MicroID considered harmful (to privacy)

Brown University Technical Report CS-08-09
June 20, 2008
http://www.cs.brown.edu/~cce/microid/

C. Chris Erway
Department of Computer Science
Brown University
cce@cs.brown.edu

ABSTRACT

MicroID is a deployed Internet standard designed for use as a lightweight, decentralized identity primitive in web applications and communities. This study presents the standard's specification and deployment, and analyzes the security and privacy of MicroID, describing attacks that can be used to compromise the privacy of its users. Although it has been described by its designers as privacy-preserving, in practice the deployment of MicroID has put the private information of many of its millions of unwitting users at risk of compromise. We provide recommendations for changes to the standard and its deployment which prevent these attacks.

1. INTRODUCTION

For decades, the Internet has carried "user-generated content"—yet in recent years the term (and acronym, UGC) has generated much excitement on the Web, where online communities, blogs, and businesses built from the contributions of data by users have proliferated. The data collected by these sites include obvious forms such as digital media—text, images, videos, *etc.*—but also span personal, demographic, and social details, as well as metadata such as ratings and preferences. Websites that collect UGC vary in size, ranging from that of an independent blogger and her commenters to million-strong communities like YouTube, Slashdot, MySpace, Flickr, and Facebook.

The popularity of UGC-based services as a means of expression has instilled in some of its user-contributors a sense of identity, forged from the set of content associated with oneself online. However, these users typically participate in many online communities, holding separate accounts in each, making consolidation of identity and content difficult. Furthermore, these users are burdened by the chore of managing multiple logins, passwords, and personal details for each site they use.

New standards have been proposed to address these and other issues surrounding online identity consolidation. Protocols such as OpenID [16] and OAuth [13] allow for distributed authentication and access delegation. Formats such as FOAF [19] and the "microformat" XFN [19] enable the publication of "friend" edges and other social information. Groups such as the DataPortability project [4] (which include Google, Microsoft, Facebook, and others as members) promote the adoption and development of these standards. The overall goal of these efforts has been to encourage easier sharing and greater federation of identity information.

This paper focuses on one such technology, MicroID, which claims to offer "a lightweight identity layer for the web." We will present the standard (in Section 3), analyze the threats to privacy made possible by its use (in Section 5), and empirically evaluate the severity of these privacy threats to users of popular websites that have adopted it (in Section 6).

2. BACKGROUND

The MicroID standard was introduced in March 2006 by Jeremie Miller, who explained,

> One of the best ways to lay claim to your food as a kid is to lick it, make sure that your sibling or peers know it's yours. Well, MicroIDs are a bit like that in the digital world, you can stamp a MicroID on your content, sites, and individual pages. A hosted service or member site can include MicroIDs on every participants profile, comments, content, ratings, microformats, or anything. Then these can be independently verified and even aggregated into third party services (anonymously and *without any loss of privacy to boot*).
>
> ... The most exciting aspect is that it empowers end users with absolute control while fully protecting their rights and privacy. It's also the model of decentralized systems, allowing anyone to participate and enabling services to crawl and index and provide a fully anonymous utility. All of this is critical for anything relating to Identity on the net. [11] (emphasis added)

Miller's proposal, MicroID, involves the creation of identity tokens which can be inserted into web pages as non-visible metadata. The purpose of these tokens, according to its designers, are to enable "anyone to claim verifiable ownership over content hosted anywhere on the web" [9], and to "enable service providers to 'stamp' information and reputation based on a validated URI associated with an individual" [12].

The MicroID standard has enjoyed moderate success with "Web 2.0" sites and advocates of new web identity systems. Under the slogan "small decentralized verifiable identity," MicroID's web site [9] provides specifications for developers, and advocates for its deployment, leading to its adoption by more than a dozen popular online services. The DataPortability project lists it among recommended standards for use by developers and individuals [15]. In 2007, a draft specification of MicroID was submitted to the IETF [12], where it remains a work in progress as of the time of writing.

3. SPECIFICATION

A MicroID token is constructed by applying a cryptographic hash function to a pair of uniform resource identifiers (URIs) identifying the individual and resource to be linked together:

$$H(H(individual\_uri)||H(resource\_uri))$$

Typically SHA-1 is used as H. The *individual URI* identifies the individual and is used to verify claims of ownership: it may describe the individual in any way that can be phrased as a URI (such as an OpenID URL, or an XMPP address), but in practice an email address (as `mailto:` URI) is typically used. The *resource URI* describes the content that is being associated with the individual: in practice this is the HTTP URL of a web resource that the individual wishes to "claim." For example,

```
SHA-1( SHA-1("mailto:cce@cs.brown.edu") +
       SHA-1("http://cs.brown.edu/~cce") )
```

denotes the hashing steps necessary to link this author (identified by email address) to his homepage under the MicroID standard. Finally, the URI scheme names, along with that of the hash algorithm, are prepended to the hash (in hexadecimal) to produce the full MicroID token (*e.g.*, `mailto+http:sha1:375960cdf566ee457ba372bdaac4ab3bc85b0b62`).

MicroID tokens are intended for use in HTML documents; the specification recommends inserting them as META tags on pages (*e.g.*, a user's profile page) or as class attributes near HTML elements (*e.g.*, a user's comments or ratings) associated with the individual described in the first URI.

Trust

Since anyone with knowledge of two URIs can trivially create a token linking them, some verification of URIs (*e.g.*, email address verification) must be performed for a token to carry any meaning. The MicroID specification admits this issue, but leaves the task of URI verification up to the publisher (or "issuer"), essentially requiring that token consumers blindly trust any website containing a MicroID token to have been verified. They recommend only that:

> An issuer should not generate a MicroID until it has verified that the individual or service provider has control over a given entity URI. Methods for such verification are out of scope for this specification and may vary according to local service policies and the URI scheme in question. [12]

This requirement of trust in the MicroID publisher forces token consumers to evaluate publishers on a site-by-site basis: while identity verification might be expected from institutions and businesses, it may not be a reasonable expectation from less-reputable sites. Bogus tokens might be used to spoof or "frame" targeted identities by attempting to link them to content with which they have no association. The MicroID developers answer this concern by pointing out that bogus tokens fall outside the typical use case, since a MicroID user would only be interested in claiming content that he actually owns:

> Q: Anyone can make a MicroID for me if they know my email address, how does that prove anything?
>
> A: Yes! A MicroID doesn't prevent spoofing, it simply enables ownership verification. I know that doesn't make sense, but think about it with a real world example, pet tags. You put your phone number on your pet's collar (or microchip implant these days) to identify that pet as yours. Sure, anyone can label their pet as yours, but what do you care? With the microchip example it's even more clear, when you go to the vet they can check your ID and match it with the implant owner data, they validate that you own that pet. MicroID allows a service to validate that the content you link to on some other site, is actually yours if you claim it to be. [10]

However, as explained in Sections 5 and 6, bogus MicroID tokens pose much less of a threat to users than verified ones. By requiring that publishers verify their users' identities, and use that private information (now more valuable for having been verified) to create tokens, the MicroID standard has put this private information at risk of attack.

4. DEPLOYMENT

Publishers

While the MicroID specification remains officially agnostic on recommending URI schemes for identifying individuals, in practice all known deployments have focused solely on linking email addresses to HTTP URLs. Sites that have become MicroID publishers employ the tokens in META tags on their users' personal profile pages, combining each user's profile page URL and registered email address to generate and publish a MicroID token on their behalf.

As of the time of writing, at least 14 websites—ranging from large UGC communities like Digg and Last.fm to smaller sites like WikiTravel, and the identity providers MyOpenID and ClaimID—publish MicroID tokens on user pages [5]. Nearly all the deployments currently known to this author now publish MicroID tokens on behalf of all their users automatically; of these, only one allows users to disable token publication (this exception is WikiTravel, which does not publish tokens by default). This practice, however, weakens the privacy of users of services that have deployed MicroID, as explained below in Section 5.

Consumers

Uses for these millions of published tokens have been slow to develop. Only two known MicroID consumers exist: the identity provider ClaimID and social search engine Wink. Both offer MicroID-based verification of ownership claims, through a process that works as follows:

Consider a registered ClaimID user who wishes to claim ownership of a web page, such as her personal blog or Digg user profile. She provides the URL of this page to ClaimID and clicks "verify." The ClaimID server retrieves the page and looks for a META tag containing a MicroID token; if one is found, the email addresses she has registered with ClaimID and the aforementioned URL are hashed together to generate one or more MicroID tokens, which are compared to the retrieved token. If the page's token matches a generated token, the verification process is complete and the word "*Verified*" is added to the user's ClaimID public profile, next to a

link to the newly-claimed page. If a token is not found, the user is shown a META tag containing a token for her claim, and is instructed to add it to the source of her page.

For these MicroID consumers, the process facilitates the construction of a "social web of trust" centered on a user's ClaimID or Wink profile page. The intention of these services is to enable users to collect links to pages and services representing them on the web, and thus present to the world a consolidated list of their identities and contributed content through *Verified* links from this central source.

5. SECURITY AND PRIVACY

MicroID has much in common with password hashing techniques used for decades in Unix systems. In this view, the URL of the web page being asserted as owned by a user resembles the salt, while the user's email address serves as the password, since it is the private element that, ideally, should be kept secret from outsiders.

However, continuing this analogy, one could say that the deployment of MicroID has placed its users' email addresses at risk of dictionary attacks [8]. Rather than hide these hashed secrets in a `shadow` file, MicroID adopters publish them for all to see, in convenient, semantic form: even the URI schemes and hash algorithms (*i.e.* `mailto+http:sha1`) are included in plain view, helping attackers to more easily guess the plaintext responsible for a token's hash. Particularly, verified email addresses, while longer than conventional passwords, are subject to stricter formatting constraints.

This places the email addresses of users at risk of compromise, contradicting privacy assumptions that affected users may have previously held. Digg and Last.fm, for example, tell users that their email addresses will not be published in their profiles, and will be used only for site-related communication. This leads to assumptions on the anonymity of account identities that are weakened by MicroID adoption.

Individual privacy

Consider, for example, a hypothetical Digg user who posts comments or other materials defaming an institution such as her employer or government. This user might have assumed that since her email address was not published by Digg, her comments were made anonymously—or, at least, reasonably safe from being linked to her real-life identity. Yet armed with a list of suspected email addresses (*e.g*, provided by a corporate IT department, from tax records, or a "watch list"), such an institution might be able to verify ownership of her Digg profile against her will (and without need for cooperation with administrators of Digg), discover her real-life identity, and take punitive action.

In the absence of a list of candidate addresses, brute-force cracking would require more significant computational effort but remains feasible; current desktop computers can perform tens or hundreds of thousands of SHA-1 hash operations per second. In this regard, MicroID, which requires only $2n + 1$ SHA-1 hash steps to check n email addresses against a token for a known URL, is comparably much weaker than password-based cryptographic standards like PBKDF2 [6], which prescribe at a minimum 1000 hash iterations to protect against cracking attempts. Even with these key strengthening techniques, software- and hardware-based approaches can mount brute-force attacks on PBKDF2-encrypted passwords at speeds ranging from hundreds to one thousand password attempts per second [3, 14].

Email address harvesting

Widespread MicroID adoption may also benefit spammers, who might seek to harvest lists of verified email addresses from published tokens. Spammers already employ brute-force techniques such as directory harvest attacks [1] (DHAs), whereby lists of common names are provided as candidate recipients to mail servers; acceptance of mail for a guessed name indicates a verified address, enabling a spammer to build sender lists from targeted domain names of value.

Email addresses harvested from MicroID tokens offer greater benefits to marketers than DHAs, since in practice these tokens appear on profile pages alongside a user's site activity history (*e.g*, a Last.fm user's favorite and recently played songs; Digg user's links, comments, and ratings). The ability to use a MicroID publisher's user information for marketing campaigns (*e.g*, selling ringtones to Last.fm users) add value to these harvested addresses. This weakness, again, contradicts users' assumptions of privacy, since many sites issue privacy policies promising that registered email addresses will not be sold or given to third parties.

Hash functions and non-interactivity

These weaknesses stem from the simplicity of MicroID's design (often described as one of its benefits) and a misuse of cryptographic hash functions. As one of its developers explained in his blog,

> Jeremie also called MicroID radically simple, and he was absolutely right. The core technology of MicroID is a simple hashing function ... and this radically simple technology may change how we think of ownership and social trust on the web.
>
> ... The URL and the email are *hashed together* to produce a unique identifier ... which becomes a *shared secret* between a content provider and a verifier. [18] (emphasis original)

However, simple hashing fails to protect this "shared secret," and is precisely the reason why MicroID fails to protect user privacy. Cryptographic hash functions provide collision resistance, not encryption. Publishing hash outputs as MicroID does yields weak protection against cracking attacks, leading to the potential compromise of private information. Even with key strengthening techniques, email addresses present easier targets for guessing than passwords.

Finally, publishing tokens on the web removes any requirement of interactivity from the verification process, compromising forward privacy: an attacker can retrieve tokens and challenge them off-line, without raising alarms at the publisher's end. We evaluate attacks on these weaknesses in the next section.

6. EVALUATION

This section evaluates weaknesses in MicroID deployments at three popular websites: Digg, Last.fm, and ClaimID. We attempt to find the email addresses of users of these websites through dictionary methods similar to those used to check for weak passwords.

Methodology

MicroID tokens were retrieved from the three services using published sources at each site. Digg provided easiest access to these tokens: their public API's "list users" method allows callers to retrieve batches of usernames, 100 at a time, starting at any index in the list of all registered users. Subsequent "get user" calls for each username provide the user's MicroID, profile URL, and full name information, if the user has supplied it. Using this API we retrieved MicroID tokens and information for 56,775 random users, 31% of whom included full names.

Last.fm and ClaimID provide no API for retrieving information about users, but do use a fixed URL scheme for publishing user profiles, allowing for discovery of profile URLs using search engines. Yahoo!'s Site Explorer [20], which provides in tab-delimited format details on the first thousand results of any search, was thus used to retrieve URLs likely to contain user profiles. This yielded 917 ClaimID and 784 Last.fm profiles; however, these profiles were not chosen at random, since their appearance in the search results were likely influenced by factors such as the number of incoming links to each profile page. In parsing the retrieved pages, all of which included a field containing name information, a minority of names were found to be the same as the username (save for differences in punctuation) and have been left out of the "users supplying full names" tally in Table 1.

Rule-based guessing attack

Here we assess the vulnerability of the tokens to practical attacks, *i.e* those that would require significantly less effort than pure brute-force cracking attempts. Two key observations guide our address-guessing methodology: (1) many users choose predictable, consistent usernames when they sign up for online services, and (2) many (if not most) email users today use a small number of providers to receive email. The latter has also been observed by online marketers [7, 17], one providing evidence of a long-tail effect: he lists 10 domains receiving 65% of his newsletters in 2006, with the top 5 representing 57% of emails. We used their reports to compile a list of 34 candidate domains for use in this attack.

The selection of candidate local-part usernames was guided by the first observation above: that many users choose usernames similar to their email address when registering for online services, occasionally adding numbers or punctuation when conflicts arise. We also drew inspiration from the technique used by Unix password-checking programs of permuting name information found in the GECOS field of `passwd` files to generate weak passwords. These intuitions yielded a list of rules for permuting usernames, first and last names, initials, numbers, and punctuation together in various order in attempting to guess a user's email address. These rules are not as exhaustive as those used by password-checking programs: they are intended to provide a first-pass assessment of the vulnerability of user email addresses, rather than a utility for harvesting spam lists.

	Digg	ClaimID	Last.fm
Total users examined	56,775	917	784
Users supplying full names	17,339	637	708
Guesses based on:			
Solely username	12,413	171	105
Permuted username	383	24	3
Permuted full name	1,498	117	41
Top 5 email domains	12,627	300	139
Total addresses guessed	14,294	312	149
Percentage of total	25%	34%	19%

Table 1: Details of rule-based guessing attack.

Results

The results of this experiment, detailed in Table 1, show that a large minority—between a third and a fifth—of users of these services are at risk of having their email addresses easily guessed by an attacker using their MicroID token and profile information. Surprisingly, the majority of those email addresses "cracked" required only a username and a list of the five most popular email domains to discover, requiring an almost trivial number of hash operations (eleven).

This suggests that a spam harvester could easily run this attack while concurrently making API calls or downloading profile pages, and face a greater performance bottleneck due to network latency and server response time than for hash computation. Success rates in harvesting these email addresses likely far outreach those due to cruder methods such as directory harvest attacks, and yield better results, since each address discovered describes a verified user with a complete history of activity on her profile page. Furthermore, the attacker can choose to continue more exhaustive brute-force attacks offline, without requiring interactive communication with the user or publisher after a token has been retrieved.

The figures produced by this evaluation suggest that as many as half a million users may be vulnerable to this simple attack, allowing unauthorized parties to discover their email address and link them to their activities online. Registered user counts for Last.fm and ClaimID are unknown, but requests to the Digg API currently indicate the site lists over 2.6 million registered members.

7. RECOMMENDATIONS

Adopters of MicroID, by assuming hash functions protect user privacy, have been lulled into a false sense of security. However, both stronger and simpler methods can be used to accomplish MicroID's goals without opening the vulnerability described in the previous section. Here we present several recommendations for users and websites.

Stop hashing e-mail addresses

Clearly, email addresses should not be used as identifiers and published in such a way that enables attackers to easily guess them. Current MicroID publishers concerned for the privacy of user email addresses should stop publishing MicroID tokens in this way immediately. Individuals who wish not to have their email address linked to their user profiles and content should remove tokens from content they control. (It should also be noted that the FOAF RDF vocabulary includes a `mbox_sha1sum` property similarly designed for

"privacy and SPAM-avoidance reasons," [2] but this element has not been deployed as widely as MicroID.)

Use nonce-based email addresses

In the short term, users affected by MicroID deployment can attempt to prevent attacks such as the one above by registering for services with less easily-guessable email addresses. For example, most providers allow users to postpend a plus sign and a token to their email (*e.g.* `user+token@gmail.com`) to create a new email address that will be delivered to their mailbox. However, this technique does not protect against brute-force attacks. Of course, some individuals already publish their email address on their web pages; for these users, inserting a MicroID token on these pages simply adds a different representation of the same information.

Alternate identity URIs

The use of alternate forms of identity URIs, such as those requiring user authentication, could prevent MicroID from being used to compromise user privacy. Web standards such as OpenID [16] and OAuth [13] already permit the delegation of authentication between web services; these protocols rely on more conventional cryptographic algorithms and interactivity to implement authentication ([12] mentions that an identity URI might also refer to an OpenID URL). This would allow Alice, a Digg user, to claim ownership of her profile page by using OpenID to prove that she holds a valid Digg account. The ownership process (described in Section 4) through which the word *"Verified"* appears near a link on Alice's ClaimID profile could then be implemented by simply asking Digg to verify that Alice held a valid Digg account. However, this process (requiring multiple rounds of communication between the individual, verifier, and authenticator) contradicts the lightweight goals of MicroID.

Alternate standards

Simpler, more lightweight approaches could also solve the problem areas addressed by MicroID. A user willing to modify her homepage's source code on behalf of a MicroID consumer in order to "claim" it (as described in Section 4) would probably not balk at, instead, inserting a random nonce token that held no private information. Though it requires page authoring skills, this approach is similar to that taken by Google's Webmaster Tools and Analytics programs to verify a user's control over a web page.

Additionally, the XFN [19] `rel="me"` link attribute, while not claiming to provide any verification features, is already widely deployed for the purpose of identity consolidation: an anchor link marked with this attribute indicates a `"me"` association between the linked content and the linking page's owner. Here, bidirectional links suffice to prove two URLs refer to the same identity: no hashing or private information is required. Furthermore, this attribute's use supports the goal of search indexing (as demonstrated by Google's Social Graph API) first advanced by MicroID's developers.

Finally, cryptographic signature schemes allow users to verifiably associate themselves with digital content: a user wishing to claim her homepage could sign its contents, or its URL, and publish the signature online. However, public verification of these claims typically rely on a public-key infrastructure, which in practice has remained an elusive goal online. The use of PKI alternatives such as a PGP-style "web of trust" or identity-based cryptography may present a direction for future work.

8. CONCLUSION

This paper presented a relatively new web standard, MicroID, which has seen adoption despite its obvious weaknesses in the area of user privacy. Strangely, the standard has seemed to profit most from vague, but enthusiastic definitions of its purpose by its proponents, coupled with desire by web startups—under the mantle of supporting "open standards"—to adopt it on assurances that it empowers users while fully protecting their privacy. In the evaluation, we show that this is not the case, instead finding that the main effect of MicroID's deployment to date has been (1) to apply the word "*Verified*" near a small number of links on the web, and (2) to put at risk of compromise the private information of hundreds of thousands of its unwitting users.

9. REFERENCES

[1] Boldizsár Bencsáth and István Vajda. Efficient directory harvest attacks. In *Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems*, July 2005.

[2] Dan Brickley and Libby Miller. Foaf vocabulary specification 0.91 (retrieved 2008-5-19). `http://xmlns.com/foaf/spec/`.

[3] Yoginder S. Dandass. Using FPGAs to parallelize dictionary attacks for password cracking. In *HICSS '08: Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, Washington, DC, USA, 2008.

[4] The DataPortability project. `http://www.dataportability.org`.

[5] ClaimID.com Inc. Known MicroID publishers. `http://claimid.com/microid`.

[6] B. Kaliski. PKCS #5: Password-based cryptography specification version 2.0. RFC 2898, Internet Engineering Task Force, 2000. `http://www.ietf.org/rfc/rfc2898.txt`.

[7] Tom Kulzer. Top 10 email domains of 2006. `http://www.aweber.com/blog/email-deliverability/top-10-email-domains-of-2006.htm`.

[8] Ben Laurie. MicroID. `http://www.links.org/?p=85`.

[9] MicroID – Small Decentralized Verifiable Identity. `http://microid.org/`.

[10] MicroID Development FAQ. `http://microid.org/blog/?cat=4`.

[11] Jeremie Miller. Introducing MicroID. `http://www.jeremie.com/blog/index.php?entry=entry060325-131146`.

[12] Jeremie Miller, Peter Saint-Andre, and Fred Stutzman. MicroID. Internet draft (work in progress), Internet Engineering Task Force, December 2007. `http://www.ietf.org/internet-drafts/draft-miller-microid-01.txt`.

[13] OAuth. `http://oauth.net/`.

[14] OpenCiphers: Sha1/pbkdf2 wpa brute-force. `http://openciphers.sourceforge.net/oc/wpa.php`.

[15] The DataPortability Project. For developers. `http://wiki.dataportability.org/display/dpmain/For+Developers`.

[16] David Recordon and Drummond Reed. OpenID 2.0: a platform for user-centric identity management. In *DIM '06: Proceedings of the 2nd ACM workshop on Digital identity management*, 2006.

[17] Derek Sivers. Most popular customer email domains. `http://cdbaby.org/stories/07/08/11/8225835.html`.

[18] Fred Stutzman. Creating a Social Web of Trust with MicroID: Part 1. `http://chimprawk.blogspot.com/2006/06/creating-social-web-of-trust-with.html`.

[19] XFN - XHTML Friends Network. `http://gmpg.org/xfn/`.

[20] Yahoo! Site Explorer. `http://siteexplorer.search.yahoo.com`.