Introduction to Malicious Code (Malware, part II)

EDA 263 – Computer Security

Original Slides: Erland Jonsson
Changes by Magnus Almgren

Story: The Morris Worm

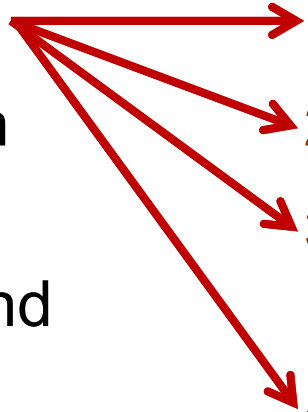
- November 3, 1988: launch of worm
  → 6,000 computers shut down (in the U.S. only)
- Internet like a small town – 100,000 computers (?)
  where people knew and trusted each other.
- Many features not built with security in mind.
  - ”doors left unlocked”
  - Internet security – mostly theoretical problem
  - What was there to protect?
- The worm changed the landscape!
  - Wakeup call that security is important!
  - Creation of CERT:s, demand for security experts (academia, industry)
- Over 25 years later, some of the same strategies still work …

http://www.washingtonpost.com/blogs/the-switch/wp/2013/11/01/how-a-grad-student-trying-to-build-the-first-botnet-brought-the-internet-to-its-knees/

The Morris Worm – Steps

Principle for function

A. Intrusion
B. Transfer of main program
C. Settling down and establishing (cracking accounts, hiding, etc)
D. Continued intrusions



Details (4 well-known attacks)

1. finding trust relations
2. guess/crack passwords
3. use debug facility in the sendmail mail handler
4. exploit bug in finger program (buffer overflow)

Finding trust relations

- The worm obtains host addresses by examining
  - the system tables */etc/hosts.equiv* and
  - */.rhosts*,
  - user files like *.forward*
  - dynamic routing information produced by the *netstat*, and finally
  - randomly generated host addresses on local networks.
- It ranks these by order of preference, but what does it mean?

Finding trust relations

- The worm obtains host addresses by examining
  - the system tables */etc/hosts.equiv* and
  - */.rhosts*,
  - user files like *.forward*
  - dynamic routing information produced by the *netstat*, and finally
  - randomly generated host addresses on local networks.
- It ranks these by order of preference, but what does it mean?

The /etc/hosts.equiv File
The /etc/hosts.equiv file contains

a list of trusted hosts for a remote system.

If a user attempts to log in remotely (using rlogin) from one of the hosts listed in this file, and if the remote system can access the user's password entry, the remote system allows the user to log in

without a password.

Finding trust relations

- The worm obtains host addresses by examining
  - the system tables */etc/hosts.equiv* and
  - */.rhosts*,
  - user files like *.forward*
  - dynamic routing information produced by the *netstat*, and finally
  - randomly generated host addresses on local networks.
- It ranks these by order of preference, but what does it mean?
- It contains names of local machines that are likely to permit **unauthenticated connections.**

Guess/crack passwords

- **Assumption**: *A user is using the same passwords on all systems*
- Crack local password file
  - Each user's account name and simple permutations of it
  - A list of 432 built-in passwords that Morris thought would be likely
    - aaa cornelius guntis noxious simon academia couscous hacker nutrition simple aerobics creation hamlet nyquist singer airplane creosote handily oceanography single albany cretin happening ocelot smile
  - All the words in the local system dictionary

- So are people better today with their passwords?

Use debug facility in the sendmail

- "trap door" in the *sendmail* SMTP mail service,
- A bug in debugging code allows the daemon to to execute a command interpreter and download code across a mail connection.

- Buffer overflow to come after the break

Internet Worm – Establishing

- **(B) Program transfer**
  - After the intrusion the program (~200 Kbytes) was transferred in a secure way (!)
- **(C) Establishing**
  - guess/crack passwords (root password was not utilised!)
  - camouflage activities (fork, simple EOR-encryption, no copy left on disk) *Compare with: stealth viruses*
  - one-time password for program transfer
- **(D) Continued Intrusions**
  - New machines were infected. There were facilities in the code to avoid multiple infections, but they did not work.

    There can also be bugs in malware…

    Thus, the main result was that the computers/network were overloaded.

CIA – **an availability failure**

Internet Worm – Establishing

- **(B) Program transfer**
  - After the intrusion the program (~200 Kbytes) was transferred in a secure way (!)
- **(C) Establishing**
  - guess/crack passwords (root password was not utilised!)
  - camouflage activities (fork, simple EOR-encryption, no copy left on disk) *Compare with: stealth viruses*
  - one-time password for program transfer
- **(D) Continued Intrusions**
  - New machines were infected. There were facilities in the code to avoid multiple infections, but they did not work.

There can also be bugs in malware…

Thus, the main result was that the computers/network were overloaded.

CIA – an availability failure

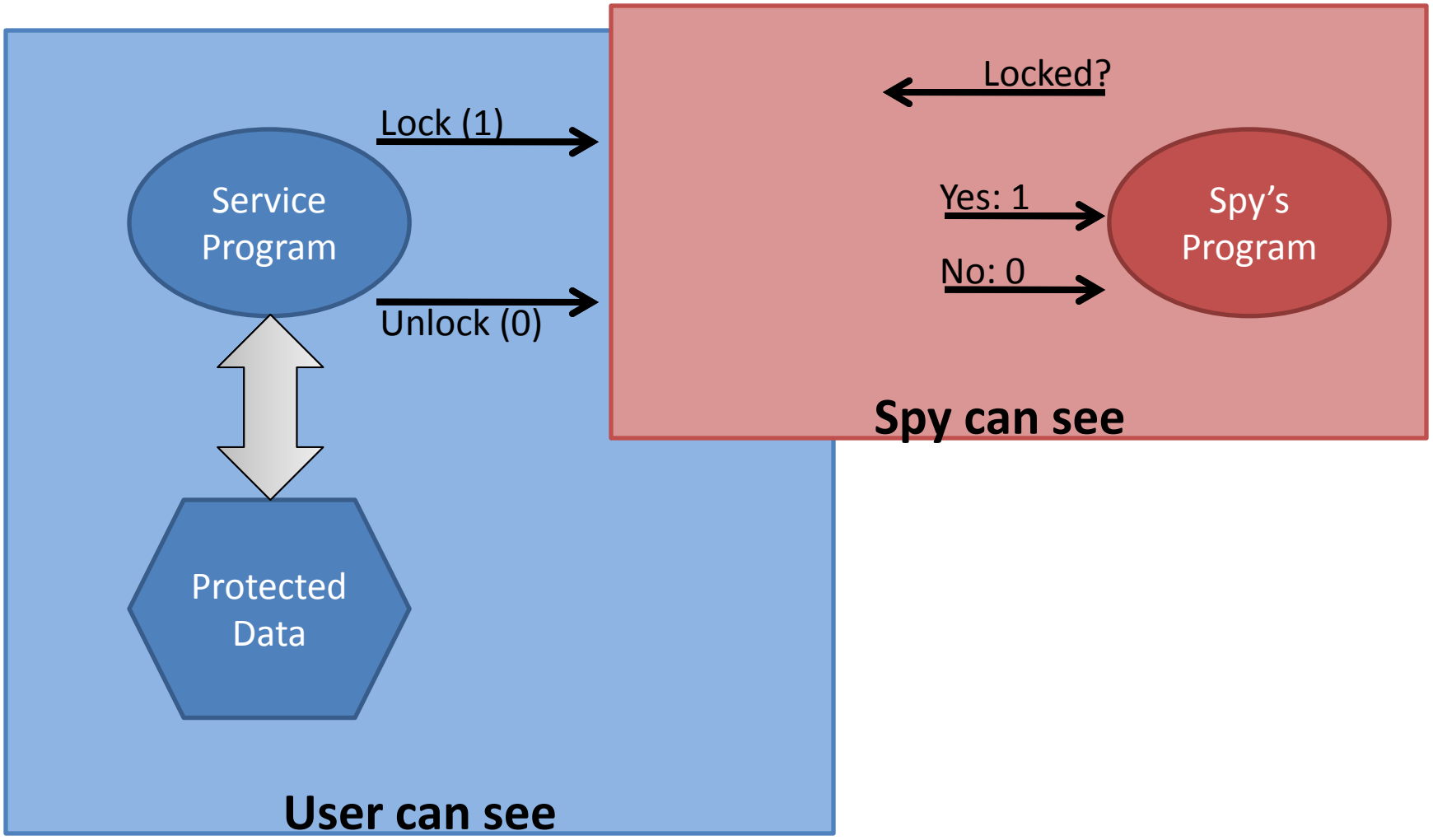
Covert Channel Basics

- a **covert channel** is a channel that leaks information from a protected area (module/program) to an unprotected area. Also called **leakage path** (swedish: hemlig kanal/dold kanal)
- its most important characterization is **bandwidth** (bits/s)
- covert channels can make use of almost any means for the information transfer
- a typical environment is a highly sensitive system
- Cmp steganography (“hidden writing”), watermarking and fingerprinting

Covert Channel Types
Storage Channels

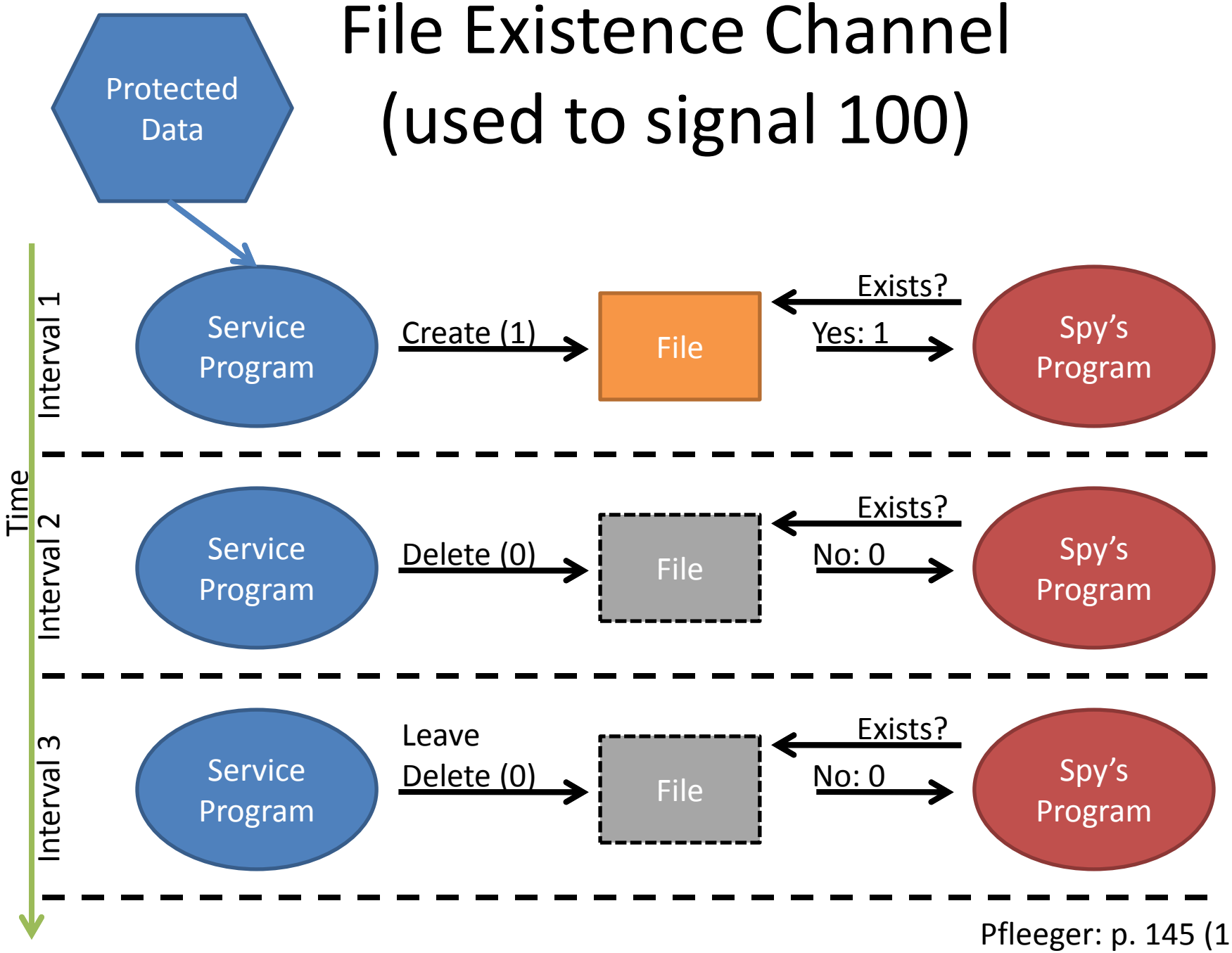
- Two main types: storage and timing channels
- **A. storage channels:**
  Eg. process 1 writes to an object and process 2 reads it
  - **A1: object attributes**:
    file attributes (length, format, date of change, ACL,…)
  - **A2: object existence**:
    check the existence of a certain file
  - **A3: shared resources**:
    use printing queue (full or empty)

File Lock Covert Channel



Pfleeger: p. 144 (154)

File Existence Channel (used to signal 100)



Pfleeger: p. 145 (155)

Example Covert Channel

Number of spaces after :

Number of lines per page

Use of "." or ":"

Last digit in field is insignificant.

```
                                UT COMPUTING CENTER
                                    AUDIT TRAIL
                                     03/04/87
                                                                   PAGE:
ACCOUNT CODE: 040099 DEP. NO: 125     CONSULTANT: JOE NICER

                                            Y MODEL/3081 ***

              NAME CPU# PGMER#      CCRE- CPU 3330-DISK -3380    TAPE   READER  PAGES  PRINTER
TI    CLASS PROGRAMMER-NAME    PI    ER CCRE-EXCP 3350-        TP     3480 LOCATION CARDS    PUNCH
2/1  878217 PROJECTI MVS1 007549  .0000     0.00        0     0      0       29      2       29
13.29.56(P) GREEN              0.0000     0.00        0     0      0L31.SR1        0        0
            2/15/87  13.29.48 FCB-6UCS-GNFORM-0316UNIT-COST-0.0110 UNITS-      2 COST-    0.022
2/15/878217 PROJECTI MVS1 007549 0.0000     0.00        0     0      0       29      2       29
13.29.56(P) GREEN              0.0000     0.00        0     0      0L31.SR1        0        0
            2/15/87  13.29.48 FCB-6UCS-GNFORM-0316UNIT-COST-0.0110 UNITS-      2 COST-    0.022
2/15/878217 PROJECTI MVS1 007549 0.0000     0.00        0     0      0       29      2       29
13.29.56(P) GREEN              0.0000     0.00        0     0      0L31.SR1        0        0
            2/15/87  13.29.48 FCB-6UCS-GNFORM-0316UNIT-COST-0.0110 UNITS-      2 COST-    0.022
2/15/878217 PROJECTI MVS1 007549 0.0000     0.00        0     0      0       29      2       29
13.29.56(P) GREEN              0.0000     0.00        0     0      0L31.SR1        0        0
            2/15/87  13.29.48 FCB-6UCS-GNFORM-0316UNIT-COST-0.0110 UNITS-      2 COST-    0.022
```

Pfleeger: p. 143 (153)

Covert Channel Types
Timing Channels

- Two main types: storage and timing channels
- **B. timing channels**
  E.g. process 1 creates some "effect" and process 2 measures time.
  - Examples:
    - vary the CPU load in e.g. 1 ms intervals (works well if only 2 processes)
    - make program execution dependent on program data

- Timing channels tend to be noisy and hard to detect.
- Countermeasure:
  - deny access to system clock (but: it is possible to make your own clock)

Information Hiding Basics

- **information hiding** is a general concept that includes
  - steganography (covert communication) and
  - (digital) watermarking.
- steganography
  - means "*hidden writing*" (as does cryptography), but here it is the ***existence*** of the message that is secret.
  - steganography "embeds a secret message in some carrier, such as an open message".
- (digital) watermarking
  - means embedding a message into a cover message, normally to discourage theft of intellectual property rights (IPR).
  - Example: media watermarking:
- cover = digital image, secret = copyright notice

Practical Steganography (1)

- Steganography was used in WWII:
  - Germans used hem stitching patterns to hide Morse Code.
  - Invisible ink, indentation etc. were also used.

http://www.washingtonpost.com/wp-dyn/content/article/2006/09/03/AR2006090300811.html

Practical Steganography (2)

Randolph Femmer /life.nbii.gov

Practical Steganography (3)

Randolph Femmer /life.nbii.gov

First chapter of “Around the world in eighty days”, Jules Verne

http://utilitymill.com/utility/Steganography_Encode

Lenny Domnitser

Practical Steganography (4)

- It is also possible to hide an image within another image.

By removing all but the last 2 bits of each color component, an almost completely black image results. Making the resulting image 85 times brighter results in the following.

http://en.wikipedia.org/wiki/Steganography

Summary

- A *covert channel* allows an inside malicious process to send sensitive data to an outside receiver, using an existing baseline communication band.
- Contrary, *steganography* presents the communication in clear sight, but in a form that is not likely to be noticed (instead of hiding it).
- *Cryptography* will be introduced in a later lecture. Here the content is concealed but the existence of the encrypted data is visible to all.

From *Analyzing Computer Security, Pfleeger, Lawrence Pfleeger*

ieee spectrum
INSIDE TECHNOLOGY
MAGAZINE MULTIM
AEROSPACE BIOMEDICAL COMPUTING CONSUMER ELECTRONICS ENERGY

TELECOM / INTERNET

FEATURE

Vice Over IP: The VoIP Steganography Threat

A growing cadre of criminals is hiding secret messages in voice data

By JÓZEF LUBACZ, WOJCIECH MAZURCZYK, KRZYSZTOF SZCZYPIORSKI / FEBRUARY 2010

Image: Mick Wiggins

http://spectrum.ieee.org/telecom/internet/vice-over-ip-the-voip-steganography-threat/