CAT OKITA

(digital) Identity 2.0

Cat Okita has more than 10 years of experience as a senior systems, security, and network professional in the financial, Internet, manufacturing and telecom sectors. Cat has spoken at LISA and Defcon about identity and reputation, co-chairs the "Managing Sysadmins" workshop at LISA, and programs for fun, in her spare time.

cat@reptiles.org

EVERYBODY "JUST KNOWS" WHAT identity is—most definitions center around identity as a "fact of being" or "defining characteristics"—but when push comes to shove, identity, like pornography, is very hard to describe exactly. Is your identity your name? Your government-issued photo ID? How you dress? What music you listen to?

What about the digital world? Is your identity your email address? Web site logins? How many "identities" do you have? In fact—is it *your* identity at all, if it's controlled and thrown around by other entities, with or (often) without your knowledge or permission? (Digital) Identity 2.0 is designed to change all of this confusion.

Identity 2.0 Is User-centric Identity Management

The basic idea behind Identity 2.0 is that it puts the users in control of their (digital) identity (or identities—think about the number of logins, memberships, and nicknames you have) and how their identity is used, managed, and given out.

"But wait!" you say. "What's this 'identity' thing anyway? Didn't you just say it's impossible to describe?"

Let's take a step back and talk about what identity is—and isn't. (To get a better idea of some of the issues, see Dick Hardt's informative and entertaining OSCON 2005 Keynote talk on Identity 2.0 [1].)

Defining (Digital) Identity

Since we're talking about computers and programs, I'm going to narrow our scope a bit and use the term "digital identity," rather than "identity," but digital identity as we're going to use it here is more than just an account name, IM handle, or email address. The following serves as a useful definition:

> A digital identity is a collection of claims attached to a digital subject (about which we can then make assertions).

A pithy summary might be:

> Digital Subject A:
>
> Claims: TK421, short, stormtrooper.
> Assertion: TK421 is a stormtrooper.
> Verification: Stormtroopers must be tall—TK421 is short.
> Result: Aren't you a little short for a stormtrooper?

Common Problems with Identity

One of the wonderful things about digital information is that it's easy to copy, modify, move around, and destroy—and once a piece of digital information's been let out to the world, it's pretty much impossible to find out where it's gone and who's doing what with it.

Since it's so easy to copy digital information—and so hard to keep track of where digital information has gone—it's extremely important to know and limit who has access to what information. Of course it's much easier to say that we need to know and limit who has access to what information than it is to actually find out who has what information about you and determine how well it's protected.

In fact, the average G8 citizen is listed in 50–100 databases. Even if each database knows only a few things about you, combining information can reveal things you thought were private. It's far too easy to picture a world where Bob's insurance company decides to change what Bob's health plan costs just because they found out that Bob checked out a book about diabetes from the library and started to buy diabetic chocolate at the grocery store—even if he was actually trying to help Alice!

On top of that, who actually owns your information? If we think about medical lab results, do those results belong to you? Your doctor? The lab? Your (medical) insurance company? Some combination of all of the above? If that's the case, who gets to make decisions about sharing that information?

In fact, the usual way that you'd find out about a piece of digital information in the wrong hands is catastrophic failure—your credit card has been used to buy cell phone equipment on a different continent, or the government wants to have a word with you about tax evasion.

Myths About Identity

There are a few common myths that come up as soon as you start discussing identity—starting with the idea that there's a single definition for identity that everybody agrees on.

ONE TRUE NAME

> Everybody should have one (and only one) true name/nym. The name should be unique and identify the individual forever.

This myth usually stems from the naive idea that it's easy to come up with a scheme that will give each person One True Name—and that everybody will cooperate and play by the rules. It also relies on the availability of some sort of central system to track who owns what True Name and some sort of mechanism for tracking people down to prevent them from using a True Name belonging to somebody else.

There are a bunch of problems with this idea, starting with the need for universal adoption, how to enforce unique identifiers, maintaining privacy, and what to do if somebody steals or misuses your One True (forever) Name. And, in the end, should we really care whether a name is unique at all? Most of us manage to fumble our way through life without having terrible problems as a result of knowing two people named Dave Smith.

ONE TRUE ROOT/REGISTRY

The One True Root/Registry is often found as a close cousin of the One True Name, and it takes form around the idea that there should be some sort of global registry for identities, like IP addresses and DNS names. It's an interesting thought—but the sheer logistics involved in just registering 6.6 billion people are mind-boggling.

Some of the reasons suggested for the One True Root are:

- It ensures that identities are unique.
- It establishes a universally valid identity.
- It can be used for tracking and legal enforcement.
- It can reduce identity theft.

But who would be trusted to run this global database? What about privacy concerns and validating (or fixing) information? What should be done when conflicts arise (as they inevitably will)? These are all hard problems and completely aside from the intransigent requirement for universal adoption.

BIOMETRICS WILL SOLVE EVERYTHING

This is a nice way of saying "If we have your [DNA|iris scan|fingerprint], we can prove, beyond a doubt, that you're . . . uh . . . you." This is all very nice, but it's much like saying that you're never lost, because you always know where you are. Not only that, but there's no way to revoke biometric credentials, so once there's any sort of bad data (No-Fly lists, anyone?) associated with your biometrics, you're just plain out of luck.

So What Is User-centric Identity Management?

That would be one of the million dollar questions. Stepping back for a moment, let's take a look at the types of things that get called "identity management," and then move on to looking at the properties we'd expect to find in a user-centric identity management system.

There's a range of things that people mean when they talk about identity management. Jon Callas helpfully broke down the types of things that get called "identity management systems" into four main categories:

IM(1): Traditional ■ AAA (authentication, authorization, accounting) ■ Per-device user accounts ■ PKI	IM(2): Traditional 2.0 (ways to make IM(1) easier, but all localized) ■ Directory services ■ LDAP ■ Single sign-on ■ NIS/NIS+ ■ Kerberos
IM(3): Database Management ■ Information management ■ Metadirectories ■ HR information resource management (phone numbers, titles, parking spaces, conference room reservations, etc.)	IM(4): Marketing ■ Loyalty programs ■ Buying habits ■ Targeted selling ■ Recommendation systems

All of these systems have a few things in common. They are local. They are controlled by a single authority, such as an employer or a retailer. Moreover, information about the user is controlled and managed by an authority other than the user—in many cases, users may not even know what information about them is in the system.

What Properties Should a User-centric Identity Management System Have?

If we look at Identity 1.0 management systems, it's pretty clear that they're far from user-centric. In fact, users have little to no control at all over their information, who has it, and what's being done with it. This obviously raises questions about privacy, security, accountability, trust, and manageability.

A number of smart people have written (at length) about the properties a user-centric identity management system should have. I'm firmly of the opinion that one of the key properties is usability. It's been proven time and time again that people reliably screw up complicated things, so if it's not easy for users to manage and understand what's being done with their information, it's like handing them a genie in a bottle. It's easy to let the genie out of the bottle but awfully hard to put the genie back (if you can at all!).

One of the most widely advertised lists of properties for a user-centric identity management system comes from Kim Cameron of Microsoft. Unfortunately, Kim's "Laws of Identity" [2] are a miserable failure when we're talking about something anybody can understand. Here are his seven laws:

1. User Control and Consent
2. Minimal Disclosure for a Constrained Use
3. Justifiable Parties
4. Directed Identity
5. Pluralism of Operators and Technologies
6. Human Integration
7. Consistent Experience Across Contexts

Dr. Ann Cavoukian [3], Ontario's Information and Privacy Commissioner, took Kim's laws and produced a notably clearer (although still rather long-winded) interpretation:

1. Personal Control and Consent
2. Minimal Disclosure for Limited Use: Data Minimalization
3. Justifiable Parties: "Need to Know" Access
4. Directed Identity: Protection and Accountability
5. Pluralism of Operators and Technologies: Minimizing Surveillance
6. The Human Face: Understanding Is Key
7. Consistent Experience Across Contexts: Enhanced User Empowerment and Control

Ben Laurie [4] of Google produced a nicely succinct set of properties for an identity management system that are easy to understand and remember and are a great place to start thinking about what we want from user-centric identity management:

1. Verifiable
2. Minimal
3. Unlinkable

I always like to make the implicit need for systems that people can use explicit and add a fourth property to Ben Laurie's three:

4. Usable

What Do These Properties Mean?

I'll cheat briefly and use Ben Laurie's pithy properties and their associated comments as a starting point.

1. Verifiable: There's often no point in making a statement unless the relying party has some way of checking whether it is true. Note that this isn't always a requirement—I don't have to prove my address is mine to Amazon, because it's up to me where my goods get delivered. But I may have to prove I'm over 18 to get the alcohol delivered.

In other words, can we prove enough about what you're claiming to believe that it's true enough and, if we can't, who's accountable for the bad information?

2. Minimal: This is the privacy-preserving bit—I want to tell the relying party the very least he or she needs to know. I shouldn't have to reveal my date of birth, just prove I'm over 18 somehow.

If we're going to take control of our identities, we have to start by not handing everything about ourselves over to anybody who asks. In security terms, this means deny all and permit selectively. In identity terms, this means anonymity with selective (and minimal) disclosure.

3. Unlinkable: If the relying party or parties, or other actors in the system, can, either on their own or in collusion, link together my various assertions, then I've blown the minimality requirement out of the water.

Of course it's impossible to be either anonymous or selective if it's easy to put together a few claims and figure out who you are or what you've been doing. I'm sure everybody's had the experience of their mom pointing out that they can't possibly have gotten mud all over if they stayed inside all day.

4. Usable: If I can't figure out how to use this system, or it's easy to do the wrong thing (whether that's giving out my information to everybody or letting somebody steal it), it doesn't matter how good the system is at everything else.

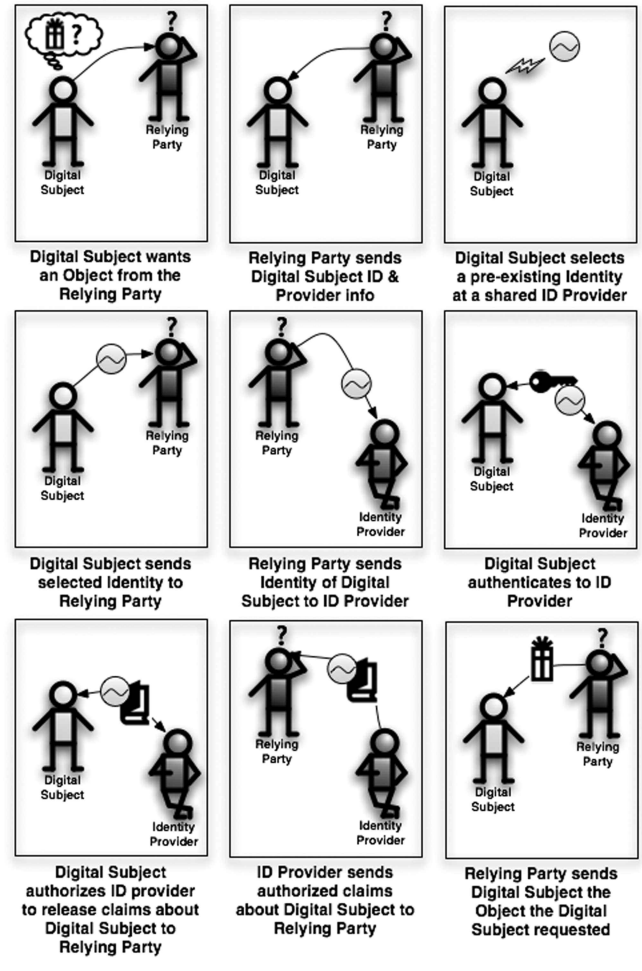
In the end, all of this is helpful only if it's easy to understand, and anybody can figure out how this whole user-centric identity management thing works—and do it without losing their identity.

All That Aside, What's Identity 2.0 Good for, Anyway?

You'd think that Identity 2.0 being user-centric identity management would mean that it's all about the user—and you'd be partly right. Identity 2.0 is definitely about the user, but there's more than just the user in the mix.

The basic architecture of identity management 2.0 has a user (Digital Subject) with one or more sets of characteristics (Identities), contacting a service provider (Relying Party), to obtain a service, which is authenticated and/or authorized by an identity management service (Identity Provider) on behalf of the user.

That means that we've got users, service providers, and identity providers, all of whom have their own vested interests. Eliot's dad wants to stop having to remember tons and tons of online user IDs and passwords; MIT wants to be able to share library logins with Harvard; Dan wants to be able to buy smut without having to give away his birthday; whitehouse.com wants Dan



to be able to buy smut easily (and without worrying about his privacy or credit card numbers going missing); the government wants to be able to identify and serve constituents without violating their civil rights; and Verisign loves the idea of providing yet another registry service. It sounds like everybody's a winner here.

Of course, the big winners in all of this focus on Identity 2.0 are the middlemen. We've got plenty of users and service providers already. The most common design for implementing Identity 2.0 does a great job of creating new business for identity providers and middleware brokers.

So, Who's Doing This Stuff?

We've got all of the usual suspects involved—big corporations on their own or in groups, the free/open source community, standards bodies—and a variety of interesting implementations.

There's been a remarkable convergence in the Identity 2.0 space over the past year. Microsoft's CardSpace is still holding down one corner. The Liberty Alliance (composed of almost everybody other than Microsoft) has formed up in another. OpenID is being cheerfully adopted by the Web 2.0 crowd and is also collaborating with Microsoft. And everybody's using (or at least supporting) SAML (the OASIS Security Assertion Markup Language). Table 1 provides a summary of those involved.

Other nontraditional players, such as Google, Yahoo!, and Amazon, are also sneaking into the identity management space, with APIs that enable single sign-on for their properties or redirect authentication queries to third-party servers.

	Verifiable	Minimal	Unlinkable	Usable	Comments
Microsoft CardSpace	Yes	Yes	No	?	http://cardspace.netfx3.com/ Requires Web Services Trust Language Typically requires additional middleware Aimed toward end-user e-commerce
Liberty Alliance	Yes	?	?	?	http://www.projectliberty.org/ Aimed primarily at enterprise use cases Single sign-on/logout, permission-based attribute sharing, circles-of-trust, interoperability certification Uses SAML
OpenID	?	Yes	?	Yes	http://openid.net/ Low/no trust authentication only Distributed free Lightweight (by comparison, at least) Actively in use, primarily with blogs (e.g., Six-Apart, Wordpress, and a variety of blog software, including moinmoin, drupal, phpBB, and mediawiki)
Shibboleth	Yes	Yes	No	Yes	http://shibboleth.internet2.edu/ Uses SAML Web single sign-on only Most common in academia
Pubcookie	Yes	Yes	No	Yes	http://www.pubcookie.org/ Web single sign-on within an organizational domain Most common in academia
Google Apps					http://google-code-updates.blogspot.com/2007/02/new-apis-for-google-apps.html Allows authentication to be redirected to a third party for hosted apps http://code.google.com/apis/accounts/AuthForWebApps.html Allows applications to authenticate against and use Google services
Yahoo! BBAuth					http://developer.yahoo.com/auth/ Allows external Web applications to authenticate against and use Yahoo! services
SAML					http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

TABLE 1: PLAYERS AT A GLANCE

Are We There Yet?

Is Identity 2.0 mature—or even walking yet? Not really. We're starting to see OpenID implementations popping up all over, and there's enough convergence in the Identity 2.0 space to suggest that we're starting to hit critical mass.

Unfortunately, there are still plenty of outstanding questions about security and why you'd want to trust a third-party identity provider (or providers) with your information. It's certainly true that you'd have fewer places where you'd have to remember passwords (or some other form of authentication), but that also means that the identity providers become richer targets, and it certainly makes collusion among providers (or relying parties and providers) much, much more interesting.

Beyond that, is it really user-centric identity management when you're still trusting and relying on third parties to do the right thing? Or is it just rearranging to whom you've contracted the outsourcing of your identity?

REFERENCES

[1] Dick Hardt gives the best talk on Identity 2.0: http://www.identity20.com/media/OSCON2005/.

[2] Kim Cameron, "The Laws of Identity": http://msdn2.microsoft.com/en-us/library/ms996456.aspx.

[3] Ann Cavoukian, "7 Laws of Identity": http://www.ipc.on.ca/images/Resources/up-7laws_whitepaper.pdf.

[4] Ben Laurie, "Laws of Identity, Revised": http://www.links.org/?p=222.