CSSE442 – Computer Security

Sid Stamm stammsl@rose-hulman.edu
Rose-Hulman Institute of Technology
Computer Science and Software Engineering Department

Identity Notes

1 Who are you?

To apply access control when you use it, a security mechanism must first authenticate you. We've discussed authentication (remember binding a user to a subject?), but how does a computer system identify a user?

In computer security, we define an *Identity* to be a computer's representation of some unique entity. This unique entity is often called a *Principal* and could be a person, a process, digital data item, or another system. Inside the system is an Identity. Outside the system, it is a Principal. They are bound together or associated by authentication.

A computer *authenticates* me by mapping a user (principal) to a subject (or identity). When I type in a username (stammsl) and password, I'm authenticating: asserting that I should be identified as stammsl so long as I provide the proper password. In this case, the authentication binds me (the principal) with the subject (stammsl).

2 Representations of Identity

Identities don't always represent people, however. Sometimes identities represent inanimate objects like files or other resources. Really, they're unique tags applied to different objects that exist outside a system.

2.1 Users and Groups

User identities are generally tied uniquely to a single human principal. We tell people not to share their passwords and many systems want fine-graned control over who has access to what.

But can a principal be connected to multiple identities in a system? Of course! You may have multiple email accounts on a server or have multiple logins for your favorite social networking site. The computer system may not want to deal with multiple identities, however, requiring you to authenticate each time you want to switch. Can you be logged onto your computer as multiple users at a time? Perhaps, if there are ways for the computer to separate each principal!

Groups are a way for a system to treat a collection of principals as one. Often times administrators want to assign rights to a collection of users (for example, all students). In this situation,

principals may authenticate under one identity, but each identity may belong to a number of groups as well — this allows the system to use group assignments to specify a protection state.

2.2 Files and Objects

Files within a system can be identified by *name*. (Often, this name includes a location, so an identity of a file may be the entire path: C:\Windows32\foo.dll)

Files *outside* the system require more location information. There are a couple of standards that are widely used ways to identify files on the Internet.

Uniform Resource Locator (URL) is a format used to tell a system how and where to find something. URLs should look pretty familiar:

```
http://www.example.com/index.html
```

Alone, URLs are not always enough to identify a file, only to *locate* it. A URL can't describe the file, only tell you where to find it.

Uniform Resource Name (URN) talk about the characteristics of the file, but not *where* the file exists. A URN is a specific format of metadata, for example, this URN would refer to Matt Bishop's Book[1].

```
urn:isbn13:978-0201440997
```

Alone, URNs are not always flexible enough to identify a file, only to *characterize* it within some namespace (like ISBN numbers).

Unform Resource Identifier (URI) is a broader category encompassing both URL and URNs. A URI can be used to identify a file by location or characteristics, though is most often used as a URL (web address). Figure 1 illustrates how these three formats relate.

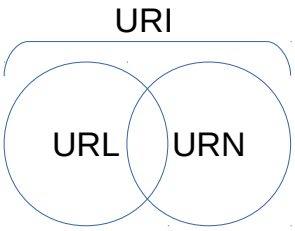

Figure 1: Diagram showing how URI, URL and URN relate.

2.3 Certificates

Certificates are intended to bind an identity to a cryptographic key. More strongly, we would like a certificate to bind a *principal* to a cryptographic key. How does a certificate properly perform this one-principal-to-an-identity mapping?

Distinguished Names are a series of key/value pairs that can identify a principal in various namespaces like "Organization" [2]. Figure 2 shows the Distinguished Name for the rose-hulman.edu certificate. This data in the certificate can be used to identify the subject, and must be "authenticated" somehow.

```
CN = www.rose-hulman.edu
OU = EIT
O = Rose-Hulman Institute of Technology
L = Terre Haute
ST = IN
C = US
```

Figure 2: The Distinguished Name in the "Subject" field of the certificate served by rose-hulman.edu

Who authenticates the subject names? Good question. The Certificate Authority (the one who signs the certificate and binds the identity to the key) is in charge of making sure the principal is properly reflected in the Subject field of the certificate (the identity).

What happens if two certificates are issued for two different principals, but the Subject field (identity) is the same? This violates our initial definition of Identity, requiring one principal per identity. X.509 assumes the certificate authorities will take care of this by properly validating all the information in the subject field, especially the CN for web sites (making sure the principal is actually the organization who runs the web site).

This is prone to error, however. The certificate authorities don't often catch attempts to fraudulently get certificates and that breaks down trust in the system.

2.4 Trust

While some systems perform identity validation or authentication directly, identifying a principal is not always straightforward. Some systems require proof of trust for authentication: you may be able to say "I am Sid Stamm", but in order to trust that statement you need to show that other people believe you. This is different than X.509 where we must blindly trust some trust anchors (the certificate authorities). In other systems, we don't have to blindly trust anyone, but can gauge how much other people generally trust an assertion of identity.

PGP (Pretty Good Privacy) relies on this type of crowd-sourced trust. Alice creates a key, then gets as many other people as possible to "sign" her key, or essentially vouch for her assertion that

she is Alice and this is her key. They do this by meeting her in a trustworthy way (such as in person). When Alice presents her key to someone, she also presents all the signatures. Someone who wants to validate this is Alice and her key can simply look at the signatures to see who has "vouched" for her. Various different measures of trust can be implemented (more signatures means more trust, only trust people who have signed my key, etc), which makes this system pretty flexible.

For example, say Alice's key (0x411CE) is signed by Bob, Charlie, and Debbie. She presents the key to Virgil and he wants to decide if he can trust that Alice controls key 0x411CE. He looks at the signatures and notices that Charlie signed it. He knows Charlie very well and ultimately trusts him, so he decides Alice's assertion is true.

On the other hand, Walt sees the same key from Alice and doesn't know any of the people who signed it. He sees many signatures though, and that is good enough for him so he decides to trust Alice controls key 0x411CE.

2.5 On the Web

How are principals identified on the World Wide Web? Many different ways. You've seen X.509 Certificates. But there are more!

Host ID. Your computer has an ID: the MAC address on your network card is unique to your machine. The public IP address given to your machine is likely unique so that packets can get to your computer. A web site's host name (DNS) is hopefully unique so you get to the right website when you type it into your browser.

Browser State. Web servers assign session IDs to users who authenticate on their site. These are short-lived identities used to connect requests together into a session. These are stored in cookies! Can you steal this identity?

3 Degrees of Identity

Strong identification is great, but sometimes you don't want to be identified. All you need is *Authorization* and not *Identification* in many situations. In order to access a shared resource, why should I need to prove who I am? Shouldn't it be good enough to prove I have the "right" to access it? To get into the buildings on my university campus, do I need to identify myself or just partially?

And perhaps I don't want to be identified at all in those situations where I don't need rights to something. This is Anonymity, and is for a future lesson.

Further Reading

[1] Matt Bishop. *"Computer Security: Art and Science."* Addison-Wesley. 2002. ISBN 978-0201440997

[2] S. Kille, "RFC 1779: A String Representation of Distinguished Names". March, 1995. `https://tools.ietf.org/html/rfc1779`