Account Management: A Deployment and Usability Problem

Phillip Hallam-Baker

VP & Principal Scientist, Comodo Group Inc.

Abstract

Account management is the biggest challenge most Web users face today. There is no shortage of technology but none has achieved a critical mass of users to date. Reasons for failure include attempting to solve problems the proposers care about rather than the ones that concern the stakeholders required to act and deployment deadlock.

The proposed solution is designed for incremental adoption. Deployment of the first stage requires only action by the user. Once a critical mass of users have opted in, the conditions are set for second stage deployment at the relying parties.

The Account Management Problem

The biggest problem most Web users face today is how to manage the tens, hundreds or even thousands of accounts that they are required to create in order to access certain features of certain Web sites.

Note that the problem as far as the user is concerned is convenience and not security or the ease with which third parties can exchange information claims referring to them.

Thus the account management problem is quite distinct from the problems that people have tried to solve with 'federated authentication' and/or 'identity' systems. While solving the account management problem may facilitate a solution to these other problems, these are not problems the user does or should care about.

Usability

Account management is a usability problem, but not one that needs deep understanding of psychology to improve. Choosing and memorizing passwords is difficult and tedious for the user. Resetting passwords that have been forgotten or confused is even more tedious.

Users have already taken matters into their own hands with a variety of tools and approaches to work around this usability disaster: They write passwords down, they re-use passwords and they game password expiry mechanisms.

We have no need of further studies to tell us how broken it is or how to 'fix' the user. The system is broken and the objective is to fix it.

Security

Security is a requirement that an account management system must meet but not a deployment incentive.

Security is management of risk to assets. Attempting to change user behavior in relation to password management is difficult because the users are asked to invest their time and effort to protect assets that almost invariably belong to another party.

I have well over a hundred accounts at various Web sites dedicated to news and various interests. Each and every one of those accounts has the same password and where possible, the same username. The content I am being asked to protect is not mine and I see no reason why I should take effort to protect it.

Deployment

The state of account management has been known to be unsatisfactory for decades. Many technical solutions have been proposed, many of which provide vastly improved usability. Yet the problem persists.

The main reason is *deployment deadlock*. Web sites are designed to support at least 95% of Web users and are thus limited to the capability of their browsers. Browser providers are unwilling to support new features that can only be used after a ten year deployment cycle has completed. Users are unwilling to adopt new technologies that would limit their access to their valuable web resources to a small number of machines with the correct browser.

Deployment Incentives

While standards activities can sometimes resolve a deployment deadlock, the account management problem has proved particularly resistant. In order to resolve the deadlock it is necessary to carefully asses the deployment incentives for each party involved and demonstrate that whenever a stakeholder is required to take an action that they have a deployment incentive to do so.

User

The user is looking to simplify their use of the Internet and Web. They want to be able to connect to Web sites with minimal hassle while controlling the amount of information they disclose in the process.

The only action that can be expected of a user is that they deploy or upgrade an application on at least one of their devices.

Relying Party

Almost all Web sites and other relying parties are looking to improve the user experience they proved. Some relying parties are also interested in reducing the risk of impersonation.

Relying parties are not necessarily expected to operate their own infrastructure or to be able to publish particular DNS records, deploy DNSSEC or acquire an SSL certificate. Their infrastructure is however expected to be capable of supporting DNS clients that make use of new DNS records.

Account Manager

In this proposal, account management is delegated to an ‘account manager’ service hosted in the cloud. The account manager is chosen by the user and is answerable to the user alone.

Many forms of account management are possible but in the ideal case an end user would have their own personal DNS domain name (e.g. hallambaker.com) and the account manager service would be provided together with Mail, Messaging and other value added services.

Unlike in other proposals, the task of account management is considered to be an important one requiring a certain degree of competence and technological capability. Thus it is expected that an account manager can correctly configure a DNS system using new resource record types and deploy some form of domain level validation (DNSSEC and/or SSL certificate) if enhanced security is to be provided.

Proposal

The proposal is developed in stages such that the initial stages build out the infrastructure necessary to support the subsequent parts.

The chief technical challenge is not devising a protocol proposal. Rather it is to select which existing technology to co-opt (e.g. SAML, Kerberos), that which may be usefully accommodated and that which to decline with prejudice.

Since the primary problems identified are usability and deployment the problem of technical implementation is taken as read. There is insufficient space here to give a full account.

Likewise the problem of privacy in authentication systems is as complex as it is consequential and does not lend itself to a compact discussion of the issues.

Account Identifier

Accounts are identified by means of a Uniform Account Identifier (UAI) minimally consisting of an account component and DNS domain component.

For user interface purposes the UAI is always rendered in RFC822 format, that is *username@domain*. For example phill@hallambaker.com.

For internal purposes where a URI form is required, a UAI is rendered in the form uai:*domain*:*username* for example uai:hallambaker.com:username.

While a UAI has the form of an email or messaging address it is not necessarily an email or messaging address and in the case that it is, it is not necessarily the user’s preferred email address or one that the user reads. Nor is it a requirement that an individual user have only one UAI. On the contrary, they may have separate accounts for acting in separate roles, as an employee of a company, as a student at a university, as a private individual.

The account manager service for a UAI is discovered by performing DNS extended service discovery on the domain component.

Level One: Account Management in the Cloud

Web browsers have supported password storage mechanisms since the earliest days of the Web. Furthermost, browsers also support storage of cookies used to persist site specific authentication credentials.

Moving this browser function into the cloud via a Web Service is an obvious development and several browser extensions are available to achieve this. There is thus a need for an open, interoperable standard to support this mechanism so that users can share their passwords across devices and browsers.

Such a mechanism should have minimal impact on the user. Once activated the mechanism would ask the user if they want to store their passwords in the cloud rather than the local machine. If the answer is yes the browser will do as directed until told to stop. The only impact on the user experience is that from this point on they will have access to all their accounts from any machine.

If the user wishes to access an account from a machine that does not support the protocol they can surf to the account manager site and find the password there.

While this approach introduces a middleman who may defect it is vastly superior to the current situation where users are forced to share their passwords across sites in order to ensure the ability to access them from multiple machines.

As far as the user is concerned, this approach has no security considerations whatsoever. If they don't trust their middleman with their bank login details, they don't sign that site up for the service. If the relying party has a problem with the security of the scheme they have a strong motivation to deploy the enhanced authentication capability described in phase two.

Level Two: Enabled Relying Party

The relying party can improve the user experience for site visitors and improve the security of account management by installing appropriate server extensions.

Most Web sites that employ accounts would prefer to make it as easy as possible for people to register for an account. Much of the difficulty of registration can be avoided if there is an infrastructure that allows a supporting Web browser to inform sites that the user has an account with a federated account manager and thus initiate a sequence that allows the user the option of automating as much of the process as they desire.

One of the key security concerns in any scheme involving a middleman is what will happen if the middleman either defects or is compromised. Another is the risk that the user's own computer is compromised. There is no shortage of unencumbered solutions to these problems. The difficulty in deployment has been caused by the lack of a suitable infrastructure in which they can be applied.

The cryptography required to implement this protocol is quite straightforward and unencumbered. The key observation being that a MAC is a perfectly adequate basis for authentication (as in HTTP DIGEST) provided that the passwords are large enough to prevent brute force attack. Humans cannot generate or remember effective passwords but computers can.

Level Three: Second-Factor Conformation

What has generally been understood as a requirement for multifactor authentication is in most cases nothing of the kind. What is really required is an independent confirmation that the user really intends to authorize a specific transaction.

A two factor authentication technique such as an OTP token or a USB smartcard can only provide us with a greater degree of certainty that Alice is involved. If Alice's machine has been compromised, such techniques cannot provide us with any greater degree of certainty that Alice intends to authorize a specific action.

In a second factor confirmation approach we ask Alice to confirm the specific action and not to confirm the fact that she is in fact Alice.

For example, let us imagine that Alice has registered her smartphone as a second factor confirmation device for interactions with her stockbroker. To access her account from one of her usual machines she uses the standard authentication procedure, she may then perform low risk tasks such as reviewing positions etc.

The second factor confirmation process is only engaged when she is attempting to perform an action designated 'high risk' for example accessing her site from a new machine or day-trading penny stocks. In this case Alice receives a message on her smartphone through an appropriate application that asks her if she really intends to allow a new computer to access her account or to place a trade for $10,000 worth of Pets.com. If Alice agrees a confirmation receipt is generated and signed with Alice's private key so that an audit trail is captured.

Conclusion

While the account management problem has proved difficult to address it is not intractable. The key to improving the situation is to look at the problem from the user's perspective and to equip the user to address the problems they care about without relying on deployment by any other party.

To improve security we should solve the problem from the user's perspective in a manner that allows for incremental deployment towards a more secure authentication mechanism, including deployment of second factor confirmation systems.