Föreläsning 1 (kap. 3)

Three classes of protection measures -

Prevention: take measures that prevent your assets frombeing damaged.

Detection: take measures so that you can detect when, how,and by whom an asset has been damaged.

Reaction: take measures so that you can recover your assets or to recover from a damage to your assets.

Definitions–

Confidentiality

- Prevent unauthorized disclosure of information
- Related to the reading of data
  - More generally – the learning of data
- Two aspects
  - Privacy: protection of personal data.
  - Secrecy: protection of data belonging to an organization.
- Achieved by encryption, access control
- Confidentiality also applies to existence of data

Integrity – make sure that everything is as it is supposed to be

- prevent unauthorized modification of information.
- Related to the writing of data
- No user of the system, even if authorized, may be permitted to modify data items in such a way that assets or accounting records of the company are lost or corrupted.
- The state that exists when computerized data is the same as that in the source document and has not been exposed to accidental or malicious alteration or destruction.
- Integrity can be achieved by CRCs, hash functions, Message Authentication Codes (MACs), Digital Signatures

Availability: prevent unauthorized with-holding of information or resources.

- The property that a product’s services are accessible when needed and without undue delay.

- The property of being accessible and usable upon demand by an authorized entity.
- Denial of Service (DoS): The prevention of authorized access of resources or the delaying of time-critical operations.

Accountability and Authentication – Users should be held responsible for their actions

- Confidentiality, integrity and availability focus on prevention
    - But we can not prevent authorized actions
- Audit information must be selectively kept and protected so that actions affecting security can be traced to the responsible party.
- Users are *identified* and *authenticated* to have a basis for access control decisions.
- The security system keeps an audit log (audit trail) of security relevant events to detect and investigate intrusions.

Nonrepudiation

- A way to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message
- Divided into
    - *Nonrepudiation of origin*: Ensures that the originator of information cannot successfully deny having sent the information.
    - Nonrepudiation of receipt: Ensures that the recipient of information cannot successfully deny having received the information.
- Can be achieved by digital signatures

Data vs Information

- Data are physical phenomena chosen by convention to represent certain aspects of our conceptual and real world. The meanings we assign to data are called information. Data is used to transmit and store information and to derive new information by manipulating the data according to formal rules.
- Information and data correspond to the two ends of the man- machine scale.

Föreläsning 4 (kap. 4)

Spoofing attacks

A spoofing attack is a situation in which one person or program successfully masquerades as another by falsifying data and thereby gaining an illegitimate advantage.

It could be prevented by with a spoofing program

- Displaying the number of faild logins, this could indicate the user that an attack has happend.
- Or use a trusted path that could guarantee that the user communicates with the operating system and not with a spoofing program. (Windows uses the CTRL+ALT+DEL command, the user should press such a secure attention key when starting a session, even when the logon screen is already displayed.
- Mutual authentication, the system could be required to authenticate itself to the user.

Password recovery using brute force, dictionary and time-memory-tradeoff attacks.

Brute force

Such an attack might be utilized when it is not possible to take advantage of other weaknesses in an encryption system (if any exist) that would make the task easier. It involves systematically checking all possible keys until the correct key is found. In the worst case, this would involve traversing the entire search space.

Dictionary

A dictionary attack uses a targeted technique of successively trying all the words in an exhaustive list called a dictionary (from a pre-arranged list of values). In contrast with a brute force attack, where a large proportion key space is searched systematically, a dictionary attack tries only those possibilities which are most likely to succeed, typically derived from a list of words for example a dictionary (hence the phrase dictionary attack) or a bible etc. Generally, dictionary attacks succeed because many people have a tendency to choose passwords which are short (7 characters or fewer), single words found in dictionaries or simple, easily-predicted variations on words, such as appending a digit.

Time–memory tradeoff

Time–memory tradeoff is a situation where the memory use can be reduced at the cost of slower program execution (and, conversely, the computation time can be reduced at the cost of increased memory use).(Behövs lite mer info)

Rainbow tables

A rainbow table is a precomputed table for reversing cryptographic hash functions, usually for cracking password hashes. Tables are usually used in recovering the plaintext password, up to a certain length consisting of a limited set of characters. It is a form of time-memory tradeoff, using less CPU at the cost of more storage.

Proper key derivation functions employ a salt to make this attack infeasible.(Behövs lite mer info)

Usage of salts in password hashing

When a password is encrypted for storage additional information, the salt, is appended to the password before encryption. The salt is then stored with the encrypted password. If two users have the same password, they will therefore have different entries in the file of encrypted passwords which slows down dictionary attacks.

How can the system help protecting against password compromise?

Biometric systems, the terms FRR, FAR and EER and how they relate, as- suming FTA is zero.

Biometric identifiers are the distinctive, measurable characteristics used to identify individuals.

Technology analysis of a biometric scheme is based on given databases of biometric samples. This analysis measures the performance of the algorithms extracting and comparing biometric characteristics. By setting the threshold for the matching algoritm, we can trade off a lower false match rate (FMR).

FMR = number of successful false matches/number of attempted false matches

Against a higher false non-match rate (FNMR)

FNMR = number of rejected genuine matches/number of attempted genuine matches

The designers of the systems have to find a right balance between those two errors. The right balance depends very much on the application. The equal error rate (EER) is given by the threshold where FMR and FNMR are equal.

The false accept rate (FAR) for the entire biometric scheme is then

FAR=FMR*(1-FTA).

Föreläsning 5 (kap. 5-6)

Discretionary vs. mandatory access control.

Discretionary access control – The owner of an object decides the access rights Mandatory access control–The system decides the access rights

Access control matrices can be implemented either as capabilities or as ACLs (Access Control List).

- Capabilities: Store access rights with *subject* (in a token that specifies this subject's access rights)
    - Difficult to determine who has access to a given object
- Access control list: Stores access rights with object (as a list with the object itself)

- Difficult to get an overview of an individual user's permissions

Powersets m.m -.-

Concept of reference monitor

The reference monitor is responsible for checking the access and it is not possible to get access to anything without involving the reference monitor.

Behövs mer?

Reference Monitor: An abstract machinery that controls all access to objects .

Security kernel: the hardware, software, firmware, that implements the reference monitor concept.

Trusted Computing Base (TCB): The set of all protection mechanisms enforcing a security policy.

The core requierements in the implementation of a reference monitor are:

- The reference validation mechanism must be tamper proof
- The reference validation mechanism must always be invoked.
- The reference validation mechanism must be small enought to be subject to analysis and tests to be sure that it is correct.

The reference monitors can practically be placed anywhere in a layered architecture. Examples:

- In hardware
- In the operating system kernel
- In the operating system (access control in Unix and Windows)
- In the services layer (Java Virtual Machine (JVM)).
- In the application

Reference Monitor Concept: Look at the last page from lecture5. IMPORTANT!

Controlled invocation

A user wants to execute an operation requiring supervisor mode. But you don't want to give the user all privileges associated with this mode without any control of what the user does. So the system only performs a predefined set of operations in supervisor mode, then returns to user mode before giving control back to the user

Föreläsning 7 (kap. 7)

Difference between real and effective UID in Unix

- **Real user ID – The ID of the logged in principal**
    - Can only be changed by root (effective user ID = 0) → this is how login work.
- **Effective user ID – The ID used for access control**
    - Can be changed by root (effective user ID = 0) to anything
        - Used by processes with effective user ID = 0 when they temporarily access files as a less privileged user
    - Can be changed by anyone (any effective user ID) to real user ID
        - This process has to be able to get back to effective user ID = 0

Difference between crypt, MD5 crypt and bcrypt in terms of performance and security. (No details of the algorithms or exactly where they are used).

Traditional crypt: Based on the DES algorithm (slightly modified). The passwords can be up to 8 characters and the salt used is 12 bits. Not good with fast algorithms for passwords (only 25 iterations). Problems: Short passwords, short salts and constant cost.

MD5 crypt: It avoids export restrictions and allow longer passwords (up to 2^64 bits). The algorithm uses 1000 iterations which makes it slow. The salt can be between 12-48 bits. The problem is that it has constant cost.

Bcrypt: It is based on block cipher blowfish. The password can be up to 72 characters and it uses 128 bit random salt. It uses an internal loop with variable cost. Has no problems (Perfect maybe? :))

Purpose of /etc/shadow file

Systems administrators can reduce the likelihood of brute force attacks by making the list of hashed passwords unreadable by unprivileged users. The obvious way to do this is to make the passwd database itself readable only by the root user. However, this would restrict access to other data in the file such as username-to-userid mappings, which would break many existing utilities and provisions. One solution is a "shadow" password file to hold the password hashes separate from the other data in the world-readable *passwd* file. For local files, this is /etc/shadow which are readable only by the root.

How access control works in Unix (read,write,execute,owner,group,other)

Three categories: User (owner), Group, Other (world)

Three access rights: Read, Write, Execute

- **Read** = list the directory
- **Write** = Delete, rename and insert files in directory
- **Execute** = access directory and access files in directory

The permission bits a checked in the following order:

1. Owner
2. Group
3. Other

If owner = r and other = rw then owner has no write permission

Setuid in Unix

Unix requires superuser privilege to execute certain operating system functions, but users should not be given superuser status. The solution of this in UNIX are setUID programs. Such programs run with the effective user ID of their owner (usually root), which give the user temporary or restricted access to files not normally accessible to other users.

You do not have to remember what is in an *inode*, but if you see an *inode* you should be able to locate the important information in it.

Bra bild på föreläsning.

Each file entry in a directory is a pointer to a data structure called an inode. It stores file information and the directory contains the filename and inode number. For information on how to find important information in the file see Lecture7 and page 113-114 in the text book.

Use of umask in Unix.

The unmask is three-digit octal number specifying the rights that should be withheld (Control default permissions, stored in /etc/profile)

Umask tells which permissions to *exclude* by default

Usage and potential problems with searchpath in Unix

Unix users interact with the operating system through a shell. A user can run a program by just typing its name without specifying the full pathname that gives the location of the program within the file system. The shell will then follow a searchpath specified by the PATH environment variable given in the .profile file in the user's home directory. When it finds the program the search will stop and the program will be executed.

It is possible to insert a Trojan horse by given it the same name as an existing program and put it in a directory that is searched earlier then the original programs directory. To defend against this type of attacks type in the full path name.

How hosts.allow and hosts.deny are used to determine access control to net- work services in Unix.

Lite oklart. Star de i boken?

Föreläsning 6 (kap.8)

The problem with the LM hash and how the NTLM hash is better. You do not have to know the exact details of how they work

NTLM hash is stored in the SAM file (local accounts)

◦ Problem 1: MD4 is a very fast hash function

◦ Problem 2: No salt is used so time-memory tradeoff attacks (rainbow tables) can be used

Possibly, also the **LM hash** is stored in the SAM file

◦ Problem 3: DES is a fast block cipher

◦ Problem 4: No salt here either...

◦ Problem 5: Passwords up to 14 characters are never better than passwords of 7 characters

◦ Problem 6: There are no lowercase characters in the effective character set

What are access tokens and what do they contain?

An **access token** contains the security information for a login session and identifies the user, the user's groups, and the user's privileges. The access token is used by Windows when the process or thread tries to interact with objects whose security descriptors enforce access control (*securable objects*).

The access token is generated by the logon service when a user logs on to the system and the credentials provided by the user are authenticated against the authentication database, by specifying the rights the user has in the security descriptor enclosed by the token. The token is attached to every process created by the user session (processes whose owner is the user).Whenever such a process accesses any resource which has access control enabled, Windows looks up in the security descriptor in the access token whether the user owning the process is eligible to access the data, and if so, what operations (read, write/modify, etc.) the user is allowed to do.

If the accessing operation is allowed in the context of the user, Windows allows the process to continue with the operation, else it is denied access.

Primary tokens can only be associated to processes, and they represent a process's security subject.

Impersonation is a security concept unique to Windows NT, that allows a server application to temporarily "be" the client in terms of access to secure objects. Impersonation has three possible levels: *identification*, letting the server inspect the client's identity, *impersonation*, letting the server act on behalf of the client, and *delegation*, same as impersonation but extended to remote systems to which the server connects (through the preservation of credentials).

The concept of privileges.

- The right to perform system related operations
  - Shuttingdown
  - Change system time
  - Backup files
  - Generate audit

Applies only to local computer. A user can have different privileges on different machines in a domain.

Access token is checked when user tries to perform privileged operation.

- Differs from access rights
  - Access to resources and tasks, not objects
  - Stored with subject
  - Admin assigns privileges

Stored in access token produced at logon.

What are security descriptors and what do they contain?

Contains security information associated with an object

Owner SID
Primary group SID
DACL
SACL

The **Owner SID** indicates the owner of the object. Objects gets an owner when they are created. The **primary group** is included for POSIX compliance. The **Discretionary Access Control List (DACL)** determines who is granted or denied access to the object. The **System Access Control List (SACL)** defines the audit policy for the object (specifying types of events that should generate audit records).

The concept of DACL, how access control lists are searched and what con- sequences does the search have? The difference between Windows and Unix is important here.

- Identifies who is allowed or denied access to an object
- If an object has no DACL, everyone has full control
- An empty DACL results in everyone is denied access
- A SID can be allowed or denied access.
- All ”deny” entries are stored in the beginning of the DACL

Searching the DACL

- Go through list of ACEs until all access requests are allowed or any access request is denied
- Otherwise deny access
- Consequences
  - Deny has higher precedence than allow
  - If user SID has read only access and user is member of group which SID has read + write, then user has read + write access (Different from Unix/ Linux)

The purpose of the SACL.

A system access control list (SACL) enables administrators to log attempts to access a secured object. Each ACE specifies the types of access attempts by a specified trustee that cause the system to generate a record in the security event log. An ACE in a SACL can generate audit records when an access attempt fails, when it succeeds, or both.

How access control on the network differs from access control on the computer.

??

Access control for a restricted token.

Restricted tokens remove privileges from a given access token. By adding a restricted SID to the token, a process with a restricted token gets access only of *both* the SID and the restricted SID are granted access.

To restrict the access rights of a program we can create a restricted SID representing this program. This SID has to be entered into the DACLs of all objects the program should have access to. Restricted SIDs can be created both for programs and object types.

Föreläsning 13 (kap.9)

How a general tracker can be used to derive sensitive information in statistical databases.

Kolla föreläsningen

In statistical databases only statistical queries are permitted (aggregation operators) ion an attribute (column) of a table. Some of these queries are: COUNT, SUM, MAX, MIN and AVG. There are several problems with this type of database:

- The individual data items are sensistive.
- Statistical queries read individual items.
- We want to provide users statistical information without compromising confidentiality of individual items.

General Tracker (how to derive sensitive data)

A tracker attack is a effective type of indirect attack (often hard to defend against). In this type of attack you only use legitimate queries to retrieve sensual information. How to do it:

- Let R uniquely idenrify row *r.*
- Let T be a query predicate such that the set returned form T and the set returned for NOT(T) are large enough.
- If we use both R OT T and R OR NOT(T) the *r* is the only row used in bith queries.
- T is called a general tracker.

Föreläsning 12 (kap.10)

You should understand how buffer overflow attacks on the stack work and how they can be prevented and detected.

How it works

The idea is to copy more data into a buffer than what has been allocated for that buffer, thereby overwriting the return address on the stack. This is possible when the programmer has used unsafe functions and if the buffer size of the destination is smaller than that of the source buffer. Overwriting the return address with an address that points to the attacker's code can force the program to run that code.

How to prevent it

A canary word could be inserted before the local variables, before returning from process the canary is checked so that it has not been changed. But if the value is known by the attacker it can just be overwritten with the same value.

The canary solution can detect the attack but not prevent it. To prevent it don't use unsafe functions (strcpy())

You should understand the idea behind SQL injection attacks and that they can be prevented by validating the input from users.

A **SQL injection** is often used to attack the security of a website by inputting SQL statements in a web form to get a badly designed website to perform operations on the database (often to dump the database content to the attacker) other than the usual operations as intended by the designer. SQL injection is a code injection technique that exploits a security vulnerability in a website's software. The vulnerability happens when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed.

A straightforward way to prevent injections is to *escape* characters that have a special meaning in SQL. One way to do this is by using mysql_real_escape_string().

Föreläsning 8 (Kap. 11-13)

The three security properties in Bell-LaPadula. The main ideas for each property.

ss-property: If access operation includes read then subject security level must dominate classification of the object.

***-property**: If access operation includes append then subject security level must be dominated by the classification of the object.

ds-property: Any access operation must be allowed in the access control matrix. The matrix can be used to pass access rights on to other users.

Tranquility in Bell-LaPadula.

Security levels and access rights never change.

Downgrade all subjects to the lowest security level. Downgrade all objects to the lowest security level. Enter all access rights in all entries of access control matrix. Everyone can now do everything. Two different opinions: Not secure, or if such a transition is required it should be OK (otherwise not implemented).

Conceptual difference between Bell-LaPadula and Biba models.

Both security models assign security levels to subjects and objects and access is granted based on these levels. The Bell-LaPadula security model focuses on confidentiality. For read access it is required that the security level of subject dominates the security level of the object. For append access it is required that the security level of the object dominates the security level of the subject (no read-up, no write-down). The Biba model focuses on integrity and for read and append (denoted modify in Biba) the opposite is true (no write-up, no read-down).

How are new integrity levels computed after reading/writing to an object in the case of dynamic integrity levels in Biba?

Static integrity levels

Similar to tranquility in BLP. Here the we can state policies where integrity levels never change. Has two different policies:

- Simple integrity policy: Corresponds to the ss-property in BLP. Has no write up.
- Integrity *-property: No read down, corresponds to *-property in BLP.

Dynamic integrity levels

Here the integrity properties automatically adjust the integrity level of an entity if it comes into contat with dirty information.

- Subject low level property: Subject s can read an object o at any integrity level. The new integrity level of the subject is the greatest lower bound of fS(s) and fO(o).
- Object low watermark property: ject s can modify an object o at any integrity level. The new integrity level of the object is the greatest lower bound of fS(s) and fO(o).

You should know the fact that Clark-Wilson focuses on integrity.

Focuses on security (integrity) in commercail systems. Says that the data is consistent if it satisfies given properties. The integrity are divided into two parts:

- **Internal consistency:** refers to the relation if the internal state of a system and can be enforced by the computing system.
- **External consistency:** refers to the relation of the internal state of a system to the real world and has to be enforced by means outside the computing system.

The general mechanisms for enforcing integrity are as follows:

- **Well-formed transactions:** the user can only change system through programs.
- **Separation of duties:** User can onyl use a certain set of programs.

Strengths and limitations of security evaluation.

Difference between Orange Book, ITSEC and Common Criteria in terms of functionality and assurance for products.

In orange book, functionality and assurance were not separated. Being evaluated in one security class meant claiming both a certain functionality and a certain assurance. Prod- ucts with low functionality could not be evaluated with high assurance and vice versa. In common criteria there is some separation between functionality and assurance. A product can be evaluated against one or more protection profiles which state the functionality of the product. The assurance level is then separated from this and is given by how much effort was put into the evaluation.

Classification of products in Orange Book, ITSEC and Common Criteria, but not the details of the different classes or evaluation levels.

Hm? Delar upp I olika classer beroende på hur sacra dom är?

Föreläsning 2 (Kap 14)

Empirically, provably and unconditionally secure.

Empirically secure – There are no known attacks on the algorithm. This is the strength of most (unbroken) stream ciphers and block ciphers, e.g., AES.

- Most common for practically used symmetric primitives
- Typically very efficient

Provably secure – The problem of breaking the algorithm can be reduced to the problem of solving a hard problem, i.e., factoring or the discrete logarithm problem. An example is RSA.

- Most common for asymmetric primitives

Unconditionally secure – It is not possible to break at all, regardless of computation time. One example is the one time pad (or Vernam cipher).

- Not common but possible

Kerckhoffs' principle.

- Only the key should be unknown to an adversary
  - o Security should not be based on the fact that the algorithm is secret

Relation between stream ciphers and OTP

Stream ciphers vs. block ciphers.

A **block cipher** is a symmetric key cipher operating on fixed-length groups of bits, called *blocks*, with an unvarying transformation. A block cipher encryption algorithm might take (for example) a 128-bit block of plaintext as input, and output a corresponding 128-bit block of ciphertext. The exact transformation is controlled using a second input — the secret key. Decryption is similar: the decryption algorithm takes, in this example, a 128-bit block of ciphertext together with the secret key, and yields the original 128-bit block of plain text.

A **stream cipher** is a symmetric key cipher where plaintext digits are combined with a pseudorandom cipher digit stream (keystream). In a stream cipher each plaintext digit is encrypted one at a time with the corresponding digit of the keystream, to give a digit of the cyphertext stream.

Stream ciphers represent a different approach to symmetric encryption **from block ciphers**. **Block ciphers** operate on large blocks of digits with a fixed, unvarying transformation. This distinction is not always clear-cut: in some modes of operation, a block cipher primitive is used in such a way that it acts effectively as a stream cipher. Stream ciphers typically execute at a higher speed than block ciphers and have lower hardware complexity.

Block cipher modes ECB, CBC and OFB.

ECB - The message is divided into blocks and each block is encrypted separately. The disadvantage of this method is that identical plaintext blocks are encrypted into identical cipher text blocks; thus, it does not hide data patterns well. In some senses, it doesn't provide serious message confidentiality, and it is not recommended for use in cryptographic protocols at all

In **ECB** mode all plaintext blocks are encrypted independently, $c_i = E_K(m_i)$ and $m_i = D_K(c_i)$. This preserves redundancy in the plaintext.

In **CBC** mode redundancy is removed by adding (xor) each new plaintext block with the previous ciphertext block, $c_i = E_K(p_i \oplus c_{i-1})$ and $p_i = D_K(c_i) \oplus c_{i-1}$.

Counter mode turns a block cipher into a stream cipher. The keystream is generated by encrypting a counter concatenated with the IV, $z_i = E_K(IV /\!/ i)$, $c_i = p_i \oplus z_i$ and $p_i = c_i \oplus z_i$.

Symmetric vs. asymmetric cryptography and which types of algorithms be- long to which group

Symmetric algorithms are much faster than asymmetric algorithms. About a factor 1000.

Symmetric algorithms can use shorter key with same security. 1024 bit RSA modulus corresponds to about 80 bit symmetric key.

Elliptic curves are often used to make public key cryptography more efficient. Both shorter keys and faster algorithms are possible.

You should be able to compute a toy RSA example, which means that you should know the RSA algorithm.

Defining properties of hash functions and the additional properties preimage resistance, second preimage resistance and collision resistance.

Ease of computation: Easy to compute h(x)

Compression: x of arbitrary bit length maps to fixed length n output.

Preimage resistance: Given y, it is difficult to find x such that h(x) = y

Second preimage resistance: Given x and h(x), it is difficult to find x′ such that x′ =/ = x and h(x′) = h(x)

Collision resistance: It is difficult to find x and x′ such that x′ =/= x and h(x′) = h(x)

Birthday paradox. You do not have to know how to derive it but you should know its meaning and consequences.

If a test has n possible outcomes, the number of tests needed before there is a collision in the outcome is in the order of √n. This gives an upper bound on the problem of finding collisions in a hash function.

Properties of MAC functions.

Defining properties

◦ Easy of computation – Given k and x, hk(x) is easy to compute.

◦ Compression – hk(x) maps x of arbitrary bit length to fixed length n output.

◦ Computation resistance – given zero or more pairs (xi,hk(xi)), it is infeasible to compute a pair (x, hk(x)) with a new message x.

Does NOT provide encryption. That has to be added separately!

Digital signatures and how they differ from MAC functions.

A MAC is calculated using a shared key. Thus anyone in possession of the shared key can both verify and calculate the MAC. Digital signatures are based on asymmetric cryptography. A private key is used to sign a message and a public key is used to verify the signature. Thus anyone can verify a signature but only the person in possession of the private key can sign a message.

A digital signature provides nonrepudiation, meaning that a person that has signed a message cannot later deny having signed it. This is not the case for a MAC.

Since digital signatures are based on asymmetric cryptography and MACs are based on symmetric cryptography, a MAC calculation is much faster than a digital signature.

MAC algorithms are e.g., HMAC or CBC-MAC. Digital signature algorithms are e.g., DSA or RSA.

You should know that El Gamal is based on the discrete logarithm problem.

Föreläsning 9 (kap. 15)

The terms key transport, key agreement, implicit key authentication, key confirmation and explicit key authentication.

Key Transport - one party creates/obtains secret key and securely transfers it to the other party (also called key distribution)

Key Agreement - Both parties contribute to the generation of the secret key

(Implicit) Key Authentication - One party knows that no one besides a specifically identified second party may gain access to a secret key

Key Confirmation - One party is assured that the second party has possession of a secret key (but identity of the other party may not be known)

Explicit Key Authentication - Both implicit key authentication and key confirmation

Certificates and certificate chains. You do not have to know every field in a certificate, only the most important ones.

Certificates are, amongst a lot things, used to verify that the verificatin keys used for authetication indeed correspond to the right party. The certificate is a digitally signed document that binds a subject to some other information. The subjects can be people, keys, names etc. The binding between a subject and key is established by the party that issues (signs) the certificate. CA is a name for the issuer.

The certificate chain is a list of certificates used to authenticate an entity. The chain begins with the certificate of that entity and each certificate in the chain is signed by the entity identified by the next certificate in the chain. The chain terminates with a root CA certificate. The root CA certificate is always signed by the CA itself.

You should know the Diffie-Hellman protocol (how it works), the main prob- lem with the protocol and how STS solves this problem. For the other proto- cols (AKEP2, password based, Needham-Schroeder and Kerberos) you should only remember the most important things about them, e.g., the problem with Needham-Schroeder and how Kerberos solves this problem. You should be able to understand them when you see them, but not remember the details by heart.

Diffie-Hellman protocol
Diffie-Hellamn is a key agreement protocol. A and B do not share a secret in advance. Instead they agree on a group G of prime order q, and on a generator g of this group. The group G could be defined as a subgroup of order q in the group of integers modulo p, where p is a large and appropriately chosen prime number.

Principal A picks a random number x and sends X=g^x to B. B picks a random number y, send Y=g^y to A, and computes X^y. On receipt of Y, A uses its own secret to compute Y^x. Because of X^y=g^xy=g^yx=Y^x. Both parties now share the secret g^xy. The security of this protocol is based on the discrete logarithm problem.

The problem with the Diffie-Hellman protocol is that there is no key authentication (no party knows with whom they share the secret). Due to this it can be exploited by a man-in-the-middle attack.

To add authentication to the Diffie-Hellman protocol the Station-to-station (STS) protocol is used.

Password-Based protocols

In password-based protocols long-term keys needs to be stored on clients. A password can represent a key. It is convenient for human interaction since it is easy to remember a password.

It is possible to use a Password P to encrypt a randomly generated session key Ks, and use the session key to encrypt further data. The problem with this protocol is that it is vulnerable to offline-dictionary attacks and brute force attacks on the password using data redundancy. Password are often badly chosen.

The Encrypted Key Exchange (EKE) protocol avoids this problem. It uses a symmetric encryption algorithm to encrypt data with the password P as the key, and also a public-key encryption system. (Use a temporary public key K, encrypted with password to encrypt session key).

Needham-Schroeder protocol

A key transport protocol. Two parties, A and B, obtain their session keys from a server S (seen as a trusted third part). Both parties share a secret key with the server in advance. A symmetric cypher is used for encryption. To avoid replay attacks nonces (random challenge, an arbitrary number only used to sign a cyrptographic communication) are included in the messages.

How the protocol works: A requests a session key K(ab) from the server S, that is intended for communication with B. In the first three protocol steps, A obtains the session key from S and forwards it to B. By checking a nonce returned in the servers message, A can verify that the session key has been used in response to its recent request and is not a replay from a former protocol run. In the last two steps, B verifies that A is currently using the same session key. In step 4 and 5, A performs a unilateral entity authentication of B.

Problems: B does not know if the session key created by S for use between A and B is fresh or not. If it is possible to break one session key it will be possible to reuse it and perform a replay attack. The adversary can then enter the protocol at message 3.

The solution to this issue is to include lifetime for tickets, which Kerberos does.

Kerberos: Almost like Needham-Schroeder but with timestamps and limited lifetimes for session keys. When B receives K(ab) from A it checks the lifetime of ticket. B then autheticates A by checking that identity is same in both ticket and authenticator.

Shortly about kerberos:

- A kerberos Authetication Sever is used together with one or several Ticket Granting Servers.
- A principal is a user or a server.

- KAS authenticates principals at login and issues Ticket Granting Tickets (TGT), which enables principals to obtain other tickets for TGSs
- TGSs issues tickets that give princpals access to network services demanding authentiation.
- Kerberos 4 uses DES as symmetric cipher, Kerberos 5 can use other algorithms.
- Users authenticate using passwords.

Föreläsning 10 (kap. 16)

Replay attacks.

A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed.

Suppose Alice wants to prove her identity to Bob. Bob requests her password as proof of identity, which Alice dutifully provides, meanwhile, Mallory is eavesdropping on the conversation and keeps the password (or the hash). After the interchange is over, Mallory (posing as Alice) connects to Bob; when asked for a proof of identity, Mallory sends Alice's password (or hash) read from the last session, which Bob accepts.

Traffic analysis.

Traffic analysis is the process of intercepting and examining messages in order to deduce a information from communication patterns. It can even be performed when the messages are encrypted and cannot be decrypted. An attacker might try to identify messages coming from the same source or find out whom is talking to whom, and how often.

The SSL handshake when RSA is used.

OBS: kolla på föreläsningsslidsen eftersom man ska kunne utgöra vilka meddelanden som är viktiga.

The purpose of the SSL handshake is to authenticate server to client, establish which algorithms to use, negotiate keys for encryption and MAC, and authenticate client to server (optional). There are ten different message types bu they depend on which key exchange method that is used and if the server autheticates the client. The handshake:

- The client start by sending a Hello message to the server. There it has attached suggested cipher suites.
- The server now responds with a server hello message, The server now selects a cipher suite that suits them both (in this case RSA). If the client needs to autheticate the server it sends its certificate chain.

- If the server needs to authenticate the client it now sends it certificate chain. The client now generate a "premaster secret" (48-byte) and uses RSA to encrypt it with the public key of the server (this is the client key exchange). This is then sent to the server.
- The server the decrypts the "premaster secret". The server also verifies the has appended to the client's message and replies. The client can now verify the hash in the servers message.

Purpose of random numbers in SSL.

It is used to provide a seed to the "premaster secret" (like salt in password hashing). It allows both the server and client to contribute to the key generation (key agreement). It avoids replay attacks. A sniffed session cannot be replayed by a fake client or fake server. Each session has a new random number.

Basics of IPsec.

IPsec provides security at the network layer. All IP datagrams are covered. It is mandatory for IPv6 but optional for IPv4. There are two different options:

- *Transport mode:* Protection for upper-layer protocols. The Protection covers IP datagram payload (TCP packets, UDP, ICMP). It offers host-to-host security (end-to-end). Both nodes need to be IPsec-aware.
- *Tunnel mode:* Offers protection for the whole IP datagram. Entire datagram plus security fields are treated as new payload of "outer" IP datagram. The original datagram is encpasulated within an outer IP datagram. IPsec processing is performed at security gateways on behalf of endpoint hosts. The hosts need not to be IPsec-aware.

There are also two major security mechanisms:

- *Authentification Header:* It provides data origin authentification and data integrity using a MAC, however, it does not protect confidentiality. AH authenticates the whole payload and most of header. It prevents IP address spoofing since source IP is authenticated. It also prevents replay attacks. The AH: header is added to the original IP packet. Look at lecture 11.
- *Encapsulating Security Payloads:* It can provide either one or both of: confidentiality and authentication. It uses symmetric encryption and MACs based on secret keys shared between endpoints. ESP specifies a header and trailing fields to be added to IP datagrams. IT can be combined with the two previous modes. (Authentification in ESP does not cover the original IP header, if this is needed AH can be added after ESP. This is called Transport Adjacency).

The difference between the 4 ways of using IPsec, {AH, ESP}× {Transport, Tunnel}.

AH guarantees connectionless integrity and data origin authentication of IP packets. Further, it can optionally protect against replay attacks by using the sliding window technique and discarding old packets

ESP is a member of the IPsec protocol suite. In IPsec it provides origin authenticity, integrity, and confidentiality protection of packets. ESP also supports encryption-only and authentication-only configurations, but using encryption without authentication is strongly discouraged because it is insecure.[13][14][15] Unlike Authentication Header (AH), ESP in transport mode does not provide integrity and authentication for the entire IP packet.

In **transport mode**, only the payload of the IP packet is usually encrypted and/or authenticated. The routing is intact, since the IP header is neither modified nor encrypted; however, when the authentication header is used, the IP addresses cannot be translated, as this will invalidate the hash value. The transport and application layers are always secured by hash, so they cannot be modified in any way (for example by translating the port numbers). Transport mode is used for host-to-host communications.

In **tunnel mode**, the entire IP packet is encrypted and/or authenticated. It is then encapsulated into a new IP packet with a new IP header. Tunnel mode is used to create virtual private networks for network-to-network communications (e.g. between routers to link sites), host-to-network communications (e.g. remote user access), and host-to-host communications (e.g. private chat).

Föreläsning 11 (Kap. 17)

The difference between packet filters, stateful packet filters and application level proxies.

Packet filters do not examine data at application level. Instead the packet filter only looks at information in the IP and TCP header, i.e., IP addresses and port numbers. Rules based on these determine if a packet is allowed to pass the firewall or if it should be blocked. A stateful packet filter is similar, but keeps in memory all ongoing connections. If a connection is initiated from inside the firewall, traffic returning to the same port will be allowed.

This memory is needed since the firewall cannot in general predict which port is used by client applications and without the memory all incoming traffic on ports that can be used by clients must be allowed. Relays application-level traffic } Sets up its own connection to remote host

Application level proxy relays application-level traffic. It sets up its own connection to remote host

- Implements the protocol
- Can filter data at application level, e.g remove attachments
- Can also be used to anonymize, fake source country etc.

Can audit and log at application level. But it’s slower then packet filters and have a higher cost.

Idea behind anomaly detection and misuse detection for an IDS, and the differences.

Intrusion Detection System (IDS) are designed to detect attacks unlike Cryptography and protocols that focuses on preventing attacks. It is important that the IDS is secure in itself. It can be based on *Misuse detection* or *Anomaly detection.*

Anomaly Detection – refers to detecting patterns in a given data set that do not conform to an established normal behavior. The patterns thus detected are called anomalies and often translate to critical and actionable information in several application domains. Anomalies are also referred to as outliers, change, deviation, surprise, aberrant, peculiarity, intrusion, etc. But *all attacks are not necessarily anomalies.*

Common metrics

- o *Counter* - Can e.g., be number of logins/hour, number of times a command is executed/login, number of password failures
- o *Gauge* - Can e.g., be number of connections to application or server
- o *Interval timer* - Length between two related events e.g., logins to an account
- o *Resource utilization* - Amount of resources used during some period e.g., pages printed, total time of program execution

Misuse Detection – Looks for attack signatures. Examines the network traffic or log files e.g failed logins. It uses a database of signatures which has to be kept up to date. It works well against attacks with a fixed behavior.

Heuristic rules can also be used

- Users should not read files in other users’ personal directories
- Users must not write to other users’ files
- User do not open devices directly, but instead through other programs
- Users should not be logged in more than once to the same system
- Users do not make copies of system programs

Purpose of Honeypot.

A honeypot is a computer that does not contain any sensitive information. No one has any (legitimate) reason to access the honeypot, so any access to this computer is by definition an attack attempt. It can be used to divert attackers from the computers containing the real information.

It can also be used to gather information about certain attacks and learn how they are performed in order to protect other parts of the system from the same attacks. They can also be placed inside a company in order to find insider threats, like corporate espionage.

Föreläsning 15 (Kap. 19)

How authentication and key agreement works in GSM.

- A key Ki is shared between the SIM and the AuC in the home network.
- The SIM sends his TMSI to the VLR which maps this to the IMSI which in turn is sent to the AuC/HLR to identify the SIM
- The AuC/HLR uses the shared key together with a random number RAND to compute a session key Kc and an expected response RES used for authentication of the SIM.
- RAND, RES and Kc are sent to the VLR
- The RAND is sent to the SIM which, together with Ki computes Kc and the response RES'
- RES' is sent to VLR which checks that it agrees with the expected response

The important improvements made to the authentication and key agreement in UMTS compared to GSM.

In UMTS the network is also authenticated, not only the SIM. An integrity key is used to calculate a MAC on signalling messages. Additionally, stronger encryption is used

The fact that the encryption algorithm in GSM is a stream cipher and in UMTS a block cipher in a stream cipher mode of operation.

CRC-32 problem, IV size problem and authentication problem in WEP.

Confidentiality: The usage of the IV in the stream cipher is flawed. Also the IV is too short.

Integrity: The integrity algorithm used is linear. Since a stream cipher adds the keystream linearly to the message it is possible to modify the message and correctly compute a new integrity check value even though the message is encrypted.

Authentication: The challenge response scheme uses a stream cipher to encrypt the re- sponse. By eavesdropping one authentication, it is easy to obtain the keystream and then authenticate as that user at any time.

The fact that WPA2 uses AES and is completely different from WEP and WPA.