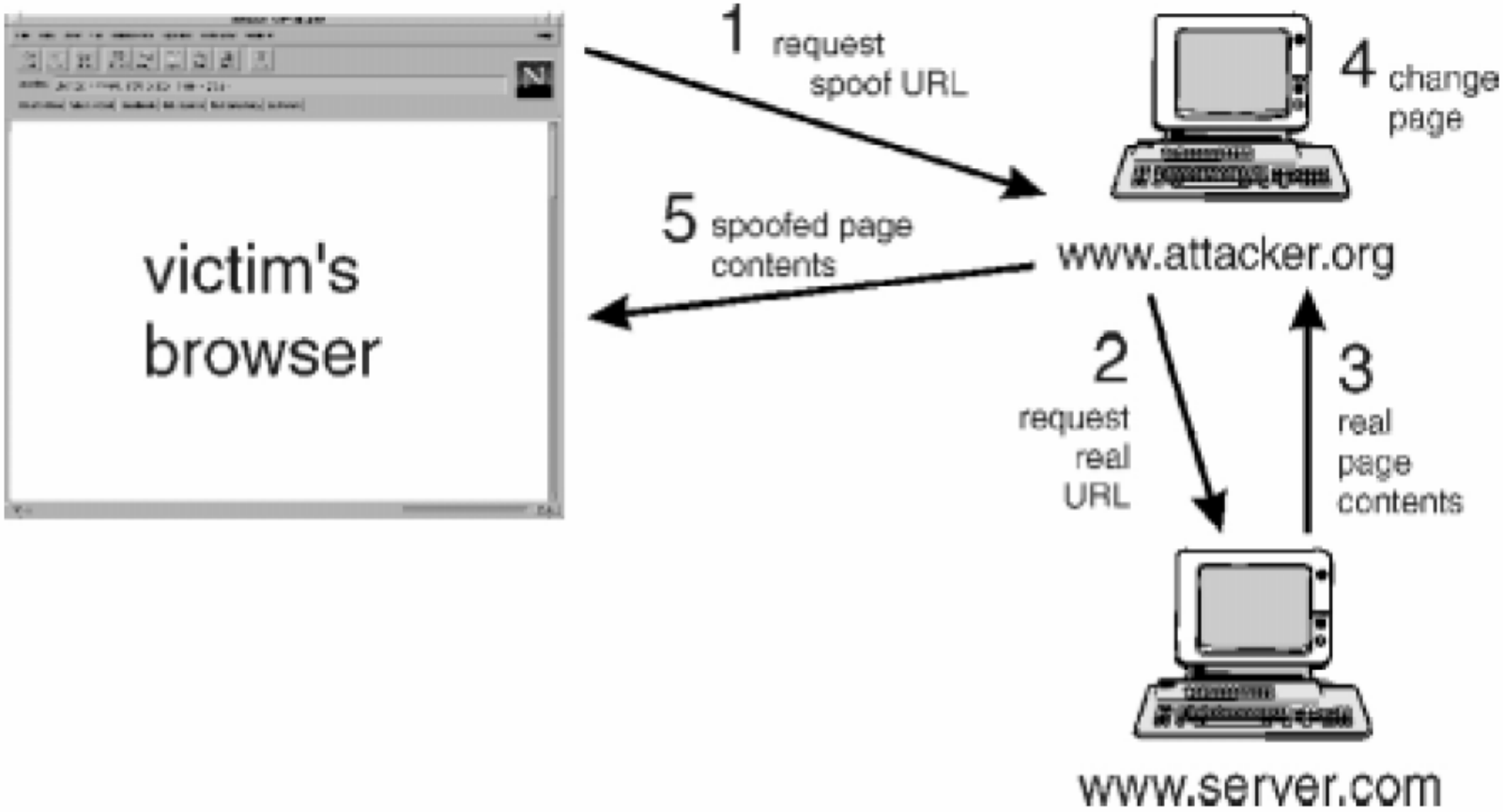
- The attacker controls the false web, and so can conduct:
  - **Surveillance**. Pages viewed and form data sent are intercepted by the attacker.
  - **Tampering**. Attacker does not have to relay victim's requests and responses correctly: data can be falsified for modified in-transit from the victim, or from the responding web server on the "real" web.

Spoofing the Whole Web

The Man in the Middle Attack

- URL rewriting
  - Prepend all URL's with the attacker's host so that requests are routed through it
    - http://home.netscape.com/ becomes http://www.attacker.org/http://www.server.com/
  - Pages are then requested through www.attacker.org, which functions as a proxy to fetch the true page (in this case, http://www.server.com/), applying any of the attacker's desired transformations in the process

Spoofing the Whole Web



“Secure” connections don’t help

- A “secure” connection in the false Web is “secure” in the sense browsers mean: there is a secure connection between the victim and the attacker’s host
  - The attacker can then create a secure connection to the real host, decrypt the received data, apply transformations, re-encrypt for the victim, and send it on

Starting and Completing the Illusion

- To start, an attacker must lure a victim into the false web, perhaps through a bogus link
- To complete the illusion, the following contextual cues provided by the browser must be falsified or hidden:
  - Status bar text
  - Location line URL
  - Viewing of document source
  - Viewing of document information
- These can be done with JavaScript

Suggested Remedies

- Common Sense
  - Disable JavaScript
  - Make sure browser location line is always visible, and pay attention to what it claims
- My remarks: What gives away an attacker site?
  - This example uses obvious names, but when sites “subcontract” to third parties, how do you tell the difference?

My remarks on shortcomings of this paper

- Attack on secure web sites is oversimplified
  - Overlooks the necessity of a server-side certificate signed by a trusted authority
    - Certificates signed by unknown authorities generate a browser warning – but do users pay attention?
- Necessity of user clicking on an attacker's link seems like a very narrow window of vulnerability
- Illusion is spoiled as soon as user manually types in a URL manually or clicks on a bookmark, which are the most likely sources for links leading to "sensitive" sites where account numbers and PINs are used

My remarks on threats not mentioned

- Malicious proxy servers configured into browser
- Malware: viruses, spyware
- Exploitation of operating system and browser vulnerabilities through web pages or attached components

Conclusions

- The appearance of a web page can be duplicated and subtly compromised
- The implied “security” of a connection only applies to the network link between a victim and the site specified in the URL bar
- The solution is common sense: Be vigilant of links for “sensitive” sites, pay attention to the Location bar

"Client-side defense against web-based identity theft" Neil Chou, Robert Ledesma, Yuka Teraguchi, Dan Boneh, and John C. Mitchell

- Web spoofing attacks now start in email: Falsified URLs are now presented in authentic looking e-mails from sites such as E*Trade or other banking sites
- "SpoofGuard" is a browser plug-in that performs a number of heuristic checks to determine a page's validity

Properties of recent attacks

- *Logos*: spoof site uses legitimate site's logos
- *Suspicious URLs*:
  - Site name has nothing to do with the honest site
  - URL is meant to appear the same on a quick glance (interchanging capital I "eye", numeric 1 "one", and lowercase l "ell", or numeric 0 "zero" and capital O "oh"),
  - URL uses IP address
  - URL uses @ "at" mark to include true site name in the URL to make it appear legitimate to user, but is used as a login/password combination by the browser (truesite.com:xxx@attackersite.com),

Properties of recent attacks

- *User input*: User is solicited for sensitive data
- *Short lived*: Spoofed site is available only long enough for attacker to spoof a large enough number of users and shut down to avoid later detection
- *Copies*: Legitimate site is copied and used with minimal changes
- *Sloppiness*: Poor spelling, grammar, and inconsistencies
- *HTTPS uncommon*: Avoids the problem of acquiring a legitimate server-side certificate

Solutions to evaluate page legitimacy

- Stateless methods that determine whether a downloaded page is suspicious
- Stateful method that take into account previous user activity
- Posted form data examination

Test scoring

- Let *TSS* be the total spoof score, scoring both individual tests, and groups of tests.
- Tests $T_1,\ldots,T_n$ are plug-in tests, each which produces a result P_i in [0,1].

$$\begin{aligned} TSS(\mathrm{page}) &= \sum_{i=1}^{n} w_i P_i \\ &+ \sum_{i,j=1}^{n} w_{i,j} P_i P_j \\ &+ \sum_{i,j,k=1}^{n} w_{i,j,k} P_i P_j P_k \\ &+ \ldots \end{aligned}$$

Stateless page evaluation

- *URL check*: Check for techniques such as the @ symbol used in deceptive URLs
- *Image check*: Maintain a database of popular e-commerce sites, and note when one of these images appears on an unaffiliated page
- *Link check*: Same tests for URL check, done to the links in the page
- *Password check*: Does the page request a password? If so, check for use of HTTPS.

Stateful page evaluation

- *Domain check*: Does the name resemble (in a Hamming distance way) but not match exactly a previously visited one? This test is admittedly crude.
- *Referring page*: When a link is clicked, are the linking and linked sites related?
- *Image-domain association*: Like the stateless image database, this does the same check with a locally assembled image database from legitimate sites

Posted form data evaluation

- *Outgoing password check*: (Domain, user name, password) triples stored, and new uses of an already used password are flagged. Passwords are stored hashed by SHA-1 to prevent information leakage.
- *Interaction with image check*: Does the page requesting information contain an image from the database that does not belong to the site?

Posted form data evaluation

- *Check of all post data*: All data is compared against stored passwords, in case the form requests a password in a non-standard way
- *Exception for search engines*: Posted data sent to known search engines is ignored

SpoofGuard

- Exists in browser memory context as a COM component for Internet Explorer
- Appears as a toolbar with visible alert
- Configuration window allows tweaking of test weights
- *(Internal structure and hooks into Internet Explorer details have been skipped)*

Evaluation

- Fourteen spoof pages of eBay's sign-in page
  - Nine are spoofs of the login page
  - Two purport to be "identity and billing verification" pages that require large amounts of personal info
  - One claims to be a "random maintenance" page
  - The last two claim the user could win a car if they provide the login data

Evaluation

- SpoofGuard noted that all pages require passwords, but were not secured with https
- SpoofGuard noted that the eBay image was present on pages not actually associated with eBay
- SpoofGuard recognized the repetition of username/passwords used for a legitimate site

Evaluation

- *False alarm rate* depends on how frequently new accounts are established, and how often history cache is cleared
- Tests are not foolproof: A clever attacker could split a password entry into separate fields, or modify an image in a way not likely to be noticed by the user, but enough to alter the image hash

Server-side assistance

- Although this solution is client-side, some assistance from the honest server could increase accuracy
  - HTML attribute to designate confidential fields
  - Images tagged on pages in which they appear to designate them as only to appear on their site
  - Site-specific salting of password hashing, so that passwords recovered from one site cannot be applied to a second

Conclusion

- As there is no definite means of detecting a spoofed site, we must use a number of heuristic checks which, in combination, prove reliable.
- These heuristics will force attackers to work harder, and much like spam filtering and virus detection, techniques will have to continue to evolve.
- Use of digitally signed email protects against “phishing” attacks.