

# Lecture 7

# Network management

- Keep the network healthy
  - Fault detection for networks, gateways, and critical servers
  - Schemes for notifying an administrator of problems
  - General network monitoring
  - Documentation and visualisation of the network
  - Administration of network devices from a central site

# FCAPS

- **F**ault Management
- **C**onfiguration Management
- **A**ccounting Management
- **P**erformance Management
- **S**ecurity Management

# Some principles first

- Make one change at a time.
- Document the situation as it was before you got involved, and document every change you make
- Start at one end of a network and work through the system's critical component
- Communicate regularly
- Work as a team
- Use the layers of the network to negotiate the problem. Start at the “top” or “bottom”.

# Network trouble shooting

- Do you have physical connectivity?
- Is your interface configured properly?
- Do your ARP tables show other hosts?
- Is there a firewall on your local machine?
- Is there a firewall anywhere between you and your destination?
- If firewalls are involved, do they pass ping packets?
- Can you ping the localhost?
- Can you ping other local hosts by IP address?
- Is DNS working properly?
- Can you ping other local hosts by hostname?
- Can you ping hosts on another network?
- Do high level services work?

# ping

- The *ping* command send an ICMP ECHO\_REQUEST packet to a target host and waits to see if the host answers back.
- Routing tables, physical networks, and gateways are all involved.
- The output for ping shows the host's IP address, the ICMP sequence number of each response packet, and the round trip travel time.
- Use `-n` option if you suspect that DNS is not working

# What information can you get from ping

- ICMP sequence number tells you if packets have been dropped. If packets have been dropped, then
  - run *tracert* to discover the route to the target host
  - *ping* the intermediate gateways in sequence
- The round trip time gives you insights into the overall performance of a path through a network.

# traceroute

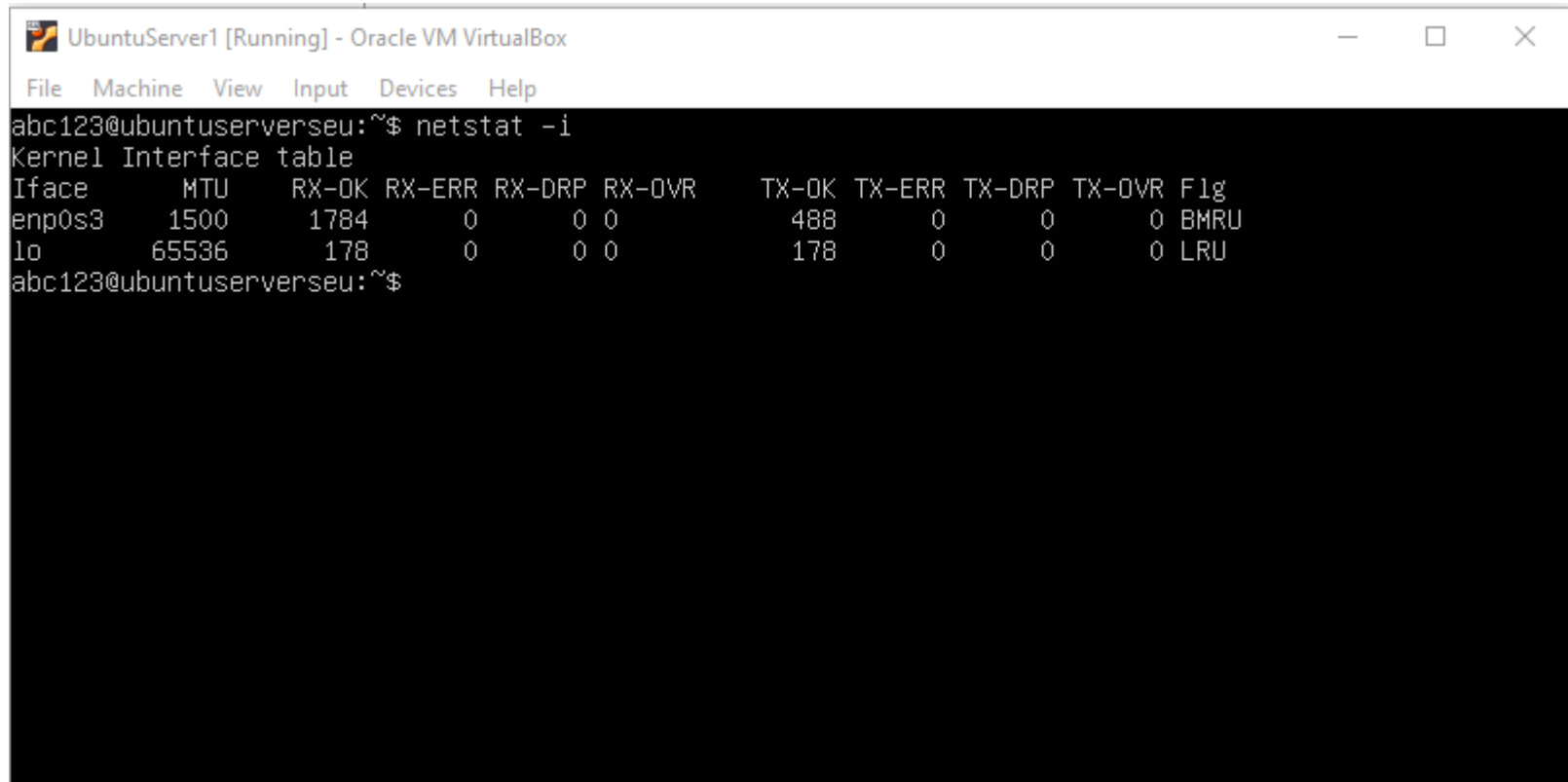
- The traceroute command uncovers the sequence of gateways through which an IP packet travels to reach its destination.
- traceroute works by setting the time-to-live field (TTL) of an outbound packet to an artificially low number.
- There are traceroute-like utilities, try *mtr*
- Inverse traceroute, see <http://traceroute.org>



# netstat

- *netstat* collects a wealth of information about the state of your computer's networking software
  - Interface statistics
  - Routing information
  - Connection tables
- *netstat* can be used to
  - Inspect interface configuration information
  - Monitor the status of network connections
  - Identify listening network services
  - Examine the routing table
  - View the operational statistics for various network protocols

# Inspecting interface configuration



The screenshot shows a terminal window titled "UbuntuServer1 [Running] - Oracle VM VirtualBox". The terminal displays the output of the command `netstat -i`, which shows the kernel interface table. The output is as follows:

```
abc123@ubuntuserverseu:~$ netstat -i
Kernel Interface table
Iface      MTU      RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
enp0s3     1500     1784   0      0 0       488    0      0      0 BMRU
lo         65536     178    0      0 0       178    0      0      0 LRU
```

The terminal prompt is `abc123@ubuntuserverseu:~$`.

UbuntuServer1 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
abc123@ubuntuserverseu:~$ ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe8c:6109 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:8c:61:09 txqueuelen 1000 (Ethernet)
    RX packets 1784 bytes 2114296 (2.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 488 bytes 54536 (54.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 178 bytes 14841 (14.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 178 bytes 14841 (14.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

abc123@ubuntuserverseu:~$ _
```

- Errors can show up. A large number of errors on a single machine suggests a problem
  - What are the error rates of the neighbouring machines?
  - Is there a problem with that machine's interface or connection?
  - Or is there a media or network problem?
- You should not see collisions even when the network is under heavy load
  - Is the flow control enabled on your switches and routers?

# Monitoring the status of network

```
UbuntuServer1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp6       0      0 [::]:ssh               [::]:*                  LISTEN
udp        0      0 localhost:domain        0.0.0.0:*               *
udp        0      0 ubuntu:bootpc           0.0.0.0:*               *
raw6       0      0 [::]:ipv6-icmp         [::]:*                  7

Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix   2      [ ]         DGRAM          LISTENING    31999    /run/user/1000/systemd/notify
unix   2      [ ACC ]     SEQPACKET    LISTENING    15035    /run/udev/control
unix   2      [ ACC ]     STREAM       LISTENING    32002    /run/user/1000/systemd/private
unix   2      [ ACC ]     STREAM       LISTENING    32008    /run/user/1000/gnupg/S.dirmngr
unix   2      [ ACC ]     STREAM       LISTENING    32009    /run/user/1000/gnupg/S.gpg-agent.extra
unix   2      [ ACC ]     STREAM       LISTENING    32010    /run/user/1000/gnupg/S.gpg-agent.ssh
unix   2      [ ACC ]     STREAM       LISTENING    32011    /run/user/1000/gnupg/S.gpg-agent.browser
unix   2      [ ACC ]     STREAM       LISTENING    32012    /run/user/1000/gnupg/S.gpg-agent
unix   2      [ ACC ]     STREAM       LISTENING    16946    @/org/kernel/linux/storage/multipathd
unix   3      [ ]         DGRAM          LISTENING    14688    /run/systemd/notify
unix   2      [ ACC ]     STREAM       LISTENING    14691    /run/systemd/private
unix   6      [ ]         DGRAM          LISTENING    14699    /run/systemd/journal/dev-log
unix   2      [ ACC ]     STREAM       LISTENING    14701    /run/systemd/journal/stdout
unix   9      [ ]         DGRAM          LISTENING    14703    /run/systemd/journal/socket
unix   2      [ ACC ]     STREAM       LISTENING    19306    /run/uuidd/request
unix   2      [ ACC ]     STREAM       LISTENING    19314    /run/snapd.socket
unix   2      [ ACC ]     STREAM       LISTENING    19316    /run/snapd-snap.socket
unix   2      [ ACC ]     STREAM       LISTENING    14891    /run/lvm/lvmpolld.socket
unix   2      [ ACC ]     STREAM       LISTENING    19339    /var/run/dbus/system_bus_socket
unix   2      [ ]         DGRAM          LISTENING    14989    /run/systemd/journal/syslog
unix   2      [ ACC ]     STREAM       LISTENING    15131    /run/lvm/lvmetad.socket
unix   2      [ ACC ]     STREAM       LISTENING    25010    /var/snap/lxd/common/lxd/unix.socket
unix   2      [ ACC ]     STREAM       LISTENING    19338    @ISCSIADM_ABSTRACT_NAMESPACE
```

- Send-Q and Recv-Q show the sizes of the local host's send and receive queues for the connection
  - These numbers should tend toward 0
- State has meaning for TCP only
  - LISTEN
  - ESTABLISHED
  - TIME\_WAIT
- netstat -a is primarily useful for debugging higher-level problems once you have determined that basic networking facilities are working correctly
  - E.g., the state SYN\_SENT identifies a process that is trying to contact a non-existent network server
  - What about SYN\_WAIT?

# Identifying listening network services

```
UbuntuServer1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

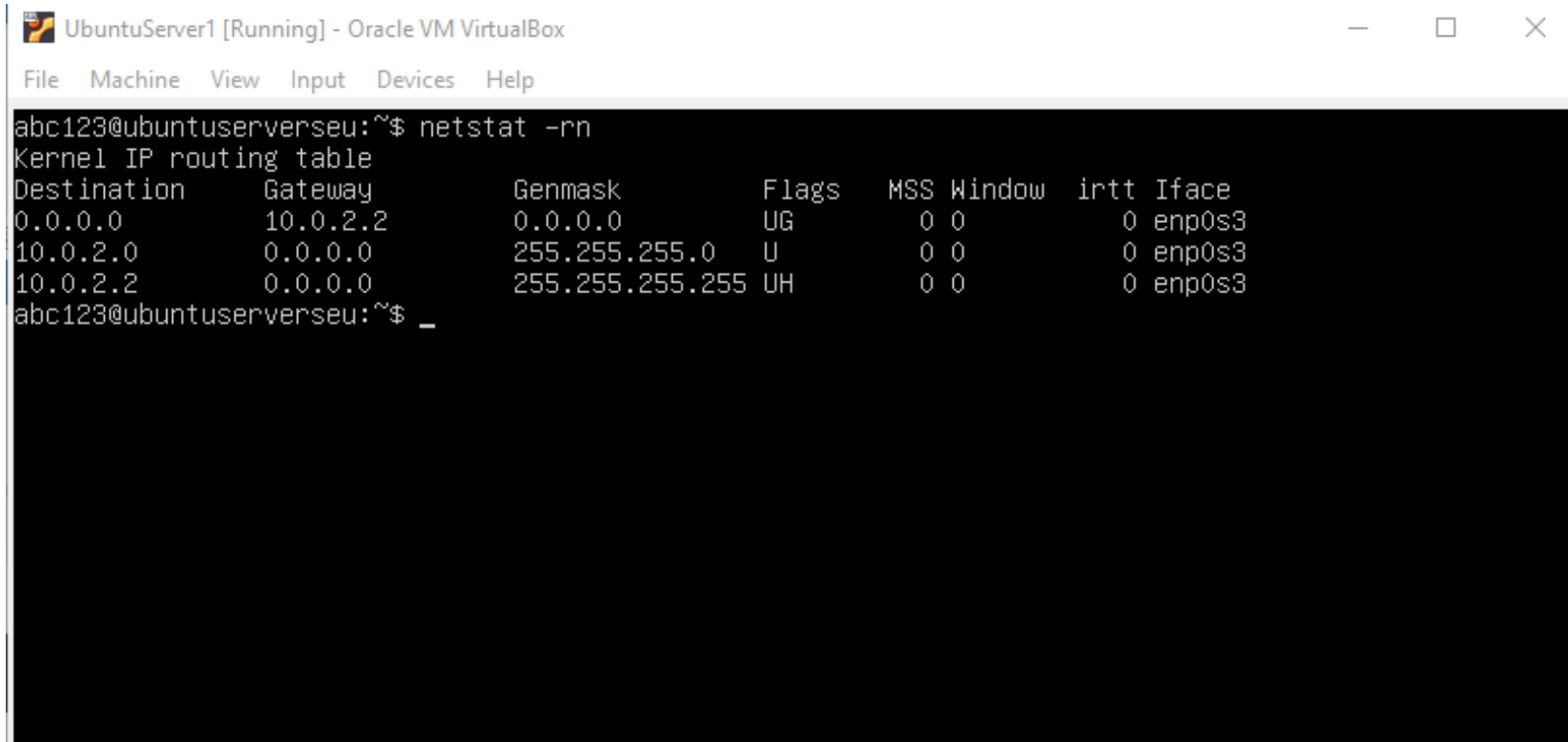
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN      -
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN      -
udp        0      0 localhost:domain        0.0.0.0:*               -          -
udp        0      0 ubuntuerverseu:bootpc   0.0.0.0:*               -          -
raw6       0      0 [::]:ipv6-icmp          [::]:*                   7          -

Active UNIX domain sockets (only servers)
Proto RefCnt Flags               Type           State         I-Node  PID/Program name  Path
unix   2      [ ACC ]     SEQPACKET     LISTENING      15035    -                 /run/udev/control
unix   2      [ ACC ]     STREAM        LISTENING      32002    2717/systemd      /run/user/1000/systemd/private
unix   2      [ ACC ]     STREAM        LISTENING      32008    2717/systemd      /run/user/1000/gnupg/
/S.dirmngr
unix   2      [ ACC ]     STREAM        LISTENING      32009    2717/systemd      /run/user/1000/gnupg/
/S.gpg-agent.extra
unix   2      [ ACC ]     STREAM        LISTENING      32010    2717/systemd      /run/user/1000/gnupg/
/S.gpg-agent.ssh
unix   2      [ ACC ]     STREAM        LISTENING      32011    2717/systemd      /run/user/1000/gnupg/
/S.gpg-agent.browser
unix   2      [ ACC ]     STREAM        LISTENING      32012    2717/systemd      /run/user/1000/gnupg/
/S.gpg-agent
unix   2      [ ACC ]     STREAM        LISTENING      16946    -                 @/org/kernel/linux/s
torage/multipathd
unix   2      [ ACC ]     STREAM        LISTENING      14691    -                 /run/systemd/private
unix   2      [ ACC ]     STREAM        LISTENING      14701    -                 /run/systemd/journal
/stdout
unix   2      [ ACC ]     STREAM        LISTENING      19306    -                 /run/uid/request
unix   2      [ ACC ]     STREAM        LISTENING      19314    -                 /run/snapd.socket
unix   2      [ ACC ]     STREAM        LISTENING      19316    -                 /run/snapd-snap.sock
et
unix   2      [ ACC ]     STREAM        LISTENING      14891    -                 /run/lvm/lvmpolld.sock
et
unix   2      [ ACC ]     STREAM        LISTENING      19339    -                 /var/run/dbus/system
_bus_socket
--More--
```

- What processes on this machine are listening on the network for incoming connections?
- It's also helpful to look at machines from an outsider's perspective by running a port scanner such as *nmap*.



# Examining the routing table



The screenshot shows a terminal window titled "UbuntuServer1 [Running] - Oracle VM VirtualBox". The terminal displays the output of the command `netstat -rn`, which shows the kernel IP routing table. The output is as follows:

```
abc123@ubuntuserverseu:~$ netstat -rn
Kernel IP routing table
Destination      Gateway          Genmask         Flags   MSS Window  irtt Iface
0.0.0.0          10.0.2.2        0.0.0.0         UG      0 0        0 enp0s3
10.0.2.0         0.0.0.0         255.255.255.0   U       0 0        0 enp0s3
10.0.2.2         0.0.0.0         255.255.255.255 UH      0 0        0 enp0s3
abc123@ubuntuserverseu:~$ _
```

- Flags characterise the route
  - U means up
  - G is a gateway
  - H is a host route
  - D indicates a route resulting from an ICMP redirect

# Is UNIX/Linux secure

- Of course not.
- UNIX is optimised for convenience and doesn't make security easy or natural.
- The software that runs on UNIX systems is developed by a large community of programmers.
- Most administrative functions are implemented outside the kernel.
- Open source distributions may have better security than that of closed operating systems.

# How security is compromised?

- Social engineering
  - Human users are the weakest links on the chain of security
  - Ensure your user community has a high awareness of security threats
- Software vulnerabilities
  - Programming errors are exploited by hackers to manipulate systems into doing whatever they want
  - Patch the software
- Configuration errors
  - Software is developed to be useful instead of annoying
  - Balance security and usability

# Security tips

- Keep the system updated with the latest patches is an administrator's highest-value security chore
- Disable any services that are not necessary
- Consider setting up a secure host to act as a central logging machine
- Regular system backups are an essential part of any site security plan
- UNIX is relatively immune from virus. Just because it is difficult does not mean it is impossible.

- A Trojan horse is a program which contains some hidden functionality.
- When an unsuspecting person executes the code, the hidden functionality does something to compromise security.
  - If I write a program to emulate ls, but also mail me a copy of /etc/shadow then somehow get my code executed as root, I can get a copy of a file I should not have been able to read. I could put my version of ls in /tmp, then hope that the root user has . (dot) in their path before /usr/bin. If the root user is in the /tmp directory and they type ls, then my code is run rather than the system ls (unlikely but possible).
  - I could use the same trick on non-root users since it is more likely they will have an insecure path.
  - Alternatively, I could mail my mates my great new program to do X, and put a Trojan horse in it to chmod their home directories to 777 and their profile to 666.

- A rabbit is a piece of code which reproduces itself at a high rate and consumes resources intended for other purposes.
- This is basically a denial of service attack.
- There are few effective defenses but it does require an active account.

- Rootkits are programs and patches that hide important system information such as process, disk, or network activity
- For example, the hacker might replace *ls* and *ps* with hacked version so that certain files and processes will be hidden
- Although tools like OSSEC and chkrootkit are available to detect the presence of known rootkits, the time it takes to do a thorough cleaning is very long



- Packet filtering and firewall are still powerful tools to block attacks from the outside world
- Remember, passwords provide some form of basic protection, simple but useful
- Be vigilant!

# General philosophy

- Effective system security based on common sense
  - Don't put files on your system that are likely to be interesting to hackers
  - Your site's security policy should specify how sensitive information is handled
  - Don't provide places for hackers to build nests in your environment
  - Set traps to help detect intrusions and attempted intrusions
  - Religiously monitor the reports generated by security tools
  - Teach yourself about system security
  - Prowl around looking for unusual activity

# Security policy

- The System Administrator will often be expected to develop or assist in the development of IT policy within their organisation.
- If the organisation does not have a policy base you need to change the organisational perspective.
- If you can't change their perspective you should consider changing your job

- Any organisation should have policy documents to cover each key operational area.
  - security policy<sup>[L]</sup><sub>[SEP]</sub>
  - acceptable use policy
  - privacy policy

# Policy development

- Policy development is important to organisations as they define formal mechanisms for handling events.
- Policy is dynamic - it will need to be revisited as organisations aspects change.
- Policy development should draw on key areas of the organization – it should not be conducted in isolation from major actors
- Policies must be documented and be well known by staff in the organisation.
- Decisions making should be guided by policy. This is particularly true in security where things can get sensitive.

# Policy examples

- Acceptable Use Policy (AUP) describes who the legitimate users of network and information technology are. It describes what are acceptable and even unacceptable behaviors. Organisations may have multiple Acceptable Use policies representing varying levels of security (typically tied to job role and seniority).
- Privacy Policy describes how the organisation monitors computer and network resources e.g mail, network traffic, web logs, logging etc. Monitoring of such data may be considered a violation of privacy. Such policy needs to conform with State and Commonwealth legislations.

- Network Connection Policy defines how resources are connected to one another inside/ outside the organisation and what services they may/ may not offer. This may have implications for business relationships.
- Remote Access Policy should describe the risks associated with accessing resources remotely. It should provide details on how to ensure the security of credentials and what to do in the event they are lost.
- Log Retention Policy should explain what data is logged and for how long. This policy may be formulated after security policy to aide in the resolution of an incident.

# Writing Policy

- First establish a basis for writing policy.
- The best way to write security policy is by asking questions.
- You should be mindful that security depends on having a solid computer and network infrastructure.
- Policy depends on the right questions being asked.



- What do we have that needs protecting?
  - Assets can be tangible or intangible e.g. hardware? data? reputation?
- What might result from a failure to protect it?
  - Risks e.g. loss of business, perception, trust etc.
- How might it be attacked and by whom?
  - What channels/ mechanisms may we be compromised e.g. network, physical security etc.
- What is the cost of protecting it? or not protecting it?
  - Can we afford to do nothing?

- The answers to those previous questions will always vary between organisations and even between organisational units.
- (Never fall into the trap of assuming organisations are the same.)
- The answers to these questions allow us to conduct a proper risk analysis.
- Once we know what the risks are we can develop an appropriate security policy.

# Assessing and prioritising risk

- Security policy is about prioritising issues according to risk.
- Identify all of the resources in the system. This includes hardware, services and data.  
Identifying threats to the above resources that if realized would cause an interruption to the normal function of the system.
- Determining the probability that each risk will occur - calculating the risk.

- The threats do not have to be malicious. In most cases they are not.
- Non-malicious threats include such things as software bugs, power failures and fire.
- A cost/benefit analysis involves:
  - Determining the likelihood that the event will occur
  - Assigning a cost to the risk.
  - How much money will be lost if the risk is realised multiplied by the probability that it will be?

- Consider the following Scenario.
- If the cost of a fire in a network closet is \$50,000 to replace equipment and cabling and the associated down time costs are \$150,000 and we can expect that there will be one fire in the closet every 10 years then we can say that the cost of the risk is \$20,000 pa.
- What typically happens is one or more strategies or measures are priced to significantly reduce the risk.
- You should choose the strategy that has the best balance between cost and reducing the risk.

# Bad and Good Policy

- Here is an example of a Bad Security Policy.

*“All passwords shall be a minimum of 8 characters in length and must include a letter, a digit and a punctuation character.”*

- Why is it bad?- This policy contains specifics and may not scale with the technology/organisational procedures.
- A good policy is one that is well worded and not bound to any particular technology/procedure.

- *“As far as is possible, machines shall be configured so as to enforce the use of passwords which are difficult to attack via brute force search methods.”*
- Here we can change our definition of difficult in a procedure without changing the policy.
- Security procedures should appear in a separate document.

# Policy vs Procedure

- Policy is usually brief and general
- Procedures will change more frequently.
- Procedures should be documented in an operations manual which may be attached to or referred to from the security policy.
- The procedures are usually longer and far more specific. They describe the how to's rather than the why's. Procedure changes when technology and operational aspects change.
- The nature of IT requires that procedures need to change often.
- Thus the procedures manual requires frequent updates without the need to refer to upper management.
- Policy does not generally change over time. It is the founding principals governing why we do things.



# The need for management support

- For a security program to succeed you need to get the support of management.
- The administrator can influence the process, but management typically has the final say *as its them who know the business imperatives*.
- Fundamentally, its a SA's job to inform them as to the possibilities, risks, threats and benefits. More than that, its important for the SA to be informed of the business imperatives and present solutions that are molded by these business requirements.
- Security should be seen as part of an organisations business capability.

# Policy models

- An organisations security stance can be categorised by one of the following:
  - Passive security.
  - Reactive security.
  - Proactive security.

# Passive security

- This is the easiest approach to take. It is also the weakest approach.
- Examples of this model include:
  - Security through ignorance
    - “what I don’t know can’t hurt me”
  - Security through obscurity
    - “what they don’t know can’t hurt me”
  - Security through complexity
    - “what I/they don’t understand can’t hurt me”

# Reactive security

- This approach is better than passive security. However, it is usually too little too late.
- Examples of this model include:
  - Shutting the stable door after the horse has bolted
    - “we won’t have that problem again”
    - “he/she won’t do that again”
- Systems administrators should design systems which are stable, scalable and secure. If you are not doing this - you are not doing your job.

# Proactive security

- Solve the security problem before it is exploited
- Hardest to implement
  - Requires the most work on the part of the SA
  - Requires the most knowledge on the part of the SA
  - Requires the most understanding on the part of the SA
- Examples of this model include:
  - Laissez Faire
    - “what is not forbidden is permitted”
  - Draconian
    - “what is not permitted is forbidden”

# More to it than writing policy

- Many people think that security is about writing policy and putting it on the shelf.
- Security depends on procedure. A good administrator will always be on the lookout for changes in the industry, technology or organisation.
- The following section outlines things you should consider when working on security.

# Logging

- Logging does not increase the security of a system per se.
- It does give SA s a tool to help you increase the security of systems and detect when a compromise has occurred.
- If you log attempts to break in you can strengthen defenses in the area under attack.

- Logging should be done to at least 2 destinations. The local machine and a relatively secure log host.
- If a person gets root on a machine they can delete the log files, but not on the log host (hopefully).
- Alternatively, they may attack the log host first in which case the local machine logs may be your only record of events if the attack fails
- With all servers logging to a central syslog server, it becomes easier to correlate events across your company.



# syslog

- On Unix logging is typically performed by *syslog*. *Syslog* allows facilities to be defined.
- The logs can be stored on a local host or forwarded to a remote host.

# Configuring syslog server

- Each system message sent to the syslog server has two descriptive labels associated with it that makes the message easier to handle:
- The first describes the function (facility) of the application that generated it. For example, applications such as *mail* generate messages with easily identifiable facilities named *mail*. [L] [SEP]
- The second describes the degree of severity of the message.

- *syslog* defines facilities which identify programs that can log.
- Facilities include:
  - auth,
  - kern,
  - mail,
  - cron,
  - lpr and
  - several others.
- *syslog* consults file */etc/syslog.conf* to configure the daemon.

# Syslog facility

Facility code	Keyword	Description			
0	kern	kernel messages	12	-	NTP subsystem
1	user	user-level messages	13	-	log audit
2	mail	mail system	14	-	log alert
3	daemon	system daemons	15	cron	scheduling daemon
4	auth	security/authorization messages	16	local0	local use 0 (local0)
5	syslog	messages generated internally by syslogd	17	local1	local use 1 (local1)
6	lpr	line printer subsystem	18	local2	local use 2 (local2)
7	news	network news subsystem	19	local3	local use 3 (local3)
8	uucp	UUCP subsystem	20	local4	local use 4 (local4)
9		clock daemon	21	local5	local use 5 (local5)
10	authpriv	security/authorization messages	22	local6	local use 6 (local6)
11	ftp	FTP daemon	23	local7	local use 7 (local7)

- The log files are usually kept in */var/log* directory
- Sample of *syslog.conf* file
  - \*.debug  
    /var/log/messages
  - The above captures all messages from debug severity and above in the /var/log/ messages file.
  - .info;mail.none;authpriv.none;cron.none  
    /var/log/messages
  - In the above, all messages of severity “info” and above are logged, but none from the mail, authentication facilities/subsystems or cron.

**Table 5.1**   `syslog` Facilities

Severity Level	Keyword	Description
0	emergencies	System unusable
1	alerts	Immediate action required
2	critical	Critical condition
3	errors	Error conditions
4	warnings	Warning conditions
5	notifications	Normal but significant conditions
6	informational	Informational messages
7	debugging	Debugging messages

- For each facility we may specify where to put the log data via a path.
- We can also discriminate on each facility using severity levels.
- Examples include
  - info,
  - err,
  - crit,
  - emerg,
  - alert.
- It is also possible to configure syslog to accept syslog messages from remotes clients

# Syslog actions

Action	Meaning
Filename	Appends the message to a file on the local machine
@hostname	Forwards the message to the syslogd on hostname
@ipaddr	Forwards the message to the syslogd on host ipaddr
fifoname	Writes the message to the named pipe fifoname
user1,user2,...	Writes the message to the screen of users if they are logged in
*	Writes the message to all users who are currently logged in



# What to log – that is the question

- Logging too much data is counter-productive since it becomes harder to identify the important log entries.
- There are things which you should always log. Below is a list:
  - Su attempts (switch user).[L][SEP]
  - Network connections (tcpd does this).
  - Failed logins.[L][SEP]
  - Rejected file system mount operations.
  - Transaction on services e.g. Mail, FTP etc.

- Typically we would use the *cron* job scheduler to check for anomalies in the logs.
- It is not practical for a SA typically to read the logs because of the volume of text.
- Whatever logging level you choose, you should automate the monitoring of logs.
- *swatch* and *logsurfer* are 2 shareware utilities for this task. They can inform you (via pager or mail) when they detect a problem.

# Never completely trust your logs

- The UNIX *syslog* daemon will accept log requests from anyone and, in the network case, from anywhere. This means people can, potentially, insert bogus log messages into your logs.
- Similarly, a hacker can flood a host with log messages to cover illicit activity.
- You should be able to detect this, however, by appropriate log filters.
- One technique is to flood a log host for a few hours to fill the filesystem where the log files are kept. Then launch an attack.
- If it succeeds, it will be possible to modify the local log files. The administrator will know that something occurred, but it will be harder to track the intruder down.

# Human factors

- You can never predict users. Users have this ability to effect security through behaviors. We have already covered passwords in a previous lecture. There are many others.
- Examples include:
  - People storing sensitive information on insecure machines for convenience.
  - Users letting their boy/girl friend use their account.
  - Users installing their own switches in their offices (not any more).

- When setting up site security, too many people focus exclusively on the external threat. They believe firewalls etc are vital, but internal security is not considered.
- It is best to assume a hostile user base even when that is not the case.

# Physical security

- As a general rule, no system is secure against an intruder who has physical access to the machine. This is often forgotten.
- An intruder can:
  - boot from alternate media.
  - remove (or copy) disks or tapes.
  - install additional hardware. (check out MIT Aaron Swartz case)
- Machine rooms need to be secure and the data they house needs to be protected.
- This can be very difficult especially in environments where you have contractors.

- Where possible you should design your machine room with this in mind.
- You should also, where possible, exploit some of the technologies in the actual machine.  
(Check out firmware password as an option)
- Servers need to be housed in secure environments.
- Auditors may insist on good physical security, as they should. In extreme cases, it can be better to destroy resources when physical security is compromised.

# Back ups

- Backup strategies are an important adjunct to system security since, if a compromise is detected, trusted data will have to be reloaded from backups.
- Once a machine is root compromised - that's it.
- You can not trust the machine and its data any more.
- However sometimes people when they attack organisations go after the backups.
- There should be procedures to make sure things work - *this is disaster recovery policy.*



# Data integrity check

- Ideally, all of the system components not required to be modified would be read only and not modifiable even by root. This is difficult to achieve.
- Integrity checking of important files one of the best ways to detect intrusion.
- The open source utility *tripwire* or *ace* compute file checksums and compares with previously computed values and thus detects when files change.

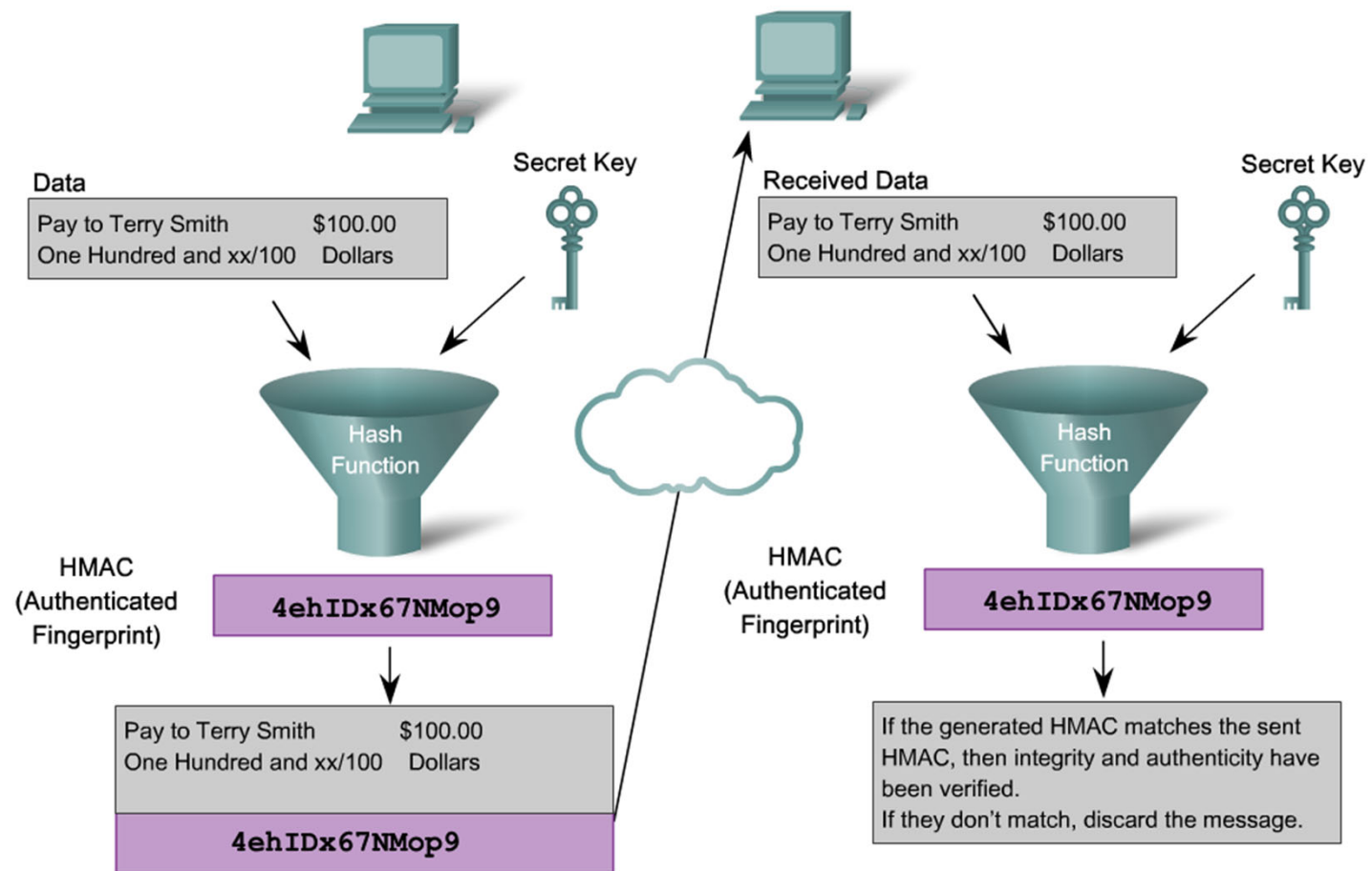
- Ideally, checksums should be stored where an intruder can not alter them (write once devices).
- Of course, the intruder may replace your version of tripwire with hacked version – or replace a shared library with a version with Trojan code but this type of intrusion is sophisticated and rare.
- Integrity checking is also a useful SA tool for detecting policy violations. Examples of this include installing software without updating software logs.

# Encryption

- Encryption is a basic tool of system security and should be used a lot more than it is.
- It is not used as much as it should be because of its high cost (in CPU) and difficulties of administration (PKI).
- Particularly sensitive information should be encrypted, especially for network transmission.

# Cryptographic hashes

- A hash function takes data (called the message) uses a cryptographic key to encode the message producing a condensed representation called the message digest
- It is hard to reverse
- Is applied in different ways:
  - To provide proof of authenticity (e.g. IP Security (IPSec) uses symmetric keys)
  - Provide authentication by generating one-time and one way response (password)
  - To provide message integrity check (digitally signed contracts) and using public key infrastructure certificates to enable secure site using a browser.



# Confidentiality through encryption

- Layers of encryption
  - Layer 2 proprietary methods
  - Layer 3 – IPSec
  - Layer 4/5 – Secure Socket Layer (SSL) or Transport Layer Security (TLS) provides session layer confidentiality
  - Application layer confidentiality – secure email, secure data base session
- Two classes of encryption algorithm
  - Symmetric algorithm – identical keys are shared between sender and receiver
  - Asymmetric algorithm - different keys are used to encrypt and decrypt messages.

# Block ciphers and stream ciphers

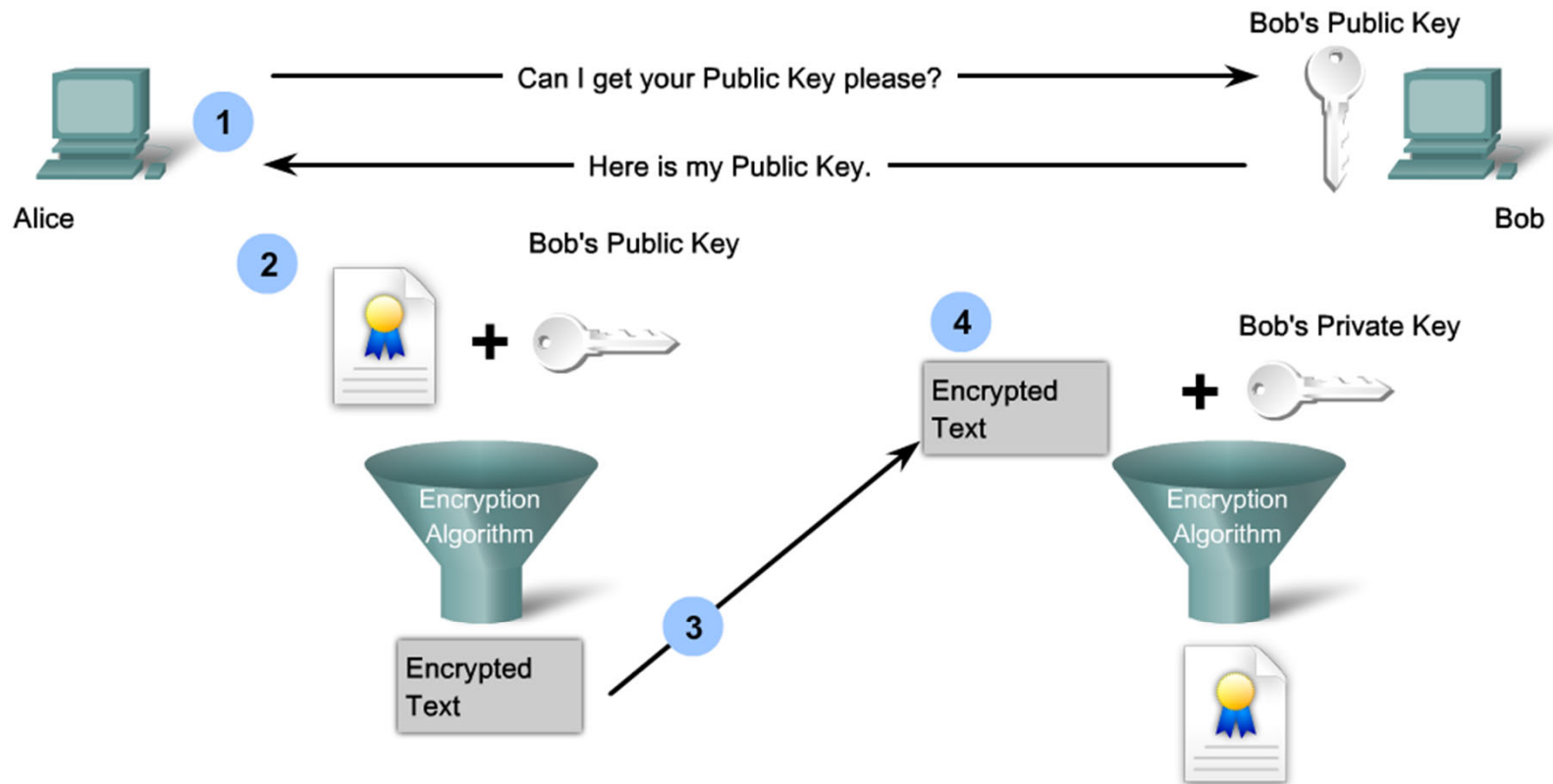
- Block ciphers transform a fixed length of plain text into blocks of cipher text 64-128 bits long. The plaintext can be retrieved by applying the secret key in reverse
- Stream ciphers encrypt text one byte at a time

# Asymmetric Encryption – Public Key Infrastructure (PKI)

- Asymmetric key algorithm does not require exchange of keys.
- Protocols that use asymmetric encryption include:
  - Internet Key Exchange (IKE) used in IPsec VPN
  - Secure Socket Layer or IETF standard TLS
  - SSH
  - Pretty Good Privacy

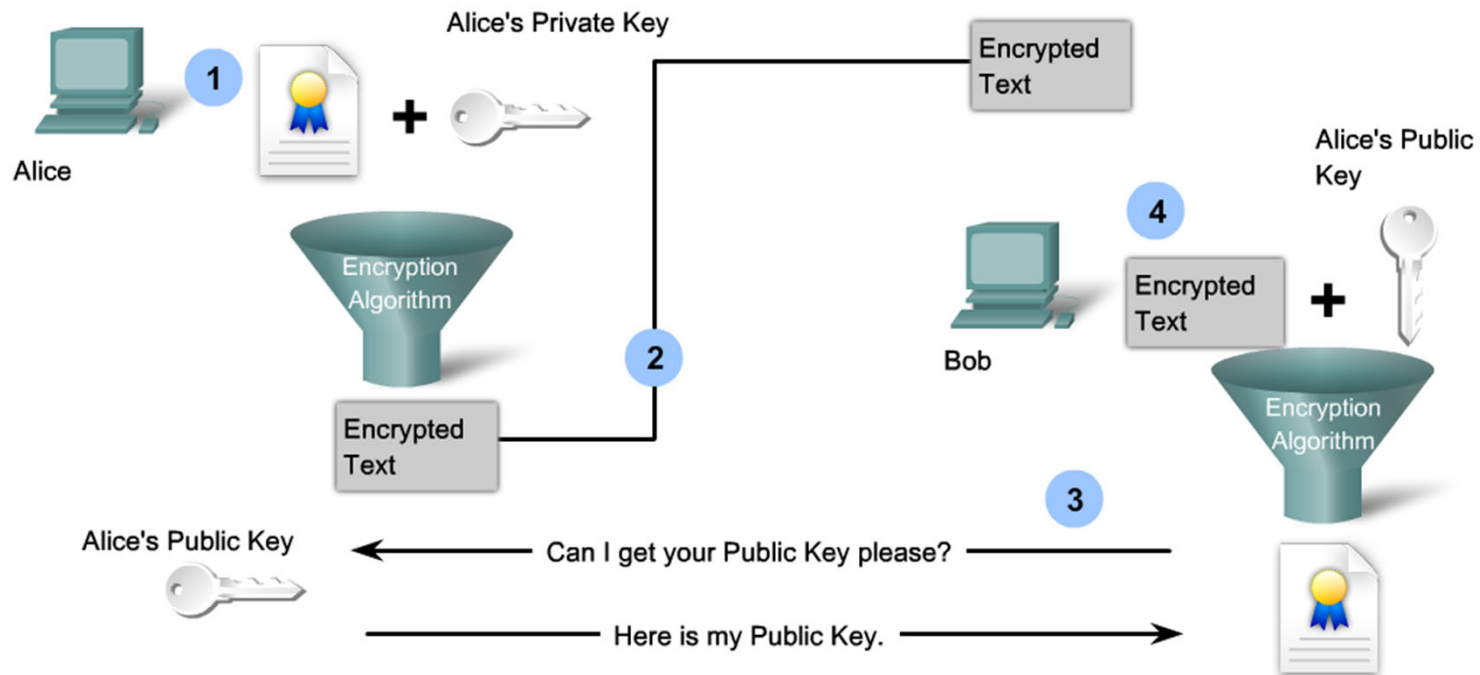


Public Key (Encrypt) + Private Key (Decrypt) = Confidentiality



1. Alice acquires Bob's public key.
2. Alice uses Bob's public key to encrypt a message using an agreed-upon algorithm.
3. Alice transmits the encrypted message to Bob.
4. Bob uses his private key to decrypt and reveal the message.

Private Key (Encrypt) + Public Key (Decrypt) = Authentication



1. Alice encrypts a message with her private key.
2. Alice transmits the encrypted message to Bob.
3. Bob needs to verify that the message actually came from Alice. He requests and acquires Alice's public key.
4. Bob uses the public key to successfully decrypt the message and authenticate that the message did, indeed, come from Alice.

# Digital signatures

- Digital signatures
  - enable unique proof of a data source which can only be generated by a single party
  - authenticate a user by using the private key of that user and the signature it generates
  - to provide authenticity and integrity of PKI certificates
  - to provide a secure timestamp using a trusted time source

# Certificate Authorities

- X.509 is a well known standard that defines basic PKI formats such as certificates and certificate revocation.
- In order to administer PKI a number of authorities exist to issue keys.
  - Single Root authorities
  - Hierarchical CA Topology
  - Cross certified CA topology.

# Trust and IP/DNS Spoofing

- Spoofing refers to the general practice of forging either sources of information or the results of queries.
- Because most of the principles of trust across networks involves knowing the source of the packets, forging packets is an effective way of bypassing system security.
- This is obviously more important as the level of trust granted a machine increases e.g. DNS Servers need to be secure leading to DNSSec

# Kerberos

- Kerberos was designed to solve some of these problems.
- Users get “tickets” which can be passed around between machines so that servers can authenticate requests.
- Tickets are just like passwords, except that they can be safely passed around the network and are immune from sniffing (because user names and password are not transferred).
- Tickets may be passed around because they are always encrypted; with keys that are only known to the authenticated users and with the current time.

# Back to the Human Aspects of Security -focussing on the SA!

- We have already covered the potential security compromises generated by users. (These include: bad choices of passwords, letting other people use their account, leave sensitive information in insecure places such as email, introducing foreign/non-authorized equipment into the corporate environment)
- System Administrator errors also play a part
  - Simple typing errors
  - Bugs in code
  - Careless system configurations<sup>[SEP]</sup>
  - Bad system default settings

- Examples of carelessness include:
  - System Administrator runs a script to dump the password and shadow files for some reason, then forgets to chmod them to 600.
  - Files left laying around containing encrypted passwords.
  - User finds and runs crack.
- These sorts of problems are much more common than attacks against encryption keys.



# What Should an Administrator Do?

- Ensure physical security.
- Stay on latest OS release. Stay on latest patch releases.
  - A large number of break-ins occur exploiting bugs that have been fixed by new OS releases or patches. Experience indicates that staying on current OS and patch levels is the single most important step in ensuring system security.
  - After OS install, audit machine security.
- Remove unnecessary services.<sup>[LSEP]</sup>
- Check permissions on important files/directories.
- Set up logging properly and monitor important logs.
- Know and control your trust relationships.

# Useful information sources

- Administrators need to keep up to date with the world. Here are some common sites/ email addresses which can keep you informed about security.

- CERT

[www.cert.org](http://www.cert.org)

- AUSCERT

[www.auscert.org.au](http://www.auscert.org.au)

# Network security

- Network security is a set of measures to mitigate risks to the network and devices using the network.
- Network security allows you to build layers of protection.
- You can erect multiple hurdles or “safety nets” to make life difficult for an attacker or to catch unexpected behavior at an early stage.

# A rationale for network security

- Are all devices on your network as secure as they should be?
- Most likely you will not have control over *every* device using the network. (Think BYOD – Bring your own device)
- Often you will not have the resources (time and knowledge) to make every device secure.
- Some devices just cannot be made sufficiently secure.
- There are certain threats that cannot be addressed without network security

# Threats

- The four basic categories of threats to networks and devices using networks are:
  - Eavesdropping<sup>[SEP]</sup>
  - Spoofing (impersonation)<sup>[SEP]</sup>
  - Modification of network communications
  - Denial of service
- These are often used in combination

# Eavesdropping

- An attacker listens to or watches the traffic on the network.
- This is a threat to confidentiality.
- The attacker may capture data that is confidential or secret.
  - (eg passwords)
- Think Wireshark...

# Spoofing

- An attacker impersonates a user or device by faking a protocol attribute that is used in an authentication process.
- This can be used to eavesdrop (man in the middle).
- Another common case is faking the source address on UDP packets – research this one for the wiki

# Modification of network communications

- An attacker hijacks communications between two devices and modifies the messages to influence the behavior of one or both parties.
- This is a threat to data integrity.



# Denial of Service attacks

- An attacker finds a way to cause a network service to stop functioning as expected
- This is a threat to accessibility.
- Often this is achieved by flooding the network with messages (classic is TCP SYN messages with fake IP source addresses).
- This is a common prelude to a spoofing attack.

# Determining what is required

- Computing security is really about ensuring that the system operates as expected (the way it was designed).
- Further, it attempts to limit the magnitude of the damage when things do go astray.
- To ensure consistency, clarity and that some body is assigned responsibility, security requirements should be codified in a “Security Policy” document.
- What level of security does your network require?
- A systematic method for answering this question is to conduct and risk analysis followed by a cost/benefit analysis.
- We have established this is an important step.

# Network security measures

- Network security measures can be implemented:
  - at the end host,
  - networking devices (routers, switches, etc)
- Generally there are two things that can be done to improve security.
  - Restrict who and what can access your network (both physically and otherwise).
  - Being careful about how data is transmitted on the network.

# Measures for end devices

- It makes sense to restrict access to services to only the people that require them (and that hopefully can be trusted).
- One method of limiting access to services is to use authentication.
  - Passwords.<sup>[L]</sup><sub>[SEP]</sub>
  - Public key authentication (signatures).
  - One shot passwords.<sup>[L]</sup><sub>[SEP]</sub>
  - Kerberos

# xinetd

- Use of *xinetd* can limit access to services (applications) through ACL.
- *xinetd* - listens for incoming requests over a network and launches the appropriate service for that request. Requests are made using port numbers as identifiers and *xinetd* usually launches another daemon to handle the request.
- *xinetd* features access control mechanisms such as TCP Wrapper ACLs, extensive logging capabilities, and the ability to make services available based on time. It can place limits on the number of servers that the system can start, and has deployable defense mechanisms to protect against port scanners, among other things.

# Measures for network devices

- Physical security is important when it comes to networks for a number of reasons:
  - Network infrastructure is often quite large and covers a large area.
  - Attacks can be made against the network if the attacker has physical access to the infrastructure.
  - Devices should be physically secured and authenticated secure.

# Data integrity

- Many protocols have been designed without much thought to security.
- This is true for IP in general.
- Most older protocols are susceptible to eavesdropping, spoofing, insertion attacks and denial of service.
- Rear guard actions include
  - Secure DNS – DNSSec
  - Secure IP - IPSec
  - Secure Shell – SSH (replaces Telnet)

# Virtual Private Networks

- A virtual private network is a tunnel over an insecure network connecting two or more trusted networks using the techniques just described.
- Specially designed routers are normally used to construct a VPN
- Two types of VPN –
  - Site to site VPN – VPN tunnel (encrypted) established between two sites using the public internet.
  - Remote Access VPN – Host has VPN client software which uses the public Internet to send it to the destination network VPN gateway.



# Packet filtering

- Access control lists
- Networking devices that understand the transport protocols and sometimes higher (typically routers) can decide which individual protocol data units (packets) it will allow into or out of the network.
- This can be used to permit or deny access to services based on rules.

# Firewalls

- Firewalls are systems that exercise a high level of control over what data enters and leaves a network.
- Typically they involve a couple of packet filters and one or more application proxies (commonly referred to as bastion hosts).
- Their role is to drastically reduce the ways an intruder can use to attack your systems.

# Intrusion Detection Systems

- Firewalls are not effect in protecting networks from worms, viruses malware etc.
- Analyzing logs is one measure but is after the event, complex and not scale.
- IDS passively monitors traffic by analyzing a copy of the traffic stream.
- Advantage is that IDS does not negatively effect actual traffic flow.
- Its disadvantage is that cant stop malicious attacks from reaching the target before being able to respond

# Intrusion Prevention Systems (IPS)

- In contrast IPS in contrast does not allow packets to enter the network until it has been analyzed.
- An IPS monitors Layer 3 and 4 traffic and analyses the content and the payload for embedded attacks.
- For example, Cisco's IPS uses a combination of detection technologies, signature-based, profile-based and protocol analysis.
- Advantage is that single-packet attacks can be foiled
- Disadvantage is that network performance can suffer if not configured properly or inappropriate choice of IPS has been made.

# Incident response plan

- Somehow you've detected that an attack has occurred or is occurring now!
- What are you going to do about it? Be prepared.
- There are several options available when you discover an attack. The right decision will depend on the type of attack and the business requirements of your organization.
  - Is it more important to get the systems back to normal?
  - Is it important to attempt to discover who the intruder or attacker is?
  - Is it important to work out exactly what the attacker did?
  - Does evidence need to be collected for legal reasons?
  - Who should be notified? Report the incident to the authorities or keep it a secret?
- Devise and communicate a recovery plan. It is important to explain what happening (but where appropriate you should omit details).