MICHAEL B. SCHER

SSO/RSO/USO/ oh no?

Mike Scher makes his way in life as a security policy and architecture consultant in Chicago. An attorney, anthropologist, and security technologist, he's been working where the policy tires meet the implementation pavement since 1993.

mscher@cultural.com

THE RECURRENT CORPORATE DRIVE TO create[1] a reduced, single, unified, or otherwise simpler sign-on process is experiencing a new growth phase, based on password authentication. The rush to use passwords, indeed to use one password for everything, should give IT security personnel pause. The move to use one password for everything strikes me as significant backsliding. It may be that I differ from many of my contemporaries in that I think even "good" passwords are not always a good idea.

The only thing that's wrong with password authentication is that it uses passwords.—Me

It's one thing to use passwords and a unified login portal when you have 1.5 million customers using your Web-based services, but it's quite another to clump an employee's corporate, financial, business, and administrative access behind a single password. That single password becomes everything, a valuable stepping-stone taking the single-sign-on system into the brave new world of "single break-in."

When passwords are strong, they can become difficult for users to manage. Forcing users to maintain six or more passwords with variant rules, change periods, and so forth won't increase real-world security. When we do find passwords manageable, it's often because we have created weak, predictable, guessable, crackable, or easily stolen passwords. When we succeed in making strong passwords manageable, we have usually done one of three things: tucked them all behind another password (e.g., cryptographic password safes, certificate stores), placed them all someplace risky (e.g., wallet, keyboard tray, text file, Word doc), or made a central authentication system of some sort that lets users just use one or a couple of passwords for everything.

Let's try to define some terms:

- *Unified sign-on:* Passwords are synchronized or authentication is centralized. Think RADIUS servers with passwords, standard Microsoft domain or Active Directory authentication, Sun's NIS/NIS+, central strong-authentication servers, or the many password-synchronizing programs that don't require a central authentication server (but which do require a system to coordinate your global password space).
- *Single sign-on:* You log in to one application or host, and other items know you're logged in, because you send some special blob of data to them that proves it. Each device validates the data you send, either by passing it to a central authentication system or by

cryptographically validating it. SSO implies a flat or tiered trust model, which must be well documented for it to be effective. Full SSO is often deployed with Web-based applications, using browser cookies or complex URLs to "pass the hash," by way of products like RSA Cleartrust and Entrust, and with many Kerberos implementations, Microsoft's included.[2]

- *Reduced sign-on:* What most companies get when attempting to implement single sign-on. At best, systems that share common authentication are grouped by security level and purpose, so the user has a handful of authentication items that "do it all" ranked by security, each group its own trust realm. At worst, there is no security coordination; all systems that could be put on SSO are on one SSO or RSO system, the rest aren't. Then, for example, significant blurring of security realms may occur: Internet-facing, cleartext Web interfaces may take the same password as the corporate benefits and health insurance service system's SSL interface, two IP addresses over.

TYPES OF REDUCED- OR SINGLE-SIGN-ON SYSTEMS

- Password synchronization/coordination systems
- Authentication store consolidation (AD, LDAP, NIS/NIS+)
- Portal-based SSO and authentication gateways (including transparent man-in-the-middle authenticators)
- Token/cookie-enabled "start pages" (sometimes portals)
- Token/cookie distribution centers (Cleartrust/Kerberos, etc.)

Risks

The risks presented by SSO systems stem from two significant audit and coordination issues: authentication credentials and user IDs.

AUTHENTICATION CREDENTIALS ARE LIKE EVIDENCE

I tend to think of authentication as an evidence problem: "Does what the user is providing meet the level of evidence that we require to prove they are who they say they are?" Passwords, for example, are weak as evidence: They can be stolen without the user's knowledge, copied, intercepted, guessed. The stronger the passwords are and the better the mechanisms over which they travel and their compliance with good password policy, the better they are as evidence. Good password policy, auditing, and a well-designed sign-on architecture can significantly boost the efficacy of password-based security.

Nevertheless, on their own, passwords don't really rise above what the legal world might term a "preponderance of the evidence" (essentially, "more likely than not") in terms of evidentiary value. In the effort to make them stronger, we combine them with other bits of evidence: physical access, IP addresses (e.g., the finance service center segment), digital certificates (which are used for most authentication purposes as large passwords protected by small passwords), soft tokens, hardware tokens, time of day, and so on. Just as with evidence in a trial, the more evidence that points to the same conclusion, the more likely we are to believe the conclusion likely is true. Indeed, "more likely than not" on its own is not so bad. It's the standard of proof required in many kinds of civil suit, and it's the standard that (everything taken in the best light for the plaintiff or prosecutor) must be met before a case may go to trial.

RATE YOUR SYSTEMS, ORGANIZE YOUR USERS

For some purposes passwords alone may be "good enough," but remember that we are talking about a shared authentication framework: The selected authentication mechanism is going to be used for many systems. The level of protection required may vary from one system to another.

Thus, before a company implements a new password-based sign-on regime, it should consider a company-wide effort to rate and rank hosts and applications by business-criticality, regulatory requirements, and susceptibility to abuse via stolen authentication credentials. Next, the company should try to organize users' network, host, and application login names into a unified or coordinated scheme. For real SSO, a company-wide identity management scheme helps administrators select the systems to which a given user should have access and the rights of each class of user, and allows ready changes in user authentication rights as responsibilities change. To ensure proper scope, the company must not only define what systems are critical, but also the kinds of system that are to be "in scope" for centralized authentication. At many organizations, Sarbanes-Oxley compliance initiatives are already producing a set of critical systems and authorized users, a first step on the path to real identity management.

The risks of not ranking systems and applications are significant. A system that ought to use a higher-strength credential could be tied into a basic password-based sign-on regime. A server that authenticates "in the clear" (internally or even over the Internet) could be tied in, exposing passwords used for more sensitive applications. Policies that are well defined, user education, and a good audit process are essential to strengthen a password-based sign-on regime.

PASSWORDS AND UNIFIED-SIGN-ON SYSTEMS: LIKE KETCHUP AND CHOCOLATE CAKE?

Depending on the company, information that is sensitive, business-critical, trade-secret, or regulated could require stronger credentials than passwords alone. Some authentication requirements (authorizing transactions or launching events that will have major ramifications) may require standards of proof that come close to "beyond a reasonable doubt." We can achieve those higher levels by using multiple factors that together rise to the level we need. Thus if we can ensure that a password has sufficient strength, is regularly audited, and is never used outside the corporate network, even for remote access, we can have a higher level of trust in it. We could perhaps trust such a password for use on more sensitive systems. Again, policy, education, and audit are key to using passwords across the enterprise, with or without a fancy sign-on system.[3]

RISKS FROM REDUCING/COORDINATING SIGN-ON

- Payoff from password-guessing attacks (dictionary and brute force) is increased. Think single break-in rather than single sign-on.
- Password theft impact increases.
- Coordinated DoS from bulk lockouts becomes a risk.
- SSO may be provided to system-level accounts or to user IDs that have varied access levels across the enterprise, placing the wrong level of safeguard on critical accounts/systems.
- Global authentication credentials may be exposed through integration with external, non-SSL Web systems or through use on third-party, insecure hosts (kiosks and the like); users may also use the same password on third-party systems, from banking to blog sites and beyond.
- Remote authentication gateways and VPNs may use the same, easily stolen authentication credential, making single break-in a real possibility.

Benefits

Don't get me wrong: A well-implemented SSO system can actually increase security, but it's going to take some work to get there. The mechanism behind single sign-on that lets the server know the user has already logged in can be more secure than the basic username/password mechanism an integrated application

used to use, as with using Kerberos tickets instead of user/pass over unencrypted channels. Policy at the company may already be ill-enforced—users regularly selecting weak passwords, leaving them in text files on their laptops, etc.—such that consolidation to just a few passwords may allow better policy enforcement along with increased user convenience.

When combined with a well-executed identity management program, the SSO system may well provide some cost savings. Old accounts on per-seat systems can be removed in a timely way; help-desk costs for password management may go down; new employees may find all their accounts up and working right away; new services can be rolled out on new systems without having users go through the "new password" process. One must take care, however, to rate these savings in a conservative way. Costs will not entirely disappear, and other issues will rise to take their place.

BENEFITS

- Convenience to both end users and developers
- Account provisioning and removal efficiency, and cost savings (including better oversight of per-seat license costs)
- Centrally enforced, consistent authentication policies
- Easier implementation of identity management systems, if the company is trying to get a grip on distributed systems
- Auditing and anomaly-based alerting for application and host login activity

Real-World Costs

Most of the costs are predictable, up-front costs to make the new authentication regime a reality. Do not, however, forget that there will be ongoing costs in addition to the systems themselves. Legacy systems in particular may house critical applications that will require careful, custom coding to enable integration without simultaneously creating security exposures. Depending on the sign-on system implemented, the company may need to step up its audit activities, policy enforcement, and change-control procedures. Lost or forgotten authentication credentials mean the user can do no work until the authentication is replaced or a temporary authentication credential is assigned (one-use password, for example). The increased need to assign temporary credentials so that users can get back up and working leads to a need for superior controls against social engineering. The SSO core systems become critical to the workday, so there need to be redundant systems in resilient (and perhaps geographically diverse), physically secure environments with fault-tolerant networking to the rest of the company.

COSTS

- Fault-tolerant systems in fault-tolerant environments
- Recurring software licenses (and per-user licenses)
- Software integration costs (internal staff and consultants)
- Any hardware/license costs for strong authentication devices
- Heightened audit and staff education costs in some environments

Real-World Quagmires

I've been unfortunate enough to witness several failed unified- or single-sign-on rollouts from the perspective of an external security consultant. The process at the large organizations involved inevitably proceeded in the following manner:

1. Users complain about having a ton of passwords with conflicting creation and change policies (some of which are misguided in terms of the real risk profile).
2. Security admins crack down on users reusing passwords or using poorly constructed passwords, making the issue both an embarrassment and even more unmanageable.
3. Users complain even more; help-desk calls go up as users forget new passwords more frequently than before.
4. Management does a study and finds that over 50% of help desk calls are about password-related or account-creation issues; management may already be looking at or paying for password reset management software.
5. Management calls for SSO, arguing that it's an efficiency and cost issue.
6. IT security calls for tokens and a central authentication system.
7. A vendor-affiliated (or vendor-owned) consulting firm does a study showing how much more token-based SSO/RSO will cost than plain-password USO/SSO because of legacy system issues and per-seat costs for tokens.
8. The consulting firm makes arguments about how much more secure passwords can be with central password policy enforcement, without regard to the costs of ensuring policy compliance.
9. Someone says, "Isn't this just going back to Sun's YP/NIS?" but is quickly hauled out of the building.
10. Millions are slated for analyst input, consultants, redundant hardware, software, licenses, and implementation.
11. By the time the SSO system is up in prototype, management finds that critical legacy systems can't be integrated with it except at very high cost.
12. Other authentication systems start coming in (again). If first time here, go back to step 1 and start over. Otherwise, continue . . .
13. Eventually, wiser organizations base all new authentication on one easy-to-use token authentication system (plain RSA key fobs or SafeWord Silver tokens, for example), while integrating Web-based applications behind one or more single-sign-on portal systems.

Conclusion

Reducing the number of times and ways a user needs to sign on to applications and systems is a worthy goal. Password policies don't scale well when users are faced with six, seven, or more sets of divergent policy implementations (even where policy is consistent). However, one must take care not to assume that integrated sign-on regimes are the only way to straighten out the authentication mess. Further, one must take great care to manage expectations, risks, costs, and timelines from the outset. Set fully articulated objectives based on realistic, cost-justified goals, and it can work.

Finally, base whatever you do on public standards, either open or easily extensible ones. Adhering to standards ensures that future development and new applications cost less to integrate, and standards will meet with readier recognition from business partners, auditors, and new systems personnel. There are several well-fleshed-out standards for portal-based and central authentication integration. For example, for portal-based and Web-integrated SSO, the Liberty Alliance standard[4] (employing SAML),[5] Kerberos, and MS Kerberos[6] approaches are solid, vetted, and well documented. For centralized authentication, there are, of course, RADIUS, LDAP (and S/LDAP), MS AD, and many others.[7]

RECOMMENDATIONS

- Combine any sign-on project with an enterprise-wide identity management project. Benefit: Sarbanes-Oxley audits may be taking you down this path already. Take advantage of the work now, while it's current.

- Combine any sign-on system with strong authentication products, such as hardware tokens on an application-class or enterprise basis, to ensure that critical credentials are less prone to theft, guessing, or other compromise.
- Carefully spell out the real goals of the project. Ensure enterprise-wide buy-in.
- Ensure that most users will experience a noticeable reduction in sign-on events. The best deployments of strong authentication have user demand escalate dramatically, even with no SSO benefit, when users find that they no longer need to deal with a dozen passwords. With an SSO result, the user buy-in can thus be even greater.
- Ensure that the project has realistic cost estimates and measurable goals. Expect integration difficulties and special development and consulting costs. Target specific problems that have readily assessed costs to the enterprise today, so that the predicted cost savings can easily be substantiated.
- Base the final technical architecture on public standards that provide integration examples, reference code, and excellent documentation.

NOTES

1. Or is it re-create? See Andrew Findlay's piece on the corporate world's quest to regain the mainframe login milieu, "Regaining Single Sign-On," at http://www.brunel.ac.uk/depts/cc/papers/regaining-sso.html.

2. The Open Group has an introduction to single-sign-on concepts and the trust relationships they entail at http://www.opengroup.org/security/sso/sso_intro.htm.

3. See G. Ellison, J. Hodges, and S. Landau, "Risks Presented by Single Sign-On Architectures," at http://research.sun.com/liberty/RPSSOA/, and Avi Rubin's risk analysis of Microsoft's Passport SSO protocol at http://avirubin.com/passport.html.

4. See Project Liberty's home pages at http://www.projectliberty.org/; for a good overview of its current market position, go to http://www.eweek.com/article2/0,1759,1619564,00.asp.

5. See http://java.sun.com/features/2002/05/single-signon.html for a practical description of SAML in the SSO realm.

6. See http://www.microsoft.com/technet/security/topics/identity/idmanage/P3Intran_1.mspx.

7. Diana Kelley and Ian Poynter gave an excellent overview of single-sign-on issues at Black Hat USA 2002. Their presentation is available at http://www.blackhat.com/presentations/bh-usa-02/bh-us-02-poynter-sso.ppt, with audio at rtsp://media-1.datamerica.com/blackhat/bh-usa-02/audio/2002_Black_Hat_Vegas_V06-Diana_Kelly_and_Ian_Poynter-Single_Sign_On_101-audio.rm.