

Lecture 11

October 14, 2022

Salting

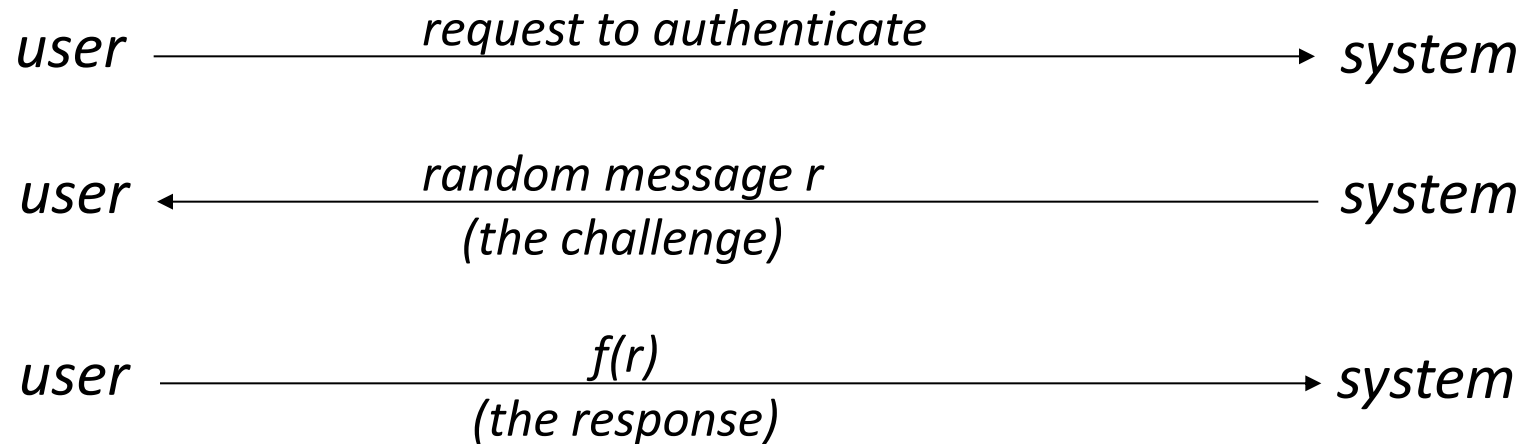
- Goal: slow dictionary attacks
- Method: perturb hash function so that:
 - Parameter controls *which* hash function is used
 - Parameter differs for each password
 - So given n password hashes, and therefore n salts, need to hash guess n

Example

- password: hello,there!1
- stored version (no line breaks in password file):
\$6\$1BSRcuVLmWnV6LET\$dJf2kPCM9Pj0yEvxAtyp8ZJIcgt
NY7QEY4J/nDc8iYx9NR610XxCFI7gewN2yduSMu2z4BOAem
TOVAn/R0yQV/
- interpretation (\$ separates parts of the password):
 - \$6\$ indicates modular password format and hashing algorithm
 - SHA-512 (1=MD5, 2=Blowfish, 3=NT-Hash [doesn't use salt, use discouraged, 5=SHA-256])
 - 1BSRcuVLmWnV6LET is salt
 - dJf2kPCM9Pj0yEvxAtyp8ZJIcgtNY7QEY4J/nDc8iYx9NR610XxCFI7gewN2yduSMu2z4BOAemTOVAn/R0yQV/ is hash of password and salt

Challenge-Response

User, system share a secret function f (in practice, f is a known function with unknown parameters, such as a cryptographic key)



One-Time Passwords

- Password that can be used exactly *once*
 - After use, it is immediately invalidated
- Challenge-response mechanism
 - Challenge is number of authentications; response is password for that particular number
- Problems
 - Synchronization of user, system
 - Generation of good random passwords
 - Password distribution problem

S/Key

- One-time password scheme based on idea of Lamport
- h one-way hash function (MD5 or SHA-1, for example)
- User chooses initial seed k
- System calculates:

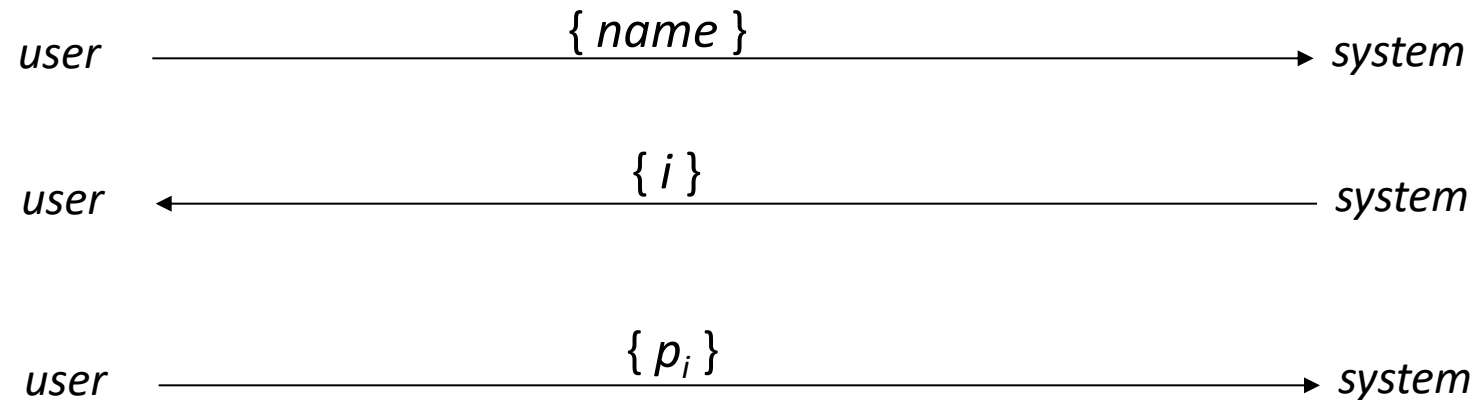
$$h(k) = k_1, h(k_1) = k_2, \dots, h(k_{n-1}) = k_n$$

- Passwords are reverse order:

$$p_1 = k_n, p_2 = k_{n-1}, \dots, p_{n-1} = k_2, p_n = k_1$$

S/Key Protocol

System stores maximum number of authentications n , number of next authentication i , last correctly supplied password p_{i-1} .



System computes $h(p_i) = h(k_{n-i+1}) = k_{n-i} = p_{i-1}$. If match with what is stored, system replaces p_{i-1} with p_i and increments i .

Hardware Support

- Token-based
 - Used to compute response to challenge
 - May encipher or hash challenge
 - May require PIN from user
- Temporally-based
 - Every minute (or so) different number shown
 - Computer knows what number to expect when
 - User enters number and fixed password

Biometrics

- Automated measurement of biological, behavioral features that identify a person
 - Fingerprints: optical or electrical techniques
 - Voices: speaker verification or recognition
 - Eyes: patterns in irises unique
 - Faces: image, or specific characteristics like distance from nose to chin
 - Keystroke dynamics: believed to be unique

Location

- If you know where user is, validate identity by seeing if person is where the user is
 - Requires a device saying where the user is, like a smart phone

Multi-Factor Authentication

- Example: “where you are” also requires entity to have LSS and GPS, so also “what you have”
- Can assign different methods to different tasks
 - As users perform more and more sensitive tasks, must authenticate in more and more ways (presumably, more stringently) File describes authentication required
 - Also includes controls on access (time of day, *etc.*), resources, and requests to change passwords
 - Pluggable Authentication Modules

Multi-Factor Authentication

- Example: “where you are” also requires entity to have LSS and GPS, so also “what you have”
- Can assign different methods to different tasks
 - As users perform more and more sensitive tasks, must authenticate in more and more ways (presumably, more stringently) File describes authentication required
 - Also includes controls on access (time of day, *etc.*), resources, and requests to change passwords
 - Pluggable Authentication Modules

Identity

- *Principal*: a unique entity
- *Identity*: specifies a principal
- *Authentication*: binding of a principal to a representation of identity internal to the system
 - All access, resource allocation decisions assume binding is correct

Files and Objects

- Identity depends on system containing object
- Different names for one object
 - Human use, *eg.* file name
 - Process use, *eg.* file descriptor or handle
 - Kernel use, *eg.* file allocation table entry, inode

More Names

- Different names for one context
 - Human: aliases, relative *vs.* absolute path names
 - Kernel: deleting a file identified by name can mean two things:
 - Delete the object that the name identifies
 - Delete the name given, and do not delete actual object until *all* names have been deleted
- Semantics of names may differ

Example: Names and Descriptors

- Interpretation of UNIX file name
 - Kernel maps name into an inode using iterative procedure
 - Same name can refer to different objects at different times without being deallocated
 - Causes race conditions
- Interpretation of UNIX file descriptor
 - Refers to a specific inode
 - Refers to same inode from creation to deallocation

Example: Different Systems

- Object name must encode location or pointer to location
 - *SSH* style: *host:object*
 - URLs: *protocol://host/object*
- Need not name actual object
 - *SSH* style may name pointer (link) to actual object
 - URL may forward to another host

Users

- Exact representation tied to system
- Example: UNIX/Linux systems
 - Login name: used to log in to system
 - Logging usually uses this name
 - User identification number (UID): unique integer assigned to user
 - Kernel uses UID to identify users
 - One UID per login name, but multiple login names may have a common UID

Multiple Identities

- UNIX systems again
 - Real UID: user identity at login, but changeable
 - Effective UID: user identity used for access control
 - Setuid changes effective UID
 - Saved UID: UID before last change of UID
 - Used to implement least privilege
 - Work with privileges, drop them, reclaim them later
 - Audit/Login UID: user identity used to track original UID
 - Cannot be altered; used to tie actions to login identity

Groups

- Used to share access privileges
- First model: alias for set of principals
 - Processes assigned to groups
 - Processes stay in those groups for their lifetime
- Second model: principals can change groups
 - Rights due to old group discarded; rights due to new group added

Roles

- Group with membership tied to function
 - Rights given are consistent with rights needed to perform function
- Uses second model of groups
- Example: DG/UX
 - User *root* does not have administration functionality
 - System administrator privileges are in *sysadmin* role
 - Network administration privileges are in *netadmin* role
 - Users can assume either role as needed

Naming and Certificates

- Certificates issued to a principal
 - Principal uniquely identified to avoid confusion
- Problem: names may be ambiguous
 - Does the name “Matt Bishop” refer to:
 - The author of this book?
 - A programmer in Australia?
 - A stock car driver in Muncie, Indiana?
 - Someone else who was named “Matt Bishop”

Disambiguating Identity

- Include ancillary information in names
 - Enough to identify principal uniquely
 - X.509v4 Distinguished Names do this
- Example: X.509v4 Distinguished Names
 - /O=University of California/OU=Davis campus/OU=Department of Computer Science/CN=Matt Bishop/
refers to the Matt Bishop (CN is *common name*) in the Department of Computer Science (OU is *organizational unit*) on the Davis Campus of the University of California (O is *organization*)

CAs and Policies

- Matt Bishop wants a certificate from Certs-from-Us
 - How does Certs-from-Us know this is “Matt Bishop”?
 - *CA's authentication policy* says what type and strength of authentication is needed to identify Matt Bishop to satisfy the CA that this is, in fact, Matt Bishop
 - Will Certs-from-Us issue this “Matt Bishop” a certificate once he is suitably authenticated?
 - *CA's issuance policy* says to which principals the CA will issue certificates

Example: Verisign CAs

- Class 1 CA issued certificates to individuals
 - Authenticated principal by email address
 - Idea: certificate used for sending, receiving email with various security services at that address
- Class 2 CA issued certificates to individuals
 - Authenticated by verifying user-supplied real name and address through an online database
 - Idea: certificate used for online purchasing

Example: Verisign CAs

- Class 3 CA issued certificates to individuals
 - Authenticated by background check from investigative service
 - Idea: higher level of assurance of identity than Class 1 and Class 2 CAs
- Fourth CA issued certificates to web servers
 - Same authentication policy as Class 3 CA
 - Idea: consumers using these sites had high degree of assurance the web site was not spoofed

Registration Authority

- Third party delegated by CA the authority to check data to be put into certificate
 - This includes identity
- RA determines whether CA's requirements are met
- If so, then it informs CA to issue certificates

Internet Certification Hierarchy

- Tree structured arrangement of CAs
 - Root is *Internet Policy Registration Authority*, or IPRA
 - Sets policies all subordinate CAs must follow
 - Certifies subordinate CAs (called *policy certification authorities*, or PCAs), each of which has own authentication, issuance policies
 - Does not issue certificates to individuals or organizations other than subordinate CAs
 - PCAs issue certificates to ordinary CAs
 - Does not issue certificates to individuals or organizations other than subordinate CAs
 - CAs issue certificates to organizations or individuals

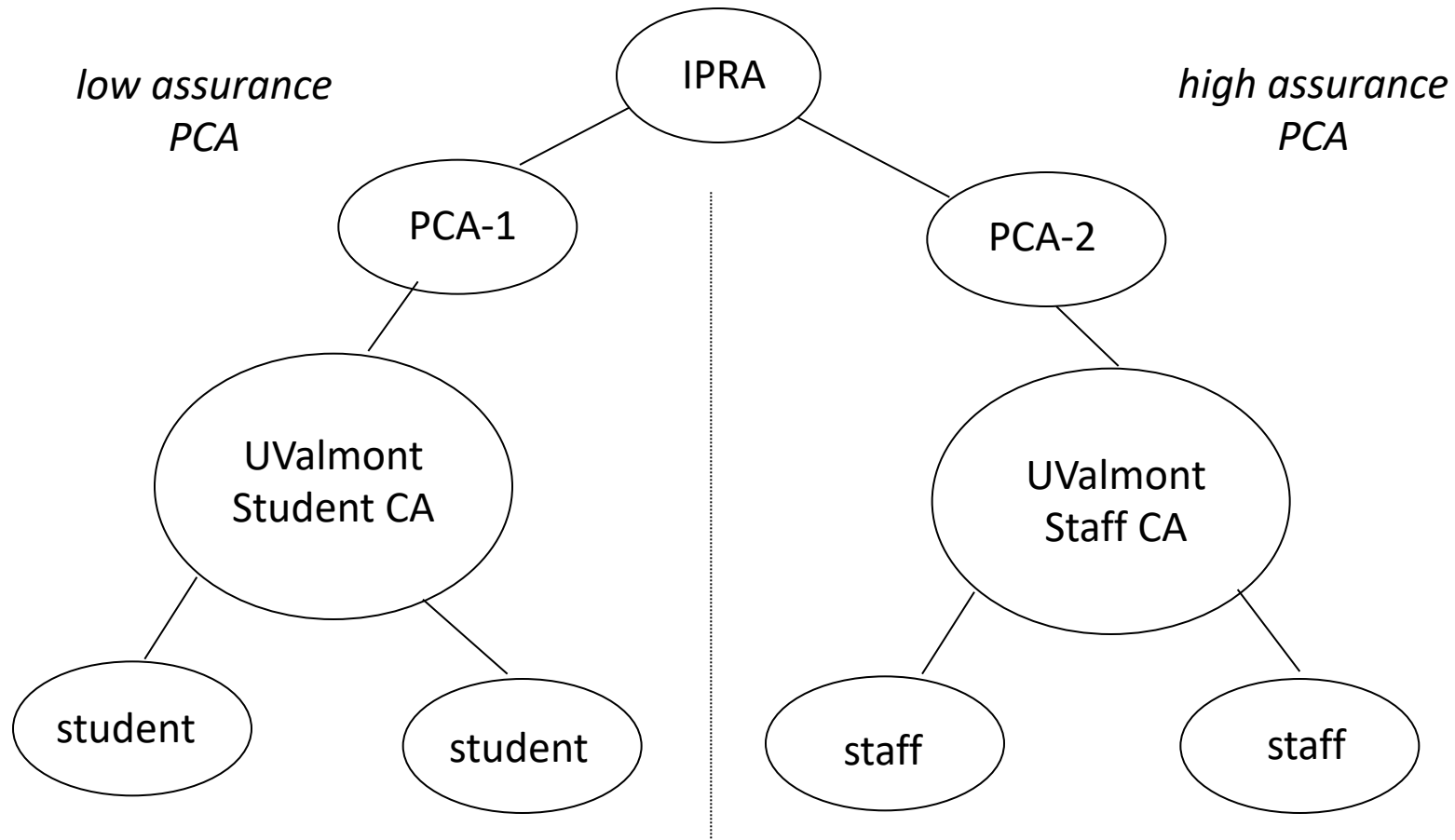
Example

- University of Valmont issues certificates to students, staff
 - Students must present valid reg cards (considered low assurance)
 - Staff must present proof of employment and fingerprints, which are compared to those taken when staff member hired (considered high assurance)

UValmont and PCAs

- First PCA: requires subordinate CAs to make good-faith effort to verify identities of principals to whom it issues certificates
 - Student authentication requirements meet this
- Second PCA: requires use of biometrics to verify identity
 - Student authentication requirements do not meet this
 - Staff authentication requirements do meet this
- UValmont establishes two CAs, one under each PCA above

UValmont and Certification Hierarchy



Certificate Differences

- Student, staff certificates signed using different private keys (for different CAs)
 - Student's signed by key corresponding to low assurance certificate signed by first PCA
 - Staff's signed by key corresponding to high assurance certificate signed by second PCA
- To see what policy used to authenticate:
 - Determine CA signing certificate, check its policy
 - Also go to PCA that signed CA's certificate
 - CAs are restricted by PCA's policy, but CA can restrict itself further

Types of Certificates

- Organizational certificate

- Issued based on principal's affiliation with organization
- Example Distinguished Name

/O=University of Valmont/OU=Computer Science Department/CN=Marsha Merteuille/

- Residential certificate

- Issued based on where principal lives
- No affiliation with organization implied
- Example Distinguished Name

/C=US/SP=Louisiana/L=Valmont/PA=1 Express Way/CN=Marsha Merteuille/

Certificates for Roles

- Certificate tied to a role
- Example
 - UValmont wants comptroller to have a certificate
 - This way, she can sign contracts and documents digitally
 - Distinguished Name
/O=University of Valmont/OU=Office of the Big Bucks/RN=Comptroller/
where “RN” is *role name*; note the individual using the certificate is not named, so no CN

Certificate Principal Identifiers

- Need not be Distinguished Names
 - Example: PGP certificates usually have email addresses, not Distinguished Names
- Permits ambiguity, so the user of the certificate may not be sure to whom it refers
 - Email addresses change often, particularly if work email addresses used
- Problem: how do you prevent naming conflicts?

Naming Conflicts

- X.509, PGP silent
 - Assume CAs will prevent name conflicts as follows
 - No two distinct CAs have the same Distinguished Name
 - No two principals have certificates issued containing the same Distinguished Name by a single CA

Internet Certification Hierarchy

- In theory, no naming collisions
 - IPRA requires each PCA to have a unique Distinguished Name
 - No PCA may certify two distinct CAs with same Distinguished Name
- In practice, considerable confusion possible!

Example Collision

John Smith, John Smith Jr. live at same address

- John Smith Jr. applies for residential certificate from Certs-from-Us, getting the DN of:

/C=US/SP=Maine/L=Portland/PA=1 First Ave./CN=John Smith/

- Now his father applies for residential certificate from Quick-Certs, getting DN of:

/C=US/SP=Maine/L=Portland/PA=1 First Ave./CN=John Smith/

because Quick-Certs has no way of knowing that DN is taken

Solutions

- Organizational certificates

- All CA DNs must be superior to that of the principal

- Example: for Marsha Merteuille's DN:

/O=University of Valmont/OU=Computer Science Department/CN=Marsha Merteuille/

DN of the CA must be either:

/O=University of Valmont/

(the issuer being the University) or

/O=University of Valmont/OU=Computer Science Department/

(the issuer being the Department)

Solutions

- Residential certificates

- DN collisions explicitly allowed (in above example, no way to force disambiguation)

/C=US/SP=Maine/L=Portland/PA=1 First Ave./CN=John Smith/

Unless names of individuals are different, how can you force different names in the certificates?

Related Problem

- Single CA issues two types of certificates under two different PCAs
- Example
 - UValmont issues both low assurance, high assurance certificates under two different PCAs
 - How does validator know under which PCA the certificate was issued?
 - Reflects on assurance of the identity of the principal to whom certificate was issued

Solution

- CA Distinguished Names need *not* be unique
- CA (Distinguished Name, public key) pair *must* be unique
- Example
 - In earlier UValmont example, student validation required using first PCA's public key; validation using second PCA's public key would fail
 - Keys used to sign certificate indicate the PCA, and the policy, under which certificate is issued

Meaning of Identity

- Authentication validates identity
 - CA specifies type of authentication
 - If incorrect, CA may misidentify entity unintentionally
- Certificate binds *external* identity to crypto key and Distinguished Name
 - Need confidentiality, integrity, anonymity
 - Recipient knows same entity sent all messages, but *not* who that entity is

Persona Certificate

- Certificate with meaningless Distinguished Name
 - If DN is
/C=US/O=Microsoft Corp./CN=Bill Gates/
the real subject may not (or may) be Mr. Gates
 - Issued by CAs with persona policies under a PCA with policy that supports this
- PGP certificates can use any name, so provide this implicitly

Example

- Government requires all citizens with gene X to register
 - Anecdotal evidence people with this gene become criminals with probability 0.5.
 - Law to be made quietly, as no scientific evidence supports this, and government wants no civil rights fuss
- Government employee wants to alert media
 - Government will deny plan, change approach
 - Government employee will be fired, prosecuted
- Must notify media anonymously

Example

- Employee gets persona certificate, sends copy of plan to media
 - Media knows message unchanged during transit, but not who sent it
 - Government denies plan, changes it
- Employee sends copy of new plan signed using same certificate
 - Media can tell it's from original whistleblower
 - Media cannot track back whom that whistleblower is

Trust

- Goal of certificate: bind correct identity to DN
- Question: what is degree of assurance?
- X.509v4, certificate hierarchy
 - Depends on policy of CA issuing certificate
 - Depends on how well CA follows that policy
 - Depends on how easy the required authentication can be spoofed
- Really, estimate based on the above factors

Example: Passport Required

- DN has name on passport, number and issuer of passport
- What are points of trust?
 - Passport not forged and name on it not altered
 - Passport issued to person named in passport
 - Person presenting passport is person to whom it was issued
 - CA has checked passport and individual using passport

PGP Certificates

- Level of trust in signature field
- Four levels
 - Generic (no trust assertions made)
 - Persona (no verification)
 - Casual (some verification)
 - Positive (substantial verification)
- What do these mean?
 - Meaning not given by OpenPGP standard
 - Signer determines what level to use
 - Casual to one signer may be positive to another