Lecture 5 - 09/20/2019

Integrity Policies

- Focus on Integrity rather than confidentiality
- Note that mots commercial financial, and industrial firms are most concerned with accuracy than disclosure

Three Primary principles of operation:

Seperation of duty

- Two or more steps are involved in a critical function.
- You should also have different people performing these steps

Separation of function.

- Avoid risk of contamination. Prod vs dev environment.

Auditing

- Keep track of all procedures that go on within your system
- Accountability and system recovery

Biba Integrity Model

(MITRE Tech. Report, 1977)

Essentially the dual of the Bell-LaPadula confidentiality model.

Integrity levels

Higher integrity levels means there is a higher level of trust that that data is correct.

A system can write to an object if integrity level of that system is greater than the integrity level of the object.	$i(s) \geq i(o)$
A system can read to an object if the integrity level of that system is less than the integrity level of the object.	$i(s) \leq i(o)$
A program S1 can call or run a program S2 iff	$i(S1) \geq i(S2)$

Integrity Systems follow a “low watermark” philosophy:
The integrity of an object is the lowest level of all the objects that contributed to its creation.
(A system is only as strong as its weakest link)

Clark-Wilson Integrity Model

(IEEE Symp. Sec. & Priv., 1987)

Unlike previous integrity models, the focus is not on “Write” and “read” and integrity labels on data, but thee “transaction” as the basic unit (which more realistically models commercial operations)

If a transaction fails part way through, you must be able to roll back to before that transaction started.

“Well formed transactions”

- Captured in “enforcement rules”
- Internal verification that everything is correct

Look at integrity of the transactions themselves

- Captured in “Certification rules”
- Who examines and certifies that the transactions are performed correctly?
- Typically this will be an outside organization instead of internal verification

Clark-wilson appears to be more appropriate for commercial settings

Hybrid Policies

(Stallings & Brown, Chapter 13)

- Some combination of confidentiality, integrity, and availability

Four examples: Chinese Wall, Clinical Information, ORCON,

ORCON

Originator Controlled Access Controlled

- Originators of documents retain control over them even after those documents are disseminated.

Very difficult to achieve in practice, especially if the objects are digital.

- Digital Rights Management (DRM) is precisely an instance of the ORCON model.
- Digital watermarking can help

RBAC

Role Based Access Control

The ability, or need, to access information may depend on job function. The idea is your "job function" is your role.

R = role (collection of job functions)
S = subject
T = transaction

trans(r)	= a set of transactions authorized to a role r
authr(s)	= a set of roles that s is authorized to assume
actr(s)	= the role that s is current possessing
canexec(s, t)	= is TRUE iff s can execute t at the current time.

axioms
$(\forall\ s) \mid (\forall\ t) \mid [canext(s,t) \rightarrow actr(s) =/= t\ ]$
$(\forall\ s) \mid [\ act(s) < authr(s)\ ]$
$(\forall\ s) \mid (\forall\ t)\ \mid [canext(s,t) \rightarrow t \in trans(actr(s))\ ]$

$RBAC_0$: is base model
$RBAC_1$: is role hierarchy (role inheritance)
$RBAC_2$: has constraints, relationships between roles
$RBAC_3$: having roles and constraints.

Policy Composition

Often there are different policy-making units in an organization that will need to be combined.

How do we combine multiple policies into a single policy?

- Access allowed by security policies on one component must be allowed by composition of components [autonomy]
- Access forbidden by … forbidden .. [security]

Security Policy:

1. Important
2. Hard
3. Only step #1

Cryptography

Encryption/Decryption

There are two main classes of algorithms that are used for encryption/decryption

- Symmetric -> Same key is used for encryption and decryption
    - Much faster than Asymmetric key cryptography
- Asymmetric -> There is a public key and a private key, the public key is shared and used for encryption.
    - Key management and distribution are much easier in an asymmetrical environment.

<u>Symmetric</u>	<u>Asymmetric</u>
K is a key that both Bob and Alice have	PU_{Bob}
	PR_{Bob}
$C = E_K(m)$	$C = E_{PUbob}(m)$
$M = D_K(c)$	$M = D_{PRbob}(c)$

Signatures

Bob has a message M and he does:
$S = D_{PRIV}(m)$

Bob uses his private key to run the decryption algorithm on a message. This is a signature. Alice can then use Bob's public key to run the encryption math on that signature and if that comes out the same it must have come from bob.

Algorithms

Symmetric

AES

- Is everywhere

DES

- Old but can still be used

3DES

- Triple DES

Blowfish, CAST-128, RC2 …

Asymmetric algorithms

RSA

- The best

ElGamal

- Less popular

DSA

- For signatures

ECDSA

- Elliptic curve version of DSA

Hashing Algorithms

SHA-1, SHA-2, SHA-3

- Secure Hash Algorithm

MD5

Mac Algorithm

HMAC, ASEMAC

Key Agreement

D-H

- Diffie-helman

Hash Algorithms

- Arbitrary Size input, Fixed size output
- One-way
    - f`(f(x)) is not possible
- Collision resistant

MAC is similar to a hash function, but it uses a key.