Chapter 13 REPRESENTING IDENTITY

- In computer science, an identity is the basis for assignment of privileges.
- An identity is also integral in the design of a protection domain.

13.1 What is Identity?

- Identity is simply a computer's representation of an identity.
- A principal is a unique entity. An identity specifies a principal.
- Authentication: binding of a principal to a representation of identity internal to the system.
  - Each systems has its own way of expressing this representation.
  - However, all decisions of access and resource allocation assume that the binding is correct.
- Identities are used for several purposes. Two main ones are for accountability and for access control.
  - Accountability: requires an identity that tracks principals across actions and changes of other identities.
  - Accountability is tied to logging and auditing. Hence, it requires an unambiguous identification of the principal involved.
  - Access control: requires an identity that the access control mechanisms can use to determine if a specific access should be allowed.
  - Most systems base access rights on the identity of the principal executing the process.

Chapter 13 REPRESENTING IDENTITY

13.2 Files and Objects

- **The identity of a file or other entity depends on the system that contains the object.**
- **Local systems identify objects by assigning names.**
- **The name may be intended for human use (e.g., a file name), for process use (e.g., a file descriptor), or for kernel use (e.g., a file allocation table).**
- **Each name may have different semantics.**
- **EXAMPLE: The UNIX OS**
  - **Four different types of file names.**
  - **The <u>inode</u> uniquely identifies a file.**
  - **Processes read files using a <u>file descriptor</u> that abstracts the inode into a representation that the process can read from, write to, and so forth.**
  - **Once created, the file descriptor cannot be rebound to a different file.**
  - **Processes and users can also use <u>file names</u> that identify files by describing their position in the file hierarchy.**
    - **UNIX file names may be absolute path names that describe the locations of files w.r.t. to the root of the UNIX file hierarchy.**
    - **UNIX file names may be relative path names that describe the locations of files w.r.t. to the directory in which the current process is executing.**

Chapter 13 REPRESENTING IDENTITY

13.3 Users

- **In general, a user is an identity tied to a single entity.**
- **Systems represent user identity in a number of different ways.**
- **The same system may use different representations of identity in different contexts.**
- **EXAMPLE: The UNIX OS**
  - **Versions of the UNIX OS usually represent user identity as an integer between 0 and a large number (65,535).**
  - **This integer is called the <u>user identification number</u> (UID).**
  - **Principals (users) may also be assigned <u>login names</u>.**
    - **Each login name corresponds to a single UID, although one UID may have different login names.**
  - **The same principal may have different identities. Typically, each identity serves a particular function.**
- **EXAMPLE: The UNIX OS**
  - **Versions of the UNIX OS provide several types of user identities.**
  - **Because a user is a subject, and a process executes on behalf of a user, the various identities are associated with processes.**

Chapter 13 REPRESENTING IDENTITY

  - The real UID is the user identity at initial login, but it can be changed.
  - The effective UID is the user identity used for access control.
  - For example, if only UID 22 can read a particular file, and a process' real UID is 22 and its effective UID is 35, the user will not be able to read the file.
  - If the process' real UID were 35 and its effective UID were 22, access would be granted.
- To limit the need for special privileges, many UNIX systems (e.g., Solaris and FreeBSD) provide a saved UID.
  - Whenever effective UID changes, the saved UID is set to the value of the effective UID before the change.
  - The user can switch among the real, effective, and saved UIDs.
  - This allows the process to be given root privileges, use them for a limited time, drop them, and reacquire them later.

Chapter 13 REPRESENTING IDENTITY

13.4 Groups and Roles

- **The entity may be a set of entities referred to by a single entity.**
- **The members of the set must be distinguishable.**
- **The set may have an identity separate from any of its elements.**
- **Principals need to share access to files.**
- **Most systems allow principals to be grouped into sets called <u>groups</u>.**
- **Groups are a short-hand tool for assigning rights to a set of principles simultaneously.**
- **There are <u>two</u> implementations of groups:**
  - **The <u>first model</u> simply uses a group as an alias for a set of principals.**
  - **Principals are assigned to groups, and they stay in those groups for the lifetime of their sessions.**
  - **The <u>second model</u> allows principals to change from one group to another.**
  - **After each change, the rights belonging to the principal as a member of the previous group are discarded and the rights of the new group are added.**

Chapter 13 REPRESENTING IDENTITY

- **The difference between these two models lies in the representations of identity.**
  - **In the former model, the identity assigned to a principal remains static.**
    - **It is the principal identity and the set of identities of each group that the principal is a part of.**
    - **This identity does not change throughout the lifetime of the session.**
  - **In the latter model, the identity of the principal is the identity of the user and the set of identities of each group of which the principal is currently a member.**
    - **It is dynamic, and if the principal changes from one group to another, the identity of that principal also changes.**
- **EXAMPLE: The UNIX OS**
  - **UNIX users are assigned membership to a group when they login.**
  - **Each process has two identities: <u>user identity</u> and <u>group identity</u>.**
  - **On older UNIX systems, each principal can be in only one group at a time.**
  - **On other UNIX systems, each principal can be in several groups at a time. On login, the user is placed into all groups of which he is a member.**

Chapter 13 REPRESENTING IDENTITY

- A role is a type of group that ties membership to function.
- When a principal assumes a role, the principal is given certain rights that belong to that role.
- When the principal leaves the role, those rights are removed.
- The rights given are consistent with the functionality that the principal needs to perform the tasks expected of members of that role.
- EXAMPLE: DG/UX System
  - System administration privileges belong to the <u>sysadmin</u> role, not the <u>root</u> user.
  - That user's rights are restricted.
  - The <u>sysuser</u> user can assume the sysadmin role to administer the host, or the <u>netadmin</u> role to administer the network.

Chapter 13 REPRESENTING IDENTITY

13.5 Naming and Certificates

- **Chapter 9 described certificates as a mechanism for binding cryptographic keys to identifiers.**
- **The identifier corresponds to a principal.**
- **The identifier must uniquely identify the principal to avoid confusion.**
- **Suppose the principals are people. The identifiers cannot be names because many different people may have the same name.**
- **How many people named "John Smith" are there?**
- **The identifier must include ancillary information to distinguish the "Matt Bishop" (a professor at UC Davis) from the "Matt Bishop" (works at Microsoft Corporation).**
- **EXAMPLE: X.509v3 certificates**
  - **X.509v3 certificates use identifiers called <u>Distinguished Names</u>.**
  - **A Distinguished Name identifies a principal.**
  - **It consists of a series of fields, each with a <u>key</u> and a <u>value</u>.**
  - **When written as strings, the fields are separated by "/" and the key and value by "=."**
  - **/O= Microsoft Corporation/OU=Quality Assurance/CN=Matt Bishop/**
  - **/O=University of California/OU=Davis campus/OU=Department of Computer Science/CN=Matt Bishop/**
  - **O: organization, OU: organizational unit, CN: common name.**

Chapter 13 REPRESENTING IDENTITY

- Certificate Authorities (CAs) vouch, at some level, for the identity of the principal to which the certificate is issued.
- Every CA has two policies controlling how it issues certificates.
  - A CA <u>authentication</u> policy describes the level of authentication required to identify the principal to whom the certificate is to be issued.
  - A CA <u>issuance</u> policy describes the principals to whom the CA will issue certificates.
- The difference between these policies:
  - The first establishes the level of proof of identity needed for the CA to accept the principal's claim of identity.
  - The second answers the question, "Given the identity of the principal, will the CA issue a certificate?"
- EXAMPLE: Verisign Corporation
  - In 1996, the corporation ran several CAs.
  - Each had its own policies of issuance and authentication for certificates.
- Individuals obtained certificates from one of three CAs.
  - <u>Class 1</u> CA: authenticated the individual's email address. This CA provided a certificate for sending and receiving email securely.
  - <u>Class 2</u> CA: required the individual to supply his real name and address (verified through an online database). This CA provided a certificate for online purchasing.
  - <u>Class 3</u> CA: required a back ground check from an investigative service. The certificate from this CA provided a higher level of assurance of identity than the other two certificates.

Chapter 13 REPRESENTING IDENTITY

- **CAs can issue certificates to organizations as well.**
- **EXAMLE:**
  - **The infrastructure organizes CAs into a hierarchical, tree-based structure.**
  - **Each node in the tree corresponds to a CA.**
  - **Consider a node that is the root of a subtree.**
  - **The CAs under that root are constrained by the policies of that root.**
  - **The subordinate nodes may issue certificates with more restrictive policies, and not with more liberal policies.**
- **Internet Policy Registration Authority (IPRA)**
  - **An X.509-compliant CA that is the top CA of the Internet certification hierarchy operated under the auspices of the ISOC [R1422].**
  - **The root of the tree is the Internet Policy Registration Authority (IPRA).**
  - **It sets policies that all subordinate CAs must follow.**
  - **It also certifies other CAs called policy certification authorities (PCAs).**
  - **Each PCA has its own issuance and authentication policies. However, thses policies must not conflict with the policies set by the IPRA.**
  - **The PCAs issue certificates to ordinary CAs, which can then issue certificates to individuals or organizations.**
  - **The IPRA and PCAs do not issue certificates to individuals or organizations.**
  - **All CAs, PCAs, and the IPRA have unique Distinguished Names.**

Chapter 13 REPRESENTING IDENTITY

- The elegance of this approach is twofold:
  - All PCA policies are public.
  - When a certificate is received, one can determine how much trust to place in the identity in the certificate (authentication policy) as well as the requirements that the holder had to meet to have the certificate issued (issuance policy).
  - Suppose the University of Valmont wished to establish a CA for both students and staff.
    - The requirements for certification for these groups are different.
    - Students must present valid registration cards to obtain certificates.
    - These certificates are low-assurance certificates because of the nature of the registration process.
    - Each certificate would be signed using the university's low-assurance certificate.
    - This certificate, in turn, is signed by a PCA that requires its subordinates CAs to make a good-faith effort to verify the identities of those to whom it issues certificates.
    - However, the university requires staff members to present proof of employment and fingerprints.
    - This provides a higher level of assurance of identity.
    - The university signs these certificates with its high-assurance certificate, which is obtained from a different PCA that requires the use of biometrics for verification of identity.

Chapter 13 REPRESENTING IDENTITY

- The certificates for <u>student John</u> and <u>professor Marsha</u> are both signed by the same organization.
- John's certificate is signed by the key corresponding to a low-assurance certificate because it is signed by the first PCA.
- Marsha's certificate is signed by the key corresponding to a high-assurance certificate because it is signed by the second PCA.
- If the policies of each PCA and also the CA are checked, the recipient of one of these certificates can tell what the policies of issuance and assurance are.
- A potential conflict: the CA has the same Distinguished Name for two different types of policies.
- EXAMPLE:
  - Marsha's certificate identifies her as being affiliated with the University of Valmont.
  - This type of certificate is called an <u>organizational certificate</u>.
  - The Internet infrastructure defines a second type of certificate: a <u>residential certificate</u> that identifies the principal's residential address.
  - Marsha has one of these certificates issued by the post office.
  - /C=US/SP=Louisiana/L=Valmont/PA=27 Russell Blvd./CN=Marsha/.

Chapter 13 REPRESENTING IDENTITY

- **The Meaning of the Identity**
  - **The authentication policy defines the way in which principals prove their identities.**
  - **Each CA has its own requirements, although they may be constrained by contractual requirements, such as with PCAs.**
  - **All rely on nonelectronic proofs of identity: biometrics (fingerprints), documents (driver's license, passports), or personal knowledge.**
  - **If any of these means are compromised, the CA may issue a certificate to the wrong person.**
    - **Each certificate is the binding of an external entity to a cryptographic key and a Distinguished Name.**
    - **If the issuer of the certificate is fooled, all who rely on that certificate may also be fooled.**
  - **EXAMPLE:**
    - **A government plans to require all citizens with a specific gene to register.**
    - **Anecdotal evidence suggests that people with that gene commit crimes slightly more than other people.**
    - **The government plans to make the law without publicity.**
    - **Aside from the civil liberties issues, there is no reputable scientific evidence to back up the belief.**
    - **A government employee decides to alert the media.**
    - **She believes that the government will deny the plan and change its approach to getting the law passed.**
    - **She feels that she may be fired if the government determines who she is.**
    - **Therefore, she decides to publicize the plans anonymously.**

Chapter 13 REPRESENTING IDENTITY

- Anonymous (or persona) certificates supply the requisite anonymity.
  - A CA issues a persona certificate under a policy that makes the Distinguished Name of the principal meaningless.
  - For example, a persona certificate with a principal Distinguished Name of /C=US/O=Microsoft Corp./CN=John Smith/ does not imply that the certificate was issued to someone named John Smith.
- EXAMPLE:
  - Our heroine obtains a persona certificate and sends a copy of the government's plan to the media, using email.
  - The government has some employees leak verifiably false information.
  - If the original whistleblower sends another message, it is less likely to be believed.
  - However, she does and uses the same certificate to authenticate the message.
  - Now, the media can check that the two messages came from the same source (or at least were signed with the same certificate).
  - The false messages were signed by different certificates.

Chapter 13 REPRESENTING IDENTITY

- **Trust**
  - The <u>goal</u> of certificates is to bind the correct identity to the public key.
  - When a user obtains a certificate, the issuer of that certificate is vouching, to some degree of certainty, that the identity corresponds to the principal owning the public key.
  - X.509v3 defines the degree of certainty in the policy of the CA that issues the certificate.
  - If a CA requires a passport as identification, the degree of certainty is <u>high</u>.
  - If a CA requires an unsworn statement of identity, the degree of certainty is <u>low</u>.
  - Even high-assurance CAs can be fooled. Passports can be stolen or forged.
  - EXAMPLE:
    - A CA requires a passport to issue a certificate.
    - The certificate will have as its Distinguished Name (DN) the name in the passport, the name of the country issuing the passport, and the passport number.
    - There are several points of trust in this policy:
      - The CA assumes that the passport is not forged and the name has not been altered.
      - The CA assumes that the country issuing the passport issued it to the person named in the passport.
      - The CA assumes that the individual presenting the passport is the individual to whom the passport was issued.
      - The users of the certificate assume that the CA has checked the passport and the individual using the passport to obtain a certificate.

Chapter 13 REPRESENTING IDENTITY

13.6 Identity on the Web

- Certificates are not ubiquitous on the Internet.
- There are other means to attach an identity to information.
- The Internet requires every host to have an address.
- The address may be fixed or may change.

- **Host Identity**
  - Host identity is intimately bound to networking.
  - If a host is not connected to a network, it can have any name because the name is used locally.
  - The ISO/OSI model is composed of 7 layers.
  - Each host, conceptually, has a principal at each layer that communicates with a peer on other hosts.
    - These principals communicate with principals at the same layer on other hosts.
    - Each principal on an individual host can have different names at each layer.
    - All names identify the same host, but each one refers to a particular context in which the host functions.
  - EXAMPLE:
    - A computer has an Ethernet MAC address of 00:05:02:6B:A8:A8:21, and an IP address of 192.168.35.89, and a host name of cherry.orchard.net.
    - At the data link level, the system is known by its Ethernet address.
    - At the network level, it is known by its IP address.
    - At the application level, it is known by its host name.
    - The system is also on an AppleTalk network, with an AppleTalk address of network 51, and node 235.

Chapter 13 REPRESENTING IDENTITY

- **Static and Dynamic Identifiers**
  - An identifier can be either static or dynamic.
  - A static identifier does not change over time.
  - A dynamic identifier changes either as a result of an event or over time.
  - Databases contain mappings between different names.
  - The best known of these is the Domain Name Service (DNS), which associates host names and IP addresses.
  - If there is no cryptographic authentication of hosts, weak authentication would be provided.
- **EXAMPLE:**
  - THE DNS contains forward records (which map host names into IP addresses), and reverse records (which map IP addresses into names).
  - A reverse domain lookup
    - occurs when a process extracts the IP address of its remote peer,
    - determines the associated host name (perhaps using the DNS),
    - then obtains the set of IP addresses associated with that host name (again, using the DNS).
    - if the IP address obtained from the peer matches any of the IP addresses associated with that host name, then the host name is accepted as the one obtained in the first lookup.
    - otherwise, the host name is rejected as untrusted.
    - all names identify the same host, but each one refers to a particular context in which the host functions.

Chapter 13 REPRESENTING IDENTITY

- **In the absence of cryptography, authentication using dynamic naming is different from authentication using static naming.**
  - **The main problem is that the association of the identity with a principal varies over time.**
  - **Therefore, any authentication based on the name must also account for the time.**
  - **For example, if the DNS record entries corresponding to the dynamic name are not updated whenever the name is reassigned, the reverse domain lookup method of authentication fails.**
- **The contrast between static and dynamic naming in authentication is worth noting.**
  - **The reverse domain lookup technique of authentication corresponds to checking a property of a principal (what it is) with static naming because the name is bound permanently to the principal.**
  - **However, that technique corresponds to checking a possession of a principal (what it has) with dynamic naming because the principal will relinquish that name at some point.**

Chapter 13 REPRESENTING IDENTITY

- Security Issues with the Domain Name Service
  - Understanding the centrality of trust in the databases that record associations of identity with principals is critical to understand the accuracy of the identity.
  - The DNS is an example of this.
  - The trustworthiness of the host name relies on the integrity of the DNS database.
  - If the association between a host name and an IP address can be corrupted, the identifier in question will be associated with the wrong host.
  - Bellovin and Schuba discuss several attacks on the DNS.
    - The goal of these attacks is to cause a victim to associate incorrectly a particular IP address with a host name.
    - They assume the attacker is able to control the responses from an authoritative DNS.
    - Control means that the attacker has control over the name server or can intercept queries to that server and return its own responses.
    - Two attacks:
      - The attacker can change the records associating the IP address with the host name, so that a query for one returns an incorrect answer for the other.
      - Cache poisoning: relies on the ability of a server to add extra DNS records to the answer to a query. The extra records added give incorrect association information.

Chapter 13 REPRESENTING IDENTITY

- **State and Cookies**
  - **A <u>cookie</u> is a token that contains information about the state of a transaction on a network.**
  - **Although the transaction can be any client-server interaction, the term cookie is most widely used in reference to interactions between Web browsers and clients.**
  - **The cookies minimize the storage requirements of the server and put the burden of maintaining required information on the client.**
- **The cookies consist of several values:**
  - **The <u>name</u> and <u>value</u> are encoded into the cookie and represent the state.**
  - **The <u>expires</u> field indicates when the cookie is valid. Expired cookies are discarded. If the field is not present, the cookies will be deleted at the end of the session.**
  - **The <u>domain</u> states the domain for which the cookie is intended. It consists of the last *n* fields of the domain name of a server. The cookie will be sent to servers in that domain. For example,** `domain=.adv.com` **specifies that the cookie is to be sent to any requesting server in the** `adv.com` **domain.**
  - **The <u>path</u> further restricts the dissemination of the cookie. When a Web server requests a cookie, it provides a domain (its own). Cookies that match that domain may be sent to the server.**
  - **If the <u>secure</u> field is set, the cookie will be sent only over secured connections (i.e., "https" – http over Secure Sockets Layer (SSL)).**

Chapter 13 REPRESENTING IDENTITY

- **EXAMPLE:**
  - Caroline logs in to a Web server (www.books.com) used to sell books.
  - She selects 2 books to buy and adds them to her shopping cart.
  - The Web server sends her a cookie with key "bought" and value "BK=234&BK=8763."
  - The domain for the cookie is ".books.com."
  - The expiration field is omitted.
  - When Caroline goes to the page to pay for the books, the server asks for the cookie "bought" belonging to the domain ".books.com."
  - The server sees that Caroline wants to buy books numbered 234 and 8763.
  - If Caroline terminates the session (e.g., by exiting her browser), the cookie would be deleted and no record would exist of the books she thought about buying.
- **Cookies can contains authentication information, both user-related and host-related.**
  - Using cookies for authentication are treated as tokens supplied by the browser to validate an identity.
  - If the interactions with the server are sensitive, the confidentiality of these cookies may be critical.

Chapter 13 REPRESENTING IDENTITY

- **Anonymity on the Web**
  - On the Internet, identification arises from associating a particular host with a connection or message.
  - The recipient can determine the origin from the incoming packet.
  - If only one person is using the originating host, and the address is not spoofed, it is easy to identify the sender with a very high degree of accuracy.
  - An <u>anonymizer</u> is a site that hides the origins of connections.
    - It functions as a proxy server.
    - A user connects to the anonymizer and tells its destination.
    - The anonymizer makes the connection. The destination host sees only the anonymizer.
    - The anonymizer forwards traffic in both directions.
  - Anonymizers work primarily on email and http traffic.
- **EXAMPLE: The Finnish anonymizer anon.penet.fi**
  - The host anon.penet.fi offered an anonymous email service.
  - One would send a letter to it, naming another destination.
  - The anonymizer would strip off the headers, assign an anonymous ID (e.g., anon374) to the letter, and record the sender and the associated anonymous ID in a database.
  - The letter would then be delivered to its destination, as though user anon374 at anon.penet.fi had sent it.
  - The recipient could not tell the original sender from the letter.
  - They would reply to the letter by sending the reply to anon374 at anon.penet.fi.
  - This letter would be anonymized in the same way the original letter was anonymized.
  - Finally, the letter would be forwarded to the real email address corresponding to anon374.

Chapter 13 REPRESENTING IDENTITY

- Anonymity on the Web
  - A <u>pseudo–anonymous</u> (pseudonymous) remailer is a remailer that replaces the originating electronic mail addresses (and associated data) of messages it receives before it forwards them, but keeps mappings of the anonymous identities and the associated origins.
    - The problem is that the binding between anonymous address and the real address is known somewhere. If that point can be made to reveal the association, anonymity ceases to exist.
  - A <u>Cypherpunk</u> (type 1) remailer is a remailer that deletes the header of an incoming message and forwards the remainder to its destination.
    - Unlike a pseudo–anonymous remailer, no record of the association between the originating address and the remailer address is kept.
    - One cannot trace the message by mapping the remailer's user name to an electronic mail address.
  - A <u>Mixmaster</u> (type 2) remailer is a Cypherpunk remailer that handles only enciphered messages and that pads or fragments messages to a fixed size before sending them.
    - This remailer prevents the attacks for type 1 remailer.
    - The contents of the incoming and outgoing messages cannot be matched because everything is enciphered.
    - Traffic analysis based on size is not possible because all messages have the same size.
    - All messages are uniquely numbered. Replay attacks, therefore, are not possible.