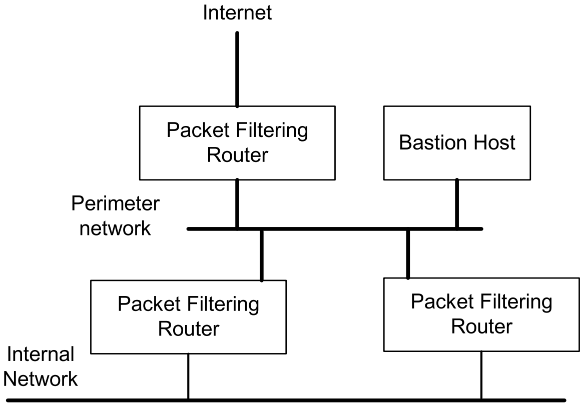
527 — Computer Networks and Distributed Systems — Tutorial 3: Security

Emil Lupu

Note that the solution notes below only briefly list (some of) the key points that should be included in an answer. They are by no means complete. In an exam, you are expected to spell out the solution more fully and include a detailed explanation of your reasoning.

1 Firewalls Architecture

A company uses the firewall set-up below.



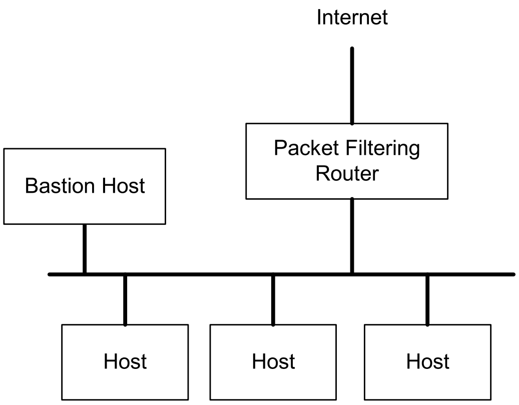
- Explain why a company would use the set-up above with two internal packet filtering routers (the ones connecting the internal network to the perimeter network).
- If the perimeter network is compromised and the internal packet filtering routers are poorly configured this set-up may lead to information leakage. Explain why this can occur and give the firewall rules necessary to avoid it.

Solution Notes:

- Performance, Reliability
- If the firewalls are poorly configured the internal routing software may decide that the shortest path between two internal hosts is through the perimeter network. To remedy to this situation it is necessary to have both internal firewalls configured to avoid routing internal traffic. Several options are possible:
  1. deny all traffic that does not go to the bastion host.
  2. deny all traffic having both source address and destination address in the internal network.

2 Firewall Configuration

A company called DoC uses the firewall set-up below. The bastion host has a *web server* (port 80), *an SMTP server* (port 25), *an FTP server* (port 21) as well as an *FTP proxy* (port 4421) and an *HTTP proxy* (port 4488). Company users can access outside information only through the proxies on the bastion host.



Give a possible set of rules for the packet filtering router in the format below, and explain what each rule does. Assume the direction of the traffic can be *in*, *out*, or *any* (both directions).

Rule No.	Protocol	Dir.	Source Addr.	Src. Port	Dest. Addr.	Dest. Port	TCP Flags	Action

Solution Notes:

Explanations, expecially of non trivial rules, are also required.

Rule	Direction	Source Address	Src Port	Dest. Address	Dest. Port	TCP Flags	Action
S1	In	DocNet	*	*	*		deny
S2	Out	#Bastion	*	*	*		deny
W1	In	*	*	Bastion	80		Allow
W2	Out	Bastion	80	*	>1024	ACK	Allow
WP1	Out	Bastion	>1024	*	80		Allow
WP2	In	*	80	Bastion	>1024	ACK	allow
F1	In	*	>1024	Bastion	21		Allow
F2	Out	Bastion	21	*	>1024	ACK	Allow
F3	Out	Bastion	20	*	>1024		Allow
F4	In	*	> 1024	Bastion	20	ACK	Allow
FP1	Out	Bastion	>1024	*	21		Allow
FP2	In	*	21	Bastion	>1024	ACK	Allow
FP3	In	*	20	Bastion	> 1024		Allow
FP4	Out	Bastion	>1024	*	20	ACK	allow
M1	In	*	*	Bastion	25		Allow
M2	Out	Bastion	25	*	*	ACK	allow
X	Any	*	*	*	*		Deny

3 Notary Service

A centralised *notary* service is used to validate that a message came from a particular sender, the time at which it was sent and that it has not been modified. Messages are sent to the notary which then forwards them on to the final destination. The notary holds a secret encryption keys for each client, but clients can share secret keys not known to the notary.

Use the following notation:

$X, K_{ab}\{Y\}$ A message contains a field X sent as plain text and a field Y encrypted with a secret key K known to A and B

- Show how A can communicate with B via the notary, but prevent the notary from reading the message. Indicate the contents of messages and explain how your system works.
- What are the disadvantages of this approach?

Solution Notes:

$A \rightarrow N$: A, B , $K_{ab}\{m\}$, $K_{an}\{A, H(K_{ab}\{m\}), T_a\}$

Message m is protected by K_{ab} known only to Alice and Bob so cannot be read by the notary. T_a is As timestamp. The Notary validates encrypted message by checking the hash digest and inserts its timestamp T_n in the message.

$N \rightarrow B$: N, $K_{bn}\{A, T_a, T_n\}$, $K_{ab}\{m\}$, $K_{an}\{A, T_a, T_n, H(K_{ab}\{m\})\}$

Bob knows the timestamp of generation and signing so can check for freshness. The notary's signature $K_{an}A, T_a, T_n, H(K_{ab}\{m\})$ is stored though Bob cannot decrypt it. Only m needs to be stored as $K_{ab}m$ can be recreated.

In case of dispute Bob sends the following to the Notary:

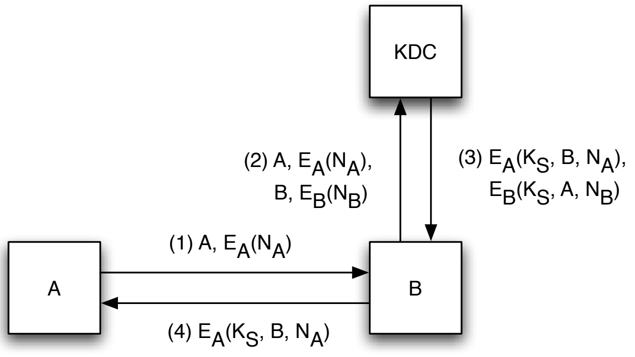
$B \rightarrow N$: B, $K_{bn}\{A, T_a, K_{ab}\{m\}\}$, $K_{an}\{A, T_a, T_n, H(K_{ab}\{m\})\}$

The notary can then verify the time, the sender and that the digest of the encrypted message is valid so m has not been changed.

All clients have to trust the notary service. If the notary service is compromised, all keys it holds have to be changed There is a single failure point and the notary service can be a bottleneck.

4 Simple Protocols

Consider the following protocol for key distribution and authentication using a KDC.



1. Describe the protocol and what it achieves.
2. Compare this protocol with the one seen during the lectures. In which situations is one or the other more suitable?

Solution Notes:

1. The protocol achieves key distribution and authentication.

2. The main difference with Needham-Schroeder is that the client does not need to talk to the KDC directly. Only the server does. So whilst Needham Schroeder is more suitable for an environment like Kerberos where both the client and the server are on the internal network, this protocol is more suitable for an environment where the client may be outside the network and only the server can talk to the KDC. Whilst many protocols are presented abstractly between Alice and Bob, practical constraints also need to be taken into account.

5 Basic Protocol Reasoning and Vulnerability

The following protocol is used by two entities A and B to exchange a message M and establish a session key K_S using a Key Distribution Centre (KDC).

$$\begin{array}{ll} A \rightarrow KDC\text{:} & A,\, B,\, N_A,\, K_A(K_S) \\ KDC \rightarrow A\text{:} & K_A(N_A),\, K_B(K_S) \\ A \rightarrow B\text{:} & A,\, K_S(M),\, K_B(K_S) \end{array}$$

Where:

- K_A and K_B denote the keys of A and B respectively, which the KDC knows,
- K_S denotes the shared session key between A and B and
- N_A denotes a random number or nonce generated by A

a) Demonstrate how using a replay attack, a man in the middle known to the KDC, can decrypt the message M.

b) Propose a modification to the protocol to counter this attack.

c) This protocol is used by the head of Arkadias secret services to arrange meeting points with her spies. Shortly after Arkadias secret services have been infiltrated by a double agent, all its spies are captured. Explain how, without relying on the previous attack, this is possible and propose a modification to the protocol to counter this possibility.

Solution Notes:

A man in the middle Z could capture the messages $K_A(K_S)$ and $K_S(M)$. Since Z is known to the KDC it can then do (pretending to be A):

?$A(Z) \rightarrow KDC$: A, Z, $K_A(K_S)$ and intercept the reply
$KDC \rightarrow A$: $K_Z(K_S)$ Z can now decrypt the session key and the message recorded earlier.

To counter this attack it is sufficient to encrypt the destination of the message:
$A \rightarrow KDC$: A ,N_A, $K_A(B, K_S)$ Even if this message is re-played it is not possible to change its destination an thus make it encrypted with a different key.

Note that the protocol does not authenticate A to B in any way. It only ensures that A is a valid user on the network known by the KDC. Thus anybody can send a meeting point to the spies.

To secure the protocol, B would need authenticated information on who the sender is. The protocol could then look as follows:

$$\begin{array}{ll} A \rightarrow KDC\text{:} & A,\, B,\, N_A,\, K_A(K_S) \\ KDC \rightarrow A\text{:} & K_A(N_A),\, K_B(A, K_S) \\ A \rightarrow B\text{:} & A,\, K_S(M),\, K_B(A, K_S) \end{array}$$

6 Public Key Infrastructures

a) On what is a Certification Authority an authority? Consider in your answer the case of SSL server certificates that contain the name of the subject (usually the name of the company) and the DNS name of the server.

b) It has been advocated that public keys are better identifiers than names and that this would simplify authorisation infrastructures. List three advantages and three disadvantages of using public keys as identifiers, carefully justifying your answers. (**Hint:** consider for example their impact on access control and anonymity.)

Solution Notes:

a) A certification authority certifies the binding between certain attributes. Usually in public key infrastructures this is the binding between a name and a public key. For attribute certificates this may be the binding between the public key and the attribute or the name and the attribute. The Certification Authority is only an authority on *issuing that certificate*. There are no guarantees that the CA is an authority on the *content* of the certificate. CAs typically list their practices and the steps taken (usually only for identity) in order to verify some of the content but compliance with those steps is usually not verified at the verifier. For X509 id certificates it is customary to issue certificates that follow a naming convention in which the keyholders name is prefixed (i.e., under the authority of) the certification authority e.g., employees of Imperial having a certificate issued by Imperial. For SSL certificates the CA is not an authority on either the DNS record or the company registration.

b) Advantages:

- Public keys are relatively unique and they are public. Considering them of sufficient length, the probability of collision would be extremely low, thereby making them suited for use as identifiers, more so than the use of common names.
- Using public keys as identifiers would obviate the need for identity certificates that bind the public key to a name (be it a distinguished name as in X.509 or an email address as in PGP). This would considerably simplify the verification in authorisations as the credentials chain linked to the verification of the identity certificate would not longer need to be given.
- Judicious use of public keys that uses different public keys for different purposes or categories of activities could provide anonymity to some degree as the user can “forget” the associated private key effectively forgetting his/her past identity.

Disadvantages:

- Public keys are difficult to remember and to recognise. It would thus be difficult for humans to make trust decisions based on recognising the name of a particular entity or the context of usage as it often appears in modern browsers.
- Because they are unique (have a low probability of collision), the use of public keys as identifiers would facilitate correlation across multiple records thus identifying patterns of behaviour across multiple sources e.g., shopping behaviour preferences. The level of certainty would be much higher than with current schemes.
- Data input by human operators (e.g., for administrative procedures) would be much more difficult and error prone.
- Because authorisations would be bound directly to public keys revocation, in case of compromise of a private key may be more difficult. The lack of certification authorities makes it more difficult to use Certificate Revocation Lists for revocation.

7 PKI Topologies

Compare the following Certification Authority (CA) topologies listing the advantages and disadvantages of each:

- A single global CA.
- A number of certificates are pre-loaded on client machines (e.g., browsers). Designated Authorities can issue certificates to authorise other CAs (called then delegated CAs) to issue certificates.
- A limited number of CAs are established independently, each for a separate administrative domain (e.g., organisation). The CAs cross certify each other.
- There are no CAs, individuals just sign each other’s keys.

Solution Notes:

Single Global CA

a1 There is a single root of total trust in the system. Everybody can preload that key.

a2 Searching for a trust chain is very efficient, since it always starts with the root.

d1 It would require a single organisation to be trusted by everyone.

d2 Should the root key need to change everything needs to change.

d3 Difficulty of registering the users to a single point, single point of failure.

Preloaded Certificates with delegation

a1 No single point of failure. and easier to register through a delegated CA.

a2 Increased availability for checking.

a3 Suitable for web environments and application deployments.

d1 Compromise of a delegate CA compromises the entire sub-hierarchy.

d2 Need for uniform rules across the certification authorities and its delegates.

Federated Multiple CAs. This model usually assumes a hierarchical name space and the subordination rules

a1 Faster lookups as only need to go up until we go across.

a2 Security can be deployed within an organisation without the need to obtain certificates from any outside organisation.

a3 Compromise of an external CA would not affect internal organisation.

d1 Requires explicit trust relationships between organisations to be set up out of band.

d2 Not suitable for deployment to individual users.

No structure: This is essentially the same as the PGP model.

a1 No need for CAs. The user defines specifically who/what it trusts.

a2 No need for a hierarchical name space.

d1 Does not scale beyond small communities.