CMPSCI 677 Operating Systems **Spring 2013**

Lecture 24: April 29

Lecturer: Prashant Shenoy Scribe: Anil Kumar Venkatesh

24.1 Security

24.1.1 Focus of Control

Some of the users can be malicious; there may be cases of unauthorized accessing of system, so we need to protect our system from these. Three approaches for protection against security threats they are a) Protection against invalid operations b) Protection against unauthorized invocations c) Protection against unauthorized users

24.1.2 Authentication

This is a process where the user can prove the system its identity and obtain access to the system. Various authentication protocols are as follows.

AP 1.0:
Here Alice sends a message to bob identifying itself, but the problem with such a fundamental approach is any third party can also claim to be Alice.

AP2.0:
The problems faced in first step can be negated by authenticating source IP address is from Alice's machine, but another problem arises here where the third part can just do IP spoofing by using a simple python code.

AP3.0:
To overcome the problems faced in 2nd step we can use a secret password, but hacker can intercept this message "sniffing". Earlier protocols like telnet the username and password is in plaintext, hence easy to sniff and crack the system.

AP3.1: Use Encryption
To overcome the problems faced in AP 3.0 we can use Symmetric key encryption where Alice & bob both know a secure key. Both Alice & Bob communicate by encrypting the message with the shared common key. This proves the authentication of Alice or Bob, since only those two have the key. But this is going fail if the hacker captures the encrypted packet and does reply attack, where the same message is sent to bob at a later time. Hence, in this scenario the encryption fails.

Prof questions what is the solution to this problem? the answers to this is "using random number", "use time stamps" , "use time out mechanisms".

Authentication Using Nonce
AP 4.0:
To overcome the problems faced in AP3.1 we can make use of nonce, when Alice sends message to Bob (unencrypted) Bob sends a nonce to Alice, then Alice encrypts the nonce and proves to Bob that it's the true Alice.

A to B: msg = "I am A" /* note: unencrypted message! */
B to A: once-in-a-lifetime value, n
A to B: msg2 = encrypt(n) /* use symmetric keys */
B computes: if decrypt(msg2)==n
then A is verified ,else A is fradulent

Authentication using Public keys

AP 5.0:
Each entity (Machine) generates two key, public and private, private is supposed to be kept secret. The public key can be distributed so that others can contact.

A to B: msg = "I am A"
B to A: once-in-a-lifetime value, *n*
A to B: msg2 = DA(*n*)
B computes: if EA (DA(*n*))== *n*
then A is verified, else A is fradulent

Problems with AP 5.0
An intruder can do the man in the middle attack, i.e. Trudy can impersonate as Alice to Bob. There needs to be a secure key distribution. The protocol is only as secure as key distribution.

Digital signatures using Public Keys
It's a way of signing the message sent, they are supposed to be equivalent to physical signatures made to a document. It's a digital form of a signature. Sender cannot repudiate message never sent and same way the receiver can never fake a received message.
Eg: suppose if A wants B to "Sign" a message M. B sends DB(M) to A , A computes if EB (DB(M)) == M , Then B has signed M.

Message digests
They are essentially hash functions; it takes an arbitrary length of strings but outputs a fixed length hash and this hash function is signed using methods specified earlier. Since the hash is significantly smaller in size and hence not computationally intensive to make Digital Signatures on this, this seems to be more efficient and better approach than hashing the entire message to authenticate. When the receiver, receives the (message + signed hash), the received messaged can be hashed and verify the signed hash.

Properties of hash function
(a) Given a digest x, it is infeasible to find a message y such that H(y) = x,
(b) It is infeasible to find any two messages x and y such that H(x) = H(y)

Hash functions

MD5:
Takes arbitrary message and converts into 128 bits constant length key. But MD5 is not secure anymore.

SHA hash functions (secure hash algorithm):
SHA-1 : 160-bit function that resembles MD5, SHA-2: family of two hash functions (SHA-256 and SHA-512), Developed by NIST and NSA

Symmetric Key exchange: trusted server
Alice sends an initiative message to the KDC that it wants to contact Bob, the KDC generates a secret key that only KDC & Alice knows ($K_{A,KDC}$) and using this key encrypts the Key shared between A & B ($K_{A,B}$). The KDC also sends the same key in the same method to B, Hence now both Alice and Bob can exchange data securely using their respective keys.

24.2 Access Control:

This gives us the capacity to restrict access to the system.
Access control list - where for every resource there is list of which user can read, write, and exit. Capabilities- Every user is basically given a list of capability, when it does an operation the user presents the list of capabilities and gains access.

24.2.1 Security in Enterprises:

Multi-layered approach to security in modern enterprises, Security functionality spread across multiple entities. Firewall guards what network packet enters or leaves the network, Deep packet inspection– In this the process looks inside the packet, not only source and destination address like the Firewall. They scan the packet for virus, malware etc. VLAN- virtual network, where the machines are on virtual Ethernet each machine can be restricted access to other machines on network. Network radius servers, Securing Wi-Fi, VPNs, Securing services using SSL, certificates, Kerberos etc.

24.2.2 Security in Internet services

Website has to run over SSL + authentication (to ensure user actually log into the system before usage) by means of password, username + captchas are used to limit access to humans and deny access to computer programs trying to bring down the website. Challenge-response authentication more secured than password authentication. Two factor authentication Gmail uses password and mobile to authenticate the user. One time passwords are used by Microsoft, where once in a lifetime challenge message is sent to a mobile device, increases the security. E.g.: Hotmail

24.2.3 Firewall

A firewall examines every packet going in/out and compares it to a rule the admin has specified, and allows or denies the action. Packet filtering based on certain rules set by the network administrator. There are other firewalls which are not just packet filtering, but are deep packet inspection firewalls and filter the packets based on content of the packet.

24.2.4 Secure Email

Some of the requirements of a secure email service are as follows, secrecy- only sender and receiver can see the mail and contents, Sender authentication – it has to be authenticated, it has to be proven that the sender is valid and authentic. Message integrity – has to make sure that, message does not get altered in transit. All this can be done using public key encryption, as follows.

Authentication and Integrity (with no secrecy)

- Alice applies hash function H to M (H can be MD5)
- Creates a digital signature DA(H(M))
- Send M, DA(H(M)) to Bob

Putting it all together

- Compute H(M), DA(H(M))
- M'= { M, DA(H(M)) }
- Generate symmetric key K, compute K(M')
- Encrypt K as EB(K)
- Send K(M'), EB(K)

24.2.4 Secure Socket Layer

Provides data encryption and authentication between web server and client. SSL lies above the transport layer. Useful for Internet Commerce, secure mail access (IMAP)

There are several features of SSL they are as follows SSL server authentication, Encrypted SSL session, SSL client authentication.

Following are the events that take place before setting up of an SSL connection.

- Browser -> Server: B's SSL version and preferences
- S->B: S's SSL version, preferences, and certificate- Certificate: server's RSA public key encrypted by CA's private key
- B: uses its list of CAs and public keys to decrypt S's public key
- B->S: generate K, encrypt K with ES
- B->S: "future messages will be encrypted", and K(m)
- S->B: "future messages will be encrypted", and K(m)
- SSL session begins…