Phishing Prevention with Hybrid Authentication/Authorization

Marc Stiegler

Abstract

Phishing is now widely recognized as the number one threat to the enterprise.Two Factor Authentication, the supposed solution to this among other problems, has been shown to be less effective than anticipated. Here we present Two Factor Access Control, blending an authentication factor with an authorization factor, to render phishing attacks ineffective.

Problem statement

As reported by the Chicago Tribune when describing a study by Verizon, "nearly every incident of online espionage in 2012 involved some sort of phishing attack"[1]. Phishing is the most important single threat that must be eliminated for enterprise security.

Many vendors of Two Factor Authentication (2FAn) tout their systems as the solution. Alas, the actual protection against phishing provided by 2FAn is limited. When Twitter suffered a particularly egregious phishing attack, they rushed a 2FAn solution to market. Within days, a YouTube video was published explaining how to phish Twitter's brand new system[2].

2FAn systems come in diverse flavors, but they all share the same basic flaw that gives phishing its power. The flaw is that the credential (a password + transient token in one popular 2FAn setup) is unbundled from the site where it is meaningful. As a consequence the user can be tricked into revealing/employing the credential at the wrong site for the wrong people, via an interactive work flow that is effectively indistinguishable from the normal flow. 2FAn vendors counter that, at least with 2FAn, the breach is limited to a single session. While limiting an attacker to a single session may be valuable in some contexts, for many applications it is quite inadequate. The attack that compelled Twitter's security upgrade achieved all its goals in a single session. We have identified a straightforward 8-step process to escalate a single-session breach of Amazon Web Services 2FAn into an enduring breach of all the virtual machines in the breached account. A more fundamental solution is required.

Solution

One can make phishing inexpressible by tightly bundling an access control factor with the site where it is meaningful. A straightforward way of achieving such tight bundling in a browser-based application is to supply the user with an unguessable url, or *webkey*, as the path to reach his page. A simple webkey might look like this:

https://verySimpleWebkey.com/demo/#s=rhmfi5qb5kwh4i

In this example, the string "rhmfi5qb5kwh4i" is unguessable. It designates a particular resource (often a user account) and, because it is unguessable and can only be acquired by an explicit grant, it is also an authorization to use that resource: the designation and the authorization are bound in a single token. Due to the nature of SSL, this token is only revealed at the site where it is meaningful.

Our solution combines such a webkey with a standard password. Thus we have one authentication factor (a password) and one authorization factor (a webkey), blended to form Two Factor Access Control (2FAcc).

In operation, the user's behavior is little changed: he clicks the webkey stored in his bookmarks (or in a shortcut file on his disk or thumb drive), arrives at the login page, and types his password. The only difference is that he does not need to type a username: the unique webkey he clicked implicitly identifies him, so the login page can greet him rather than demanding to know who he is (see Figure 1). In the event of a lost or stolen webkey, the recovery process can be much the same as the password recovery process. For example, a "lost your webkey?" link can send you an email with a new webkey, closely paralleling the lost password interaction.

Evidence the solution works

Consider the following real-world phishing vulnerability. In one enterprise familiar to the author, all employees periodically receive a "How Are We Doing?" request from the CEO. The email asks the employee to click a link to take the survey. Upon clicking the link, the browser goes to the login page where the employee presents his single-signon password. Hints that this might be a fake login request from a phishing site are relegated to the outer edges of the browser, beyond the focus of the user's attention. The user follows his natural, normal work flow, and surrenders his password. The phishing attack succeeds.

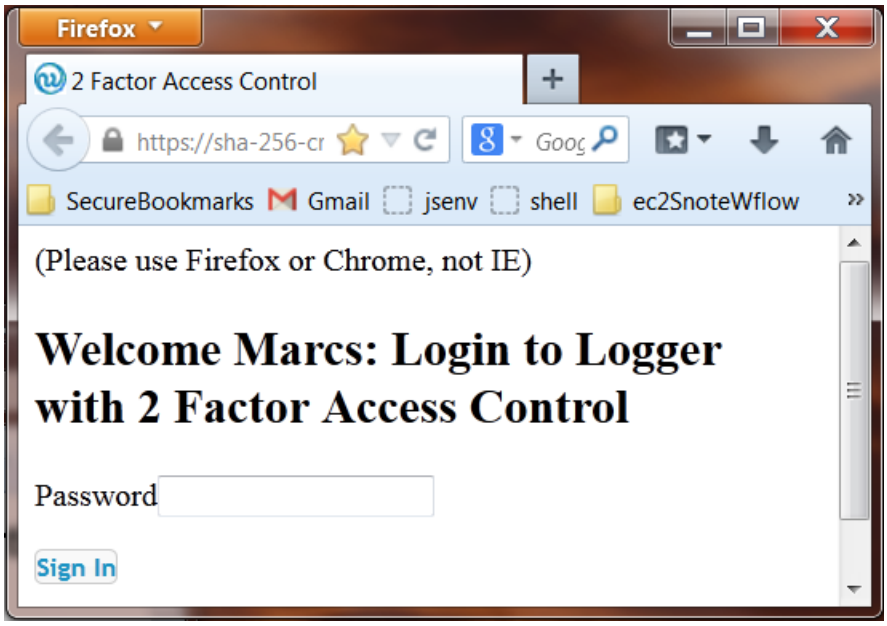

Figure 1. Simple 2FAcc logon page. Since the webkey link is known only to the authorized person, the system already knows who is logging in, and the login page may greet the user (Marcs in this case).

Now consider the utility of such password theft in the presence of a webkey acting as a second factor. The attacker can still steal the password, but it is useless without knowing the webkey that takes the employee to the private page where the password is validated to create session credentials. To trick the user into surrendering his webkey, the attacker must engage in a social engineering attack in which he persuades the user to copy his webkey out of his bookmark list and paste it into a page of the attacker's choosing. Such a demand takes the user dramatically outside the bounds of his normal work flow: though a CEO may periodically request employees to fill out a survey, CEOs generally never ask employees to deliver bookmarks to resources the CEO can access directly. Such a demand offers a strong indication to the user that something is not right in the interaction, enabling the with the smallest amount of training ("if someone asks you for your private webkey, don't give it to them") to recognize the attack and deny access to the assailant.

Some confusion exists about how vulnerable webkeys are to leakage as they traverse the Web. In fact, the risks to webkeys by url collectors are more limited than many realize. When crossing the routing fabric, SSL protects the secret part of the webkey. The vendor using webkeys to protect its clients can store only hashes of the webkeys just as it stores only hashes of passwords, and use the hash for both access and logging. In the browser, if the JavaScript for a page can successfully access the bookmarks, the history, or the location bar for a tab other than its own, that is considered a high-priority security breach. On a typical day, no such breaches can be found in any of the major browsers.

Of course, webkeys are not invulnerable to theft. One more serious risk for webkeys is exposure via shoulder surfing: browsers promiscuously display the full url for every link and page they present. However, the shoulder surfer must then capture the second factor, i.e., the password, to achieve a successful breach. Only with two independent and distinctively different breaches can one overcome two factor access control.

Competitive approaches

Many security experts blame the user and recommend more user training. Since experts generally consider themselves robust against phishing, it is alluring to believe that the problem can be solved by making the users more expert. The success of such anti-phishing training suffers from a glass-half-full dilemma. In one typical study[3], training reduced vulnerability from 30% to 17%. One can claim this is a significant improvement, yet a 100-person company that is reliably breached 17 times by every attacker might consider it to have negligible impact on the company's actual exposure.

Pure webkeys, without a second factor like a password, also make phishing inexpressible. Such webkeys are already used by diverse organizations. Google, YouTube, DropBox, IEEE and HP all use webkeys for access control for some limited purposes. However, pure webkeys are vulnerable to shoulder surfing, as discussed earlier. Using the password as a second factor mitigates shoulder surfing risk just as the webkey mitigates phishing risk.

Browser-side certificates can also eliminate phishing, but such systems suffer from usability issues that have so far prevented them from becoming mainstream. 2FAcc, in contrast, has limited impact on user behavior.

Conclusions

Phishing is the single greatest threat to the enterprise and its customers today. Current anti-phishing strategies do not adequately address the underlying problem, namely, that the credentials are unbundled from the applicable site in such a way that the attacker can trick the user into employing the credentials on the attacker's behalf in a way that makes it a seamless part of the user's work flow. Combing a standard password with an unguessable webkey binds part of the credentials tightly to the site, forcing the attacker to engage in a second, very different kind of attack, to collect all the credentials needed to successfully penetrate the account.

References

[1]"Survey finds computer phishing attacks growing in sophistication", United Press International, July 26, 2013, http://www.upi.com/Science_News/Technology/2013/07/26/Survey-finds-computer-phishing-attacks-growing-in-sophistication/UPI-76151374860723/

[2]Gonzalez, Roman, "How to Hack Twitter's New Two Factor Authentication", 2013, YouTube, http://www.youtube.com/watch?v=Qirz8j9C9QQ

[3]Kumaraguru et. al., "School of Phish: A Real-World Evaluation of Anti-Phishing Training", SOUPS, 2009, http://cups.cs.cmu.edu/soups/2009/proceedings/a3-kumaraguru.pdf