Overview

- Last Lecture
  - – Wireless networking
- This Lecture
  - – Scheduled tasks, log management, and authentication protocols
- Next Lecture
  - – DNS
  - – Readings:
    - Chapter 6 and 13 in Linux Network Administrator's Guide
    - DNS & BIND (O'Reilly)

Daemon

- A process that runs in the background and is independent of control from all terminals
- Reasons for daemons' independence of terminals
  - Prevent daemons' error message from appearing on a user's terminal
  - Signals generated from terminal keys must not affect any daemons that were started from that terminal earlier
- Typical daemons
  - **crond**, **syslogd**

Scheduled tasks

- Automating tasks
  - crond and crontab
  - crond is a very important daemon for automatically executing tasks
  - Tasks can be configured to repeat hourly, daily, weekly, …, or even per minute.
- Possible uses
  - Clean file systems
  - Log rotate
  - Check log files
  - Monitor system status and resources
  - …

syslogd

- How it works?
  - Read the configuration file */etc/syslog.conf*
  - A Unix domain socket is created and bound to the pathname */var/run/log*
  - A UDP socket is created and bound to port 514
  - Runs in an infinite loop that calls *select*, waiting for any one of the above descriptors to be readable, reads the log message, and does what the configuration file says to do with that message.
  - If the daemon receives the SIGHUP signal, it rereads the configuration file

Logging functions

- How start and close logging?
  - Create a Unix domain datagram socket and send out messages to the pathname the daemon has bound, or send them to port 514 by a UDP socket
  - *openlog()* and *closelog()*
- How to send log messages
  - *void syslog(int priority, const char *message, …);*
  - *priority* is a combination of a *level* and a *facility* shown later
  - *message* is like a format string to *printf*, with the addition of a *%m* specification, which is replaced with the error message corresponding to the current value of *errno*.

Level

- Level
  - LOG_EMERG (0): system is unusable
  - LOG_ALERT (1): action must be taken immediately
  - LOG_CRIT (2): critical conditions
  - LOG_ERR (3): error conditions
  - LOG_WARNING (4): warning conditions
  - LOG_NOTICE (5): normal but significant conditions
  - LOG_INFO (6): informational
  - LOG_DEBUG (7): debug-level messages, has lowest priority

Facility

- Identify the type of process sending the message
- Facilities
  - LOG_AUTH: security/authorization messages
  - LOG_AUTHPRIV: security/authorization messages (private)
  - LOG_CRON: from *crond*
  - LOG_KERN: kernel messages
  - LOG_MAIL: mail system
  - LOG_USER: random user-level messages (default)
  - LOG_FTP: from *ftpd*
  - LOG_LPR: line printer system
  - LOG_SYSLOG: internal messages from *syslogd*
  - LOG_LOCAL0 – LOG_LOCAL7: local, discretional use by programmers.

klogd

- klogd provides a facility for system admin to check only kernel messages (which can also be checked through syslogd)
- Kernel messages can be read from **/proc/kmsg**
- Use **/proc/sys/kernel/printk** to control the level of log messages.
  - **cat /proc/sys/kernel/printk**

syslog.conf

- Syslogd configuration file
  - /etc/syslog.conf
  - Consists of <facility>.<priority> <target> entries
    - mail.*   /var/log/maillog
    - authpriv.*   /var/log/secure
    - *.alert   root, mal
  - Use "*man 5 syslog.conf*" to find more information about the format of the file

Log processing

- Log scanning and filtering
  - Scanning: use scripts (put as a cron job) to scan key words in log files
  - Filtering: use scripts to remove useless messages from the log files
- Pros and cons of scanning and filtering
  - Scanning: can find useful information, but may have to process a large amount of log files
  - Filtering: can reduce the amount of log files but may miss some useful information.

Log processing (cont.)

- Log rotation
  - Use *logrotate* command
    - logrotate is designed to ease administration of systems that generate large number of log files. It allows automatic rotation, compression, removal, and mailing of log files. Each log file may be handled daily, weekly, monthly, or when it grows too large
  - Configuration file:**/etc/logrotate.conf** (see the manual page for logrotate)
  - Run logrotate as a cron job

Log processing (cont.)

- Store log files in computer archive
  - Legal issues regarding how long log files should be stored.
  - How to process a huge amount of log files efficiently?
- Risks of log management
  - Log files can be changed (MD5?)
  - Log files can be exposed while being transmitted (encryption?)

Cloud services for sharing?

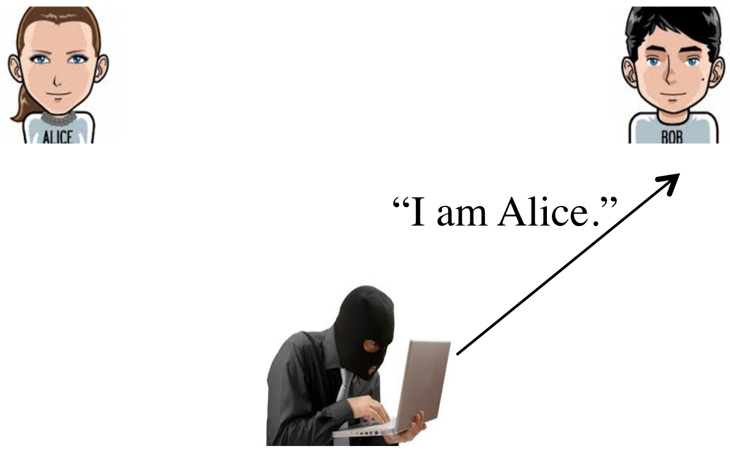
- Privacy issue
- Issue on confidential information
- Who owns the data?
- Enterprise cloud is recommended for sharing confidential documents inside an organization.
- git and svn are good tools for sharing.

Authentication Protocols (1)

- Scenario
  - Alice, the sender, wants to communicate with Bob, the receiver
  - Bob wants Alice to “prove” her identity to him
  - Trudy tries to pretend to be Alice
- AP1.0
  - Alice sends a message to Bob saying she is Alice

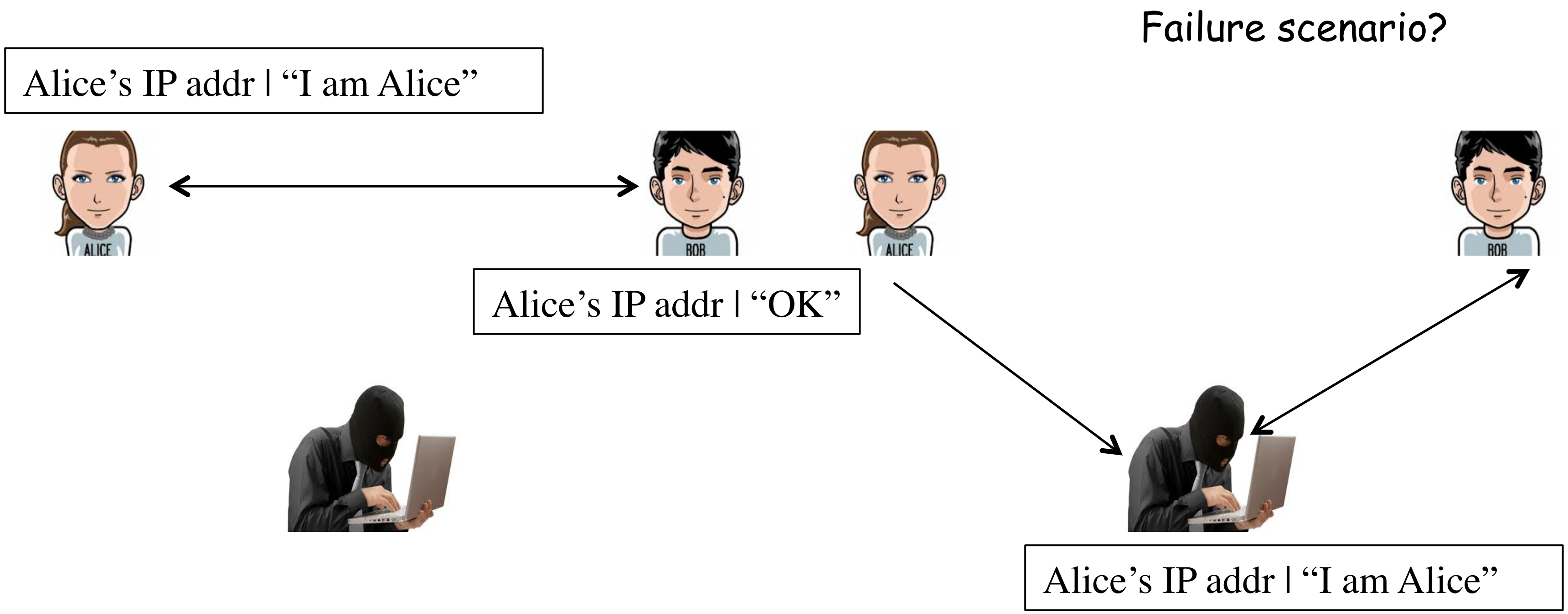
“I am Alice.”
ALICE
BOB

Failure scenario?



Authentication Protocols (2)

- AP2.0
  - Use the source IP address to authenticate
  - Fails if IP spoofing is used



Failure scenario?

Alice's IP addr | "I am Alice"

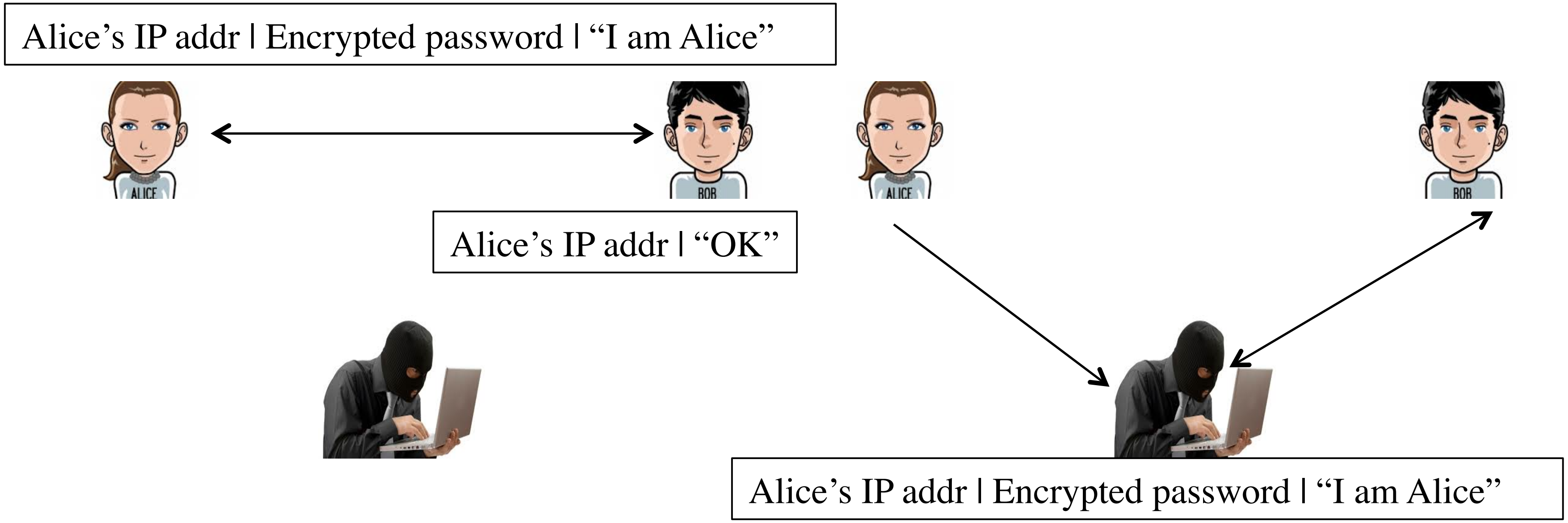
Alice's IP addr | "OK"

Alice's IP addr | "I am Alice"

Authentication Protocols (3)

- AP3.0
  - Use secret password
  - Password can be eavesdropped
  - Encrypted password can be played back

Failure scenario?



Alice’s IP addr | Encrypted password | “I am Alice”

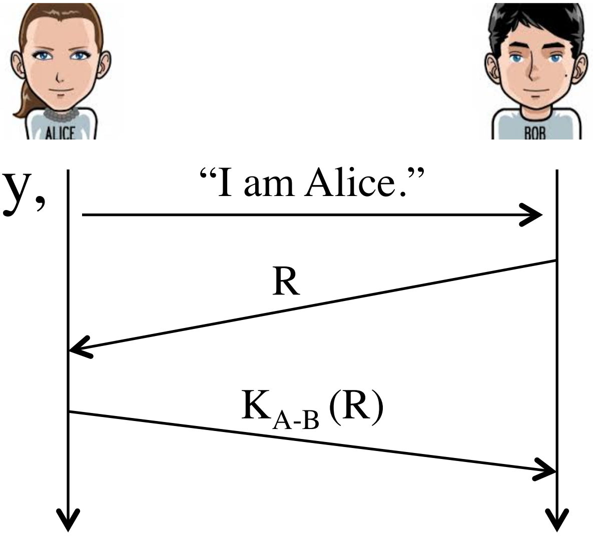
Alice’s IP addr | “OK”

Alice’s IP addr | Encrypted password | “I am Alice”

Playback attack

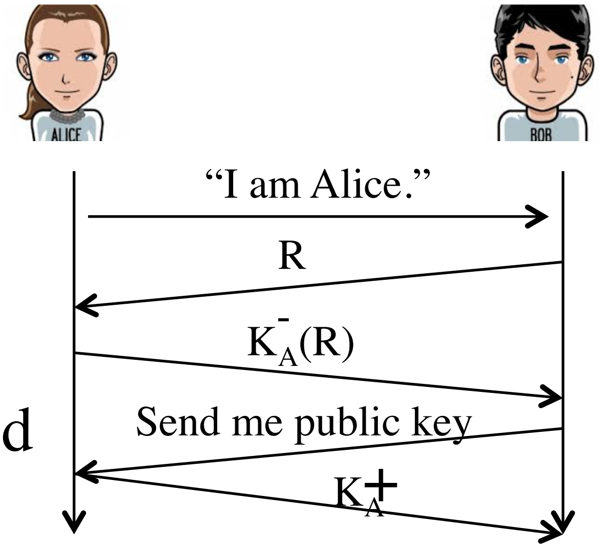
Authentication Protocols (4)

- AP4.0
  - – Use a number, called a nonce, that will be used only once in a lifetime
  - – The protocol has the following procedures
    - Alice sends "I am Alice", to Bob
    - Bob sends a nonce, R, to Alice
    - Alice encrypts the nonce using Alice and Bob's symmetric secret key, $K_{A\text{-}B}$, and sends it back to Bob
    - Bob decrypts the received message. If the decrypted nonce equals the nonce he sent Alice, then Alice is authenticated.
  - – Key distribution can be a logistic problem.



Authentication Protocols (5)

- AP5.0
  - Use the public key encryption in AP4.0
  - The protocol has the following procedures
    - Alice sends "I am Alice", to Bob
    - Bob sends a nonce, R, to Alice
    - Alice encrypts the nonce using Alice's private key A and sends the encrypted nonce back to Bob
    - Bob decrypts the received message using Alice public key. If the decrypted nonce equals the nonce he sent Alice, then Alice is authenticated.



Bob computes

$K_A^+(K_A^-(R)) = R$

The retrieval of the public key could be a security hole

Summary

- Pros and cons of filtering and scanning in log processing
- Why should daemons use log files to print error messages?
- Pros and cons of using a cloud service
- Authentication protocols