Introduction to Malicious Code (Malware, part II)

EDA 263 – Computer Security

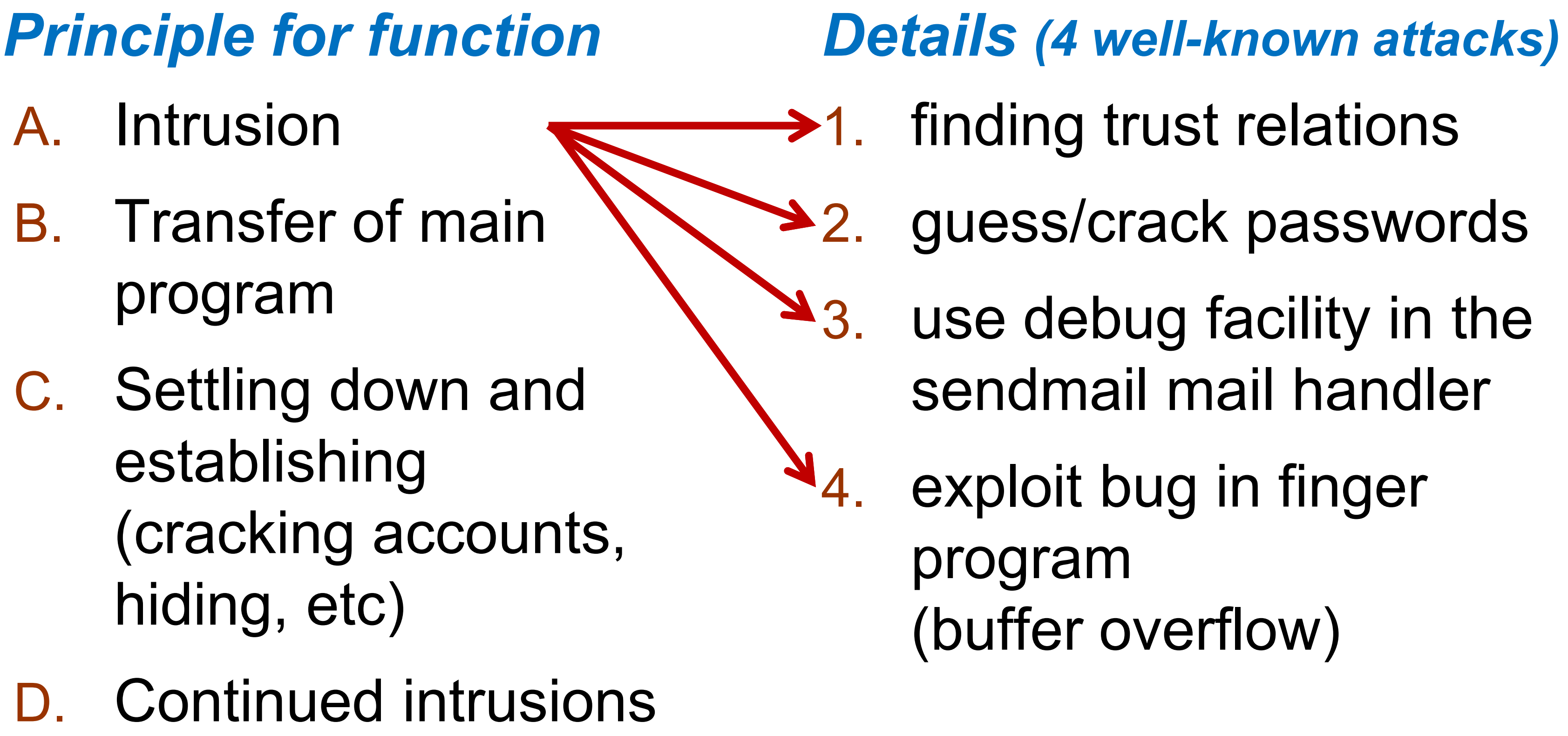
Original Slides: Erland Jonsson
Changes by Magnus Almgren

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.

CI*A* – 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.

CI**A – an availability failure**