

A L E R T

Computer Security Institute, 600 Harrison Street, San Francisco, California 94107, tel: (415) 947-6320

INSIDE INFORMATION



Tools and Techniques 4
A phishing solution that's easier to use than it is to not use.



Concerns 6
New proof-of-concept virus uses RFID tag as attack vector.



Concerns 7
Despite recent knocks to security, Mac's still a safe bet.



Privacy and Security 8
Mobile device and removable media security self-awareness test.



Power Tools 10
SSL Digger—Roots out those weak cipher specs.



Background Notes 12
Amplified DDoS attacks exploit DNS recursion.

What InfoCard Is and Isn't

We continue with our look at the state of identity management

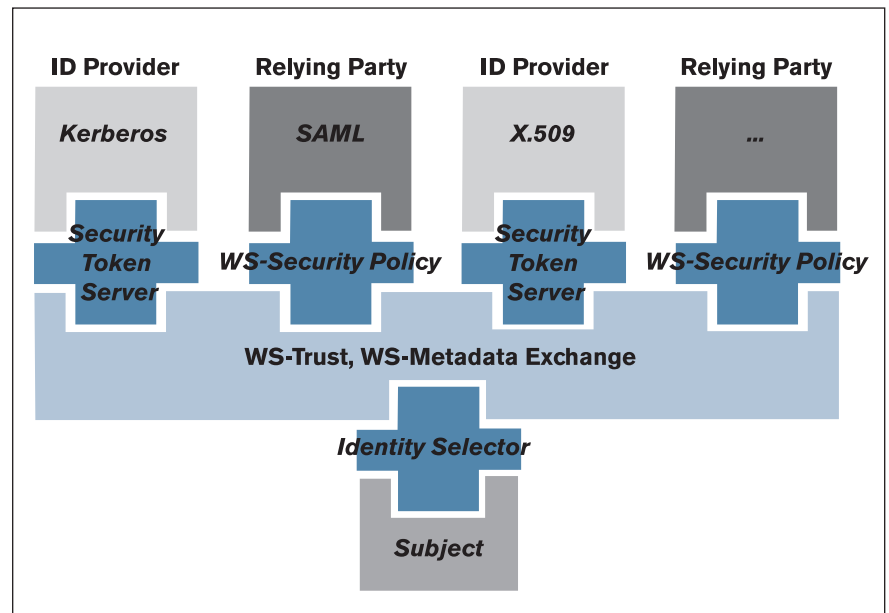
There's little doubt that the Microsoft marketing engine will get itself geared up to tell the public at large what InfoCard "is," but in the meanwhile, getting a handle on the next major security-related software introduction is remarkably difficult. It's a slippery topic.

The place to start, however, is with the diagram below from an overview of the "Identity Metaverse" by Microsoft's identity guru Kim Cameron.

The box at the very bottom of the diagram is you, the subject. If you go to a Web site or an application that requires you to establish that you're authorized to use its services (where in the past you'd have been challenged for a username and password), you'll instead be shown an interface where you can choose from what appear to be traditional "ID cards."

Simply put, that interface is InfoCard. That's it.

Or, at least, that's how to draw a line around it that differentiates it from everything else. Obviously, there's more to it than that. For one thing, it's running in a different security context than the rest of your applications on whatever operating system you happen to be running. It's supposed to be completely cordoned off in terms of memory access and the like. Other applications (and, say, viruses that have installed themselves unbeknownst to you) can't see memory that's being used by the InfoCard interface.



Cameron does note that “if you get a rootkit, you’re in trouble. But Vista makes it much less likely that you’ll get one, because you almost always run in your own context (e.g. not at ‘root’ privilege). A virus caught in your user context cannot see your InfoCard screen or memory.” There are other security gains as well, Cameron notes: “InfoCard protects against keyloggers because typing of shared secrets becomes obsolete. And social engineering attacks are mitigated because Web sites no longer control the user experience. Finally, sensitive information like a credit card number is never stored on the PC, or visible to a virus running there.”

InfoCard presents your various credential possibilities to you in the form of “cards,” so not too surprisingly there’s also a mechanism for generating your own self-signed InfoCard and then issuing encrypted tokens when the card is used (in other words, there’s a tool for making yourself into an ID Provider, which Microsoft’s documents often refer to as an IP, but which we’ll call an IDP in the hopes of not creating confusion around the already overloaded “IP” acronym)—this too is part of InfoCard.

Finally, there’s a strong sense that this is what Microsoft thinks every operating system’s authentication interface should look like: an isolated page where you pick from your various ID cards. This really isn’t about Redmond wanting everything to look like a version of Windows—in fact InfoCard is trying to look a bit different than the rest of the Windows Vista operating system. Rather, it’s supposed to look different from everything else altogether, so that you the user realize you’ve entered one of those transitional moments where you may be handing over some of your personal information.

But other than these pieces, everything else in the identity management universe is something other than InfoCard. The part where the InfoCard interface talks across the network and exchanges information isn’t InfoCard, but the WS-Trust standard. The server that creates a token that attests that you’ve got authorization to use a certain service isn’t InfoCard either, but something like a certificate authority (CA) or perhaps something a little more old-fashioned like a Kerberos server. The primary thing that InfoCard does is allow you to choose which of several identities you want to use in a given situation where you’ve been challenged for ID. The “cards” represent your various identities.

The “cards,” it’s vital to note, don’t contain information about you, per se. You won’t find your name and address or your social security number stored in one of your cards. Instead, enough metadata is stored that when the appropriate moment arrives, InfoCard can communicate to the IDP to say who you’re supposed to be. The IDP will confirm this by challenging you in one way or another (doesn’t matter to InfoCard what that way is—it’s completely agnostic in this important respect—but it may very well matter to the Web site that is requesting the information).

The Seven Laws

Kim Cameron’s paper titled “The Laws of Identity” has become something of a philosophical rallying point among the “Identity 2.0” crowd. And not without reason—it’s a compelling and surprisingly “non-Microsoft” document. Cameron’s clearly not trying to bring us a warmed-over version of Passport.

The laws, in a nutshell (and a good deal more about them is said within the paper, which can be found at http://www.identityblog.com/?page_id=352), are:

1. User Control and Consent

Digital identity systems must only reveal information identifying a user with the user’s consent.

2. Limited Disclosure for Limited Use

The solution which discloses the least identifying information and best limits its use is the most stable, long-term solution.

3. The Law of Fewest Parties

Digital identity systems must limit disclosure of identifying information to parties having a necessary and justifiable place in a given identity relationship.

4. Directed Identity

A universal identity metasystem must support both “omnidirectional” identifiers for use by public entities and “unidirectional” identifiers for private entities, thus facilitating discovery while preventing unnecessary release of correlation handles.

5. Pluralism of Operators and Technologies

A universal identity metasystem must channel and enable the interworking of multiple identity technologies run by multiple identity providers.

6. Human Integration

A unifying identity metasystem must define the human user as a component integrated through protected and unambiguous human-machine communications.

7. Consistent Experience Across Contexts

A unifying identity metasystem must provide a simple consistent experience while enabling separation of contexts through multiple operators and technologies.

So the IDP plays an important role in this, but as we mentioned above, may in some cases actually be you, as self-provider of a card (this is the situation you’ll find yourself in at a Web site that asks for a login name or e-mail address but otherwise doesn’t care who you are). The other player (besides you, the user of all this splendor) is the Web site that wants to know who you are in the first place. In today’s pre-InfoCard world, this site would normally challenge you for a username and password and check up on your assertion that you are in fact you on its own steam. With InfoCard, this site becomes a Relying Party (RP) and actually gets its assurance that you are you by way of the IDP.

There are early releases of InfoCard in the hands of developers, and blog reports so far make it clear that it’s pretty fragile just yet—it takes just the right combination of operating system release, Explorer browser preview and InfoCard code to make the thing work. It *does* work if you get it all right, but would seem that there are only a handful of non-Microsoft people in the world who’ve managed to InfoCard their way into a site (such as Cameron’s identityblog.com). As Cameron puts it, “it’s new, it’s evolving quickly, and it hasn’t stabilized yet.”

What happens at game time

So with the various pieces in place, we can walk through the mechanics of an InfoCard transaction. We’ll talk here about going to a Web site, but clearly there are other use cases, such as internal

applications that directly invoke the InfoCard interface to authenticate the user with an intranet application, perhaps built on a service-oriented architecture.

Arriving at the site

I'm an InfoCard-enabled user and I arrive at my bank, which has now implemented support for this interface. My arrival causes a page to be sent to my browser, as would always be the case. Indeed, the page my still contain all the usual paraphernalia for a traditional login.

Triggering the InfoCard process

What's also in the HTML page that is sent to my browser, however, is an HTML `<OBJECT>` tag. The browser, which also has to be up-to-date, recognizes that this object has a "type" parameter that identifies it as an InfoCard request. It therefore triggers the InfoCard dynamic link library (DLL) module. The stage is set and the screen dims (I'm not kidding, it really does dim—another way of differentiating this process from normal computing activities as well as a way of making the process harder to spoof).

InfoCard gears up

Among the parameters passed to the DLL from the `<OBJECT>` tag are the claims about the user that need to be proven. These might be things like the user's name, but on the other hand, the Web site may only need to know some anonymous piece of information, such as that the user is older than 21. Generally, the site should only have requested what it needs to know. The DLL compares the claim requests to the user's InfoCards to see what claims can be met by which cards, and then displays those that can meet the request (others are visible but grayed out).

The user picks a card and is challenged

This is an important moment if you think about it. The user may use any card that meets the requirements of the Web site's request. A user might maintain different personas with different sets of proofs for different contexts. With the selection made, the DLL contacts the IDP via WS-Trust. The IDP then does whatever it needs to do to authenticate the user. Possibly it asks for a username and password; possibly a one-time password must be used or some biometric proof supplied.

A secure token is issued and reviewed

Assuming the user successfully authenticates with the IDP (not the Web site, which is the RP in this scenario, it's important to keep in mind), the IDP places the appropriate claims into an XML document and then uses the RP's public key to encrypt them. This is sent not to the RP but back to the user's InfoCard process, which displays the claims that are about to be sent so that the user can review them.

The approved claims are forwarded

If the user is comfortable with passing the information in the claims along to the Web site, they press a Submit button and the encrypted token is forwarded to the RP, which will now grant access to the user.

The Web object in more detail

Jumping back a step, notice that the mechanism for invoking the InfoCard interface really is pretty much as simple as it sounds. A

snippet of HTML code is added to the rest of the material in the Web page, as in this example from Andy Harjanto's Infocard Weblog (<http://blogs.msdn.com/andyhar/archive/2006/02/20/535333.aspx>):

```
<FORM method="post" action="https://www.fabrikam.com/Main.aspx">
  <input type="submit" name="InfoCardSignin" value="Log in id="
    "InfoCardSignin" />
  <OBJECT type="application/infocard" name="xmlToken">
    <PARAM Name="tokenType" Value="urn:oasis:names:tc:SAML:1.0:
      assertion">
    <PARAM Name="issuer" Value="http://schemas.microsoft.com/.../
      issuer/self">
    <PARAM Name="requiredClaims" Value="http://schemas.microsoft
      com/.../emailaddress"
  </OBJECT>
</FORM>
```

Notice that this example shows a Web site that requires a SAML assertion for authentication. The RP may not get to dictate that I'll provide my credentials or that I'll provide a specific credential if there are several that meet the need, but it does get to dictate what kind of credential must be provided if it's to be considered sufficient.

Specifically, the RP can make requests concerning:

- The issuer;
- The type of token that will be returned;
- What claims must be vouched for by the token;
- Requirements regarding the kind of proof used (symmetric, public key, etc), the size of the key used in authentication, and other such details as might be required for high-security scenarios.

It's worth underscoring that the RP only receives proofs of the specific claims it requests, not access to any kind of full profile of data about the individual. The user (or, at any rate, not the RP) gets to choose where data used for this particular user's authentications are stored. This ability to separate authenticated claims from specific identities is potentially a huge gain for Internet privacy. This would be true even in relatively small ways: one can imagine being able to post comments at a blog site anonymously, but only after proving that one had the reputation (from actions at other sites) of never posting spam. Anonymity is preserved while the social good of keeping out bad actors is upheld.

On the other hand, we shouldn't overstate how much may be gained in the real world—RP's may still very well want a full complement of information, including name, address and credit card numbers, before selling you their products. And once they've got the information, they may well decide to store it, even insecurely.

As an aside, Microsoft has taken the interesting step of essentially not providing any kind of normal application/programming access to InfoCards. They are stored in their own little world; there is no API to access them. The effect of this is that cards don't get deleted or modified or added without the user's direct involvement, because these steps must be taken through the InfoCard interface.

For the InfoCard interface to be invoked, of course, there has to be some software resident on the user's system. At present, it gets there by way of a purpose-built software file (a DLL file) that has to be expressly loaded along with Internet Explorer 7. These things will be part and parcel of Microsoft Vista, when it's released

INFOCARD is continued on page 11

INFOCARD, continued from page 3

next year, but users who stick with XP will have to download these pieces in order to use InfoCard.

Given that migration to Vista is bound to take place at a measured—perhaps even downright reluctant, depending on the vicissitudes of the market—pace, one question is whether the requirement for additional specialized software will make Web site developers reluctant to get involved. Obviously, they can use pre-existing login routines for users who don't have InfoCard capability on their machines, but having two systems will just complicate life. Cameron says it's not all that much more complicated, however: "We've taken this into account so the changes to a Web site are absolutely minimal."

Organizations may or may not decide that dealing with InfoCard is worth the trouble—it will have to move beyond its current proof-of-concept stage before anyone can decide—but one thing organizations don't have to do, should they opt to use InfoCard, is run Windows servers. From the "Microsoft's Vision for an Identity Metasystem" white paper:

Non-Microsoft applications will have the same ability to use "InfoCard" to manage their identities as Microsoft applications will. Non-Windows operating systems will be able to be full participants of the identity metasystem we are building in cooperation with the industry. Others can build an entire end-to-end implementation of the metasystem without any Microsoft software, payments to Microsoft, or usage of any Microsoft online identity service.

Just to prove that this is so, Cameron, who's in charge of the InfoCard project, moved his identityblog.com over to non-Microsoft software (completely so: he's running the classic, open-source LAMP stack). The blog is running on WordPress (also open source) and he's written his own PHP scripts to handle the InfoCard login process. By Cameron's own admission, it's still a bit buggy and it lacks a certain degree of polish:

Some of the user experience is still pretty "basic". Like what happens if you click on InfoCard login and don't have InfoCards installed. When I have some time I'll make that take you to a page that tells you what InfoCards are, how they work, how to install them, and that sort of thing. But for now, the behavior should appeal to lovers of cryptic error messages.

So at least in theory, the Linux and Macintosh systems of the world could implement compatible identity selectors, RPs and IDPs that were all compatible with InfoCard. And, really, it's only that it's Microsoft doing the developing that makes it seem like InfoCard is the driving force here. In point of fact, InfoCard's mission is to work with WS-Trust, an open standard (we could quibble about how open it is, but at least there's nothing preventing anyone from using it). So the open standards for identity, such as WS-Trust, are really the driving force behind InfoCard. In any case, identity management seems to be entering something of a 2.0 phase, and there's no question that InfoCard will play a significant role in whatever that turns out to be. ■ — R.R.

COMPUTER SECURITY ALERT

Editorial Director

Robert Richardson rrichardson@cmp.com 305-455-8572

Associate Editor

Sara Peters speters@cmp.com 212-600-3066

Education Director

John O'Leary joleary@cmp.com 972-596-6384

Director

Chris Keating ckeating@cmp.com 212-600-3390

Conference Director

Jennifer Stevens jstevens@cmp.com 212-600-3353

Sr. Marketing Manager

Nancy Baer nbaer@cmp.com 415-947-6364

Special Projects

Pam Salaway psalaway@cmp.com 631-878-2205

Director of Sales

Jody Nurre jnurre@cmp.com 516-562-5055

Membership Director

Mary Griffin mkgriffin@cmp.com 516-562-5457

Design Director

Jim Shinnick jshinnick@cmp.com 413-582-0622

Project Manager

Frank Brogan fbrogan@cmp.com 212-600-3356

Conference Manager

Maggie Stumpf mstumpf@cmp.com 212-600-3179

Account Executive

Dina-Marie Frangella dfrangella@cmp.com 212-600-3031

Asst. Membership Manager

John Slesinski jslesinski@cmp.com 516-562-5743

Project Coordinator

Rebecca Kattleman rkattleman@cmp.com 212-600-3384

Fax:

847-763-9602

For address change or membership questions, e-mail csimemberservice@gocsi.com, or call 800-250-2429 (toll free) or 847-763-9602 (outside the U.S.) between 5 a.m. and 5 p.m. PST. Fax 847-763-9602.

Copyright ©2006, Computer Security Institute, 600 Harrison St., San Francisco, CA 94107. All rights reserved. Reproduction in any form is forbidden without express permission of copyright owner. Computer Security *Alert* is sent monthly to CSI members.

www.GoCSI.com

CSI 30

18 Gained
12 Lost

↑ 2

End of March: 227

The CSI 30 stock index tracks publicly traded companies that focus primarily on information security products and services. The index is weighted by relative market capitalizations at the end of 2002, when the index was set at 100. The index is designed for informational purposes only and should not be construed as investment advice.