Defenses

EECE6029

Yizong Cheng

3/14/2016

Firewall

- A wall or a bridge so that all things in and out must go through.
- Who are in and out can be monitored.
- What is allowed can be specified with rules.
- stateless firewall: rules based on header of packet, including IP addresses, port numbers, protocol
- stateful firewall: from the TCP header, one also knows the connection establishment state.
- firewall implementing an intrusion detection system (IDS) by checking the packet contents.

Addresses, Ports, and Firewall

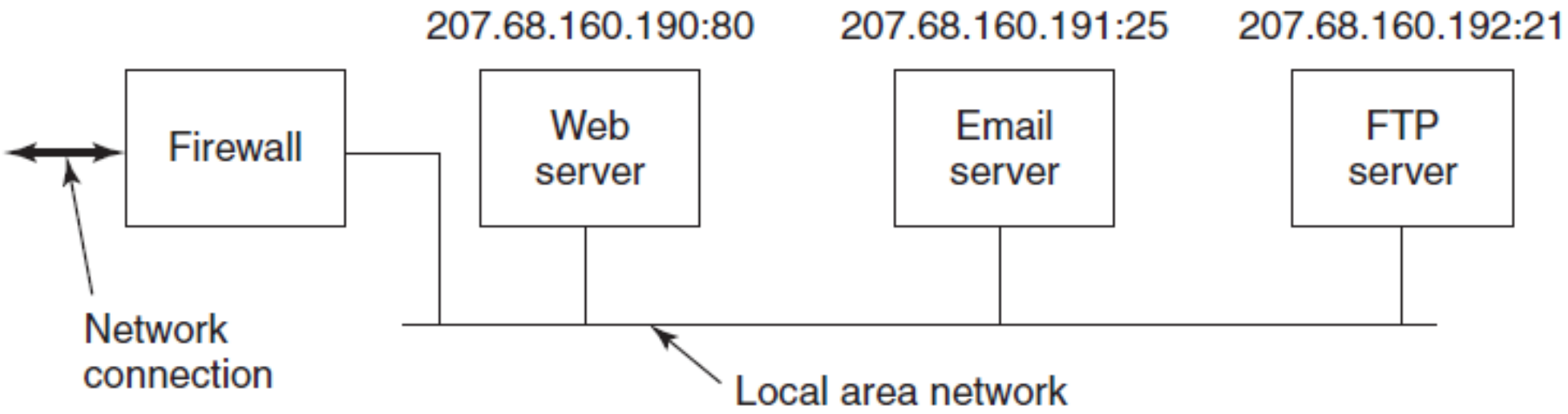


Figure 9-32. A simplified view of a hardware firewall protecting a LAN with three computers.

Taking Over a Web Server inside the Firewall

- Sending a very long URL to a Web server may force a buffer overflow and the takeover of the computer and maybe all computers inside the firewall.

Code Signing

- Run only unmodified software from reliable software vendors.
- Digital signature with public-key cryptography.
- Public key of vendor often accompanies the download, along with a certificate from some certification authority (CA).

Code Signing

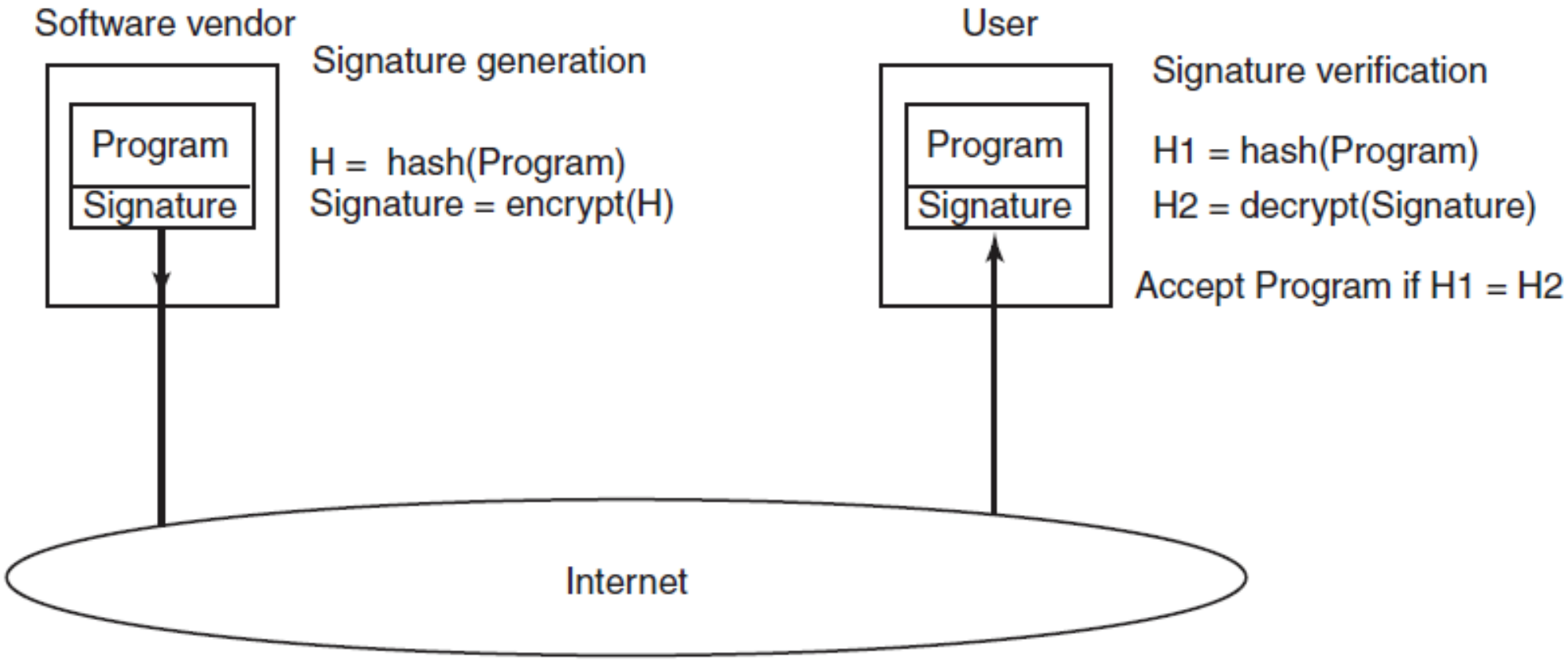


Figure 9-35. How code signing works.

Jailing and Honeypot

- A newly acquired program should be run as a process labeled “prisoner” whose system calls should all go to a trusted process called the “jailer”.
- The jailer makes a decision about whether the system call should be allowed.
- The normal UNIX debugger performs exactly this.

- Some intruder detection systems use honeypot to trap malware.

Jailing

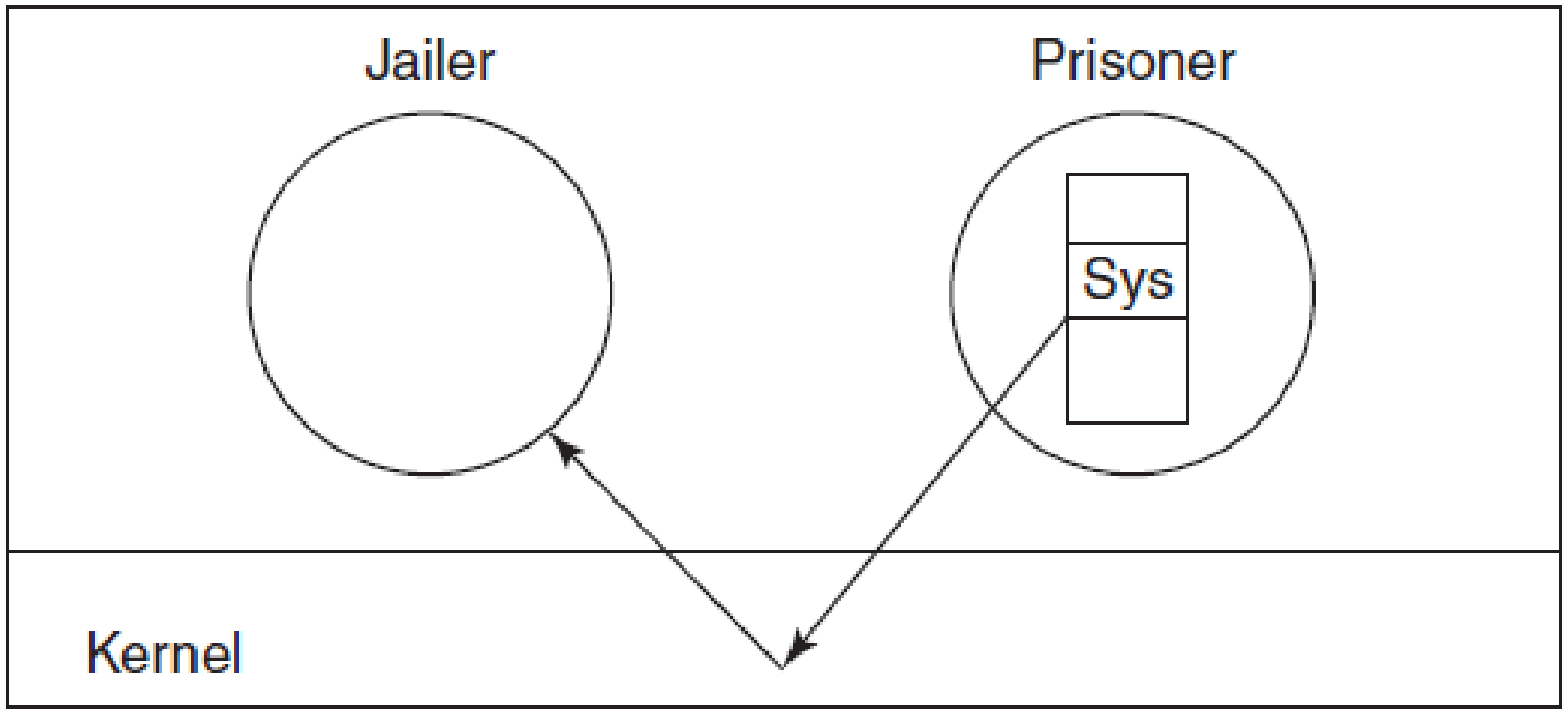


Figure 9-36. The operation of a jail.

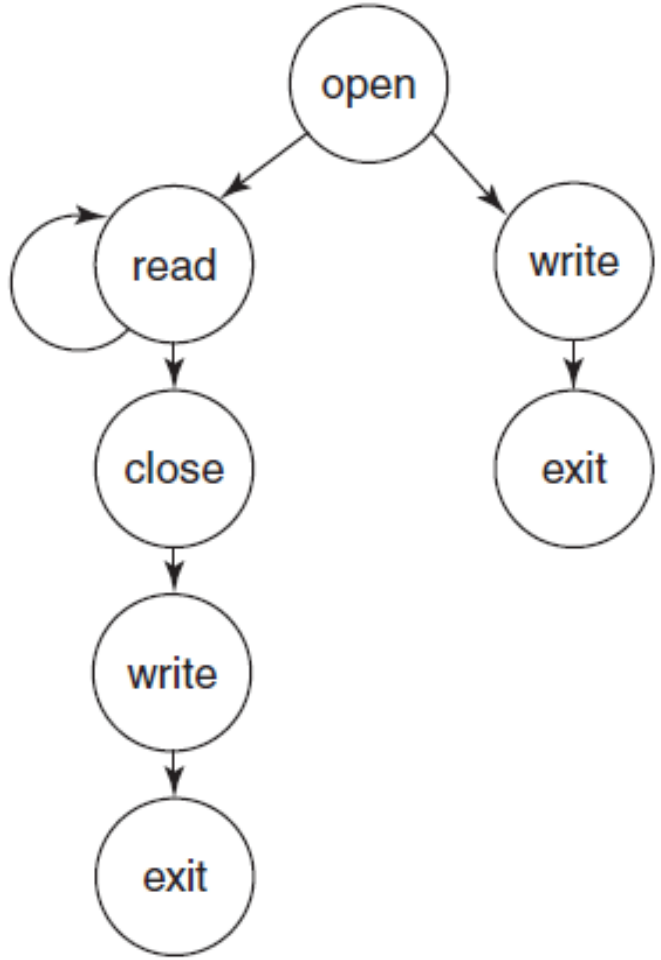
Static Model-Based Intrusion Detection

- The compiler of a program generates its system-call graph.
- The author of the code signs the graph.
- The jailer checks the actual system call sequence against paths in the graph.
- Virus injection may form system call sequences that do not match the graph and thus be detected.
- Mimicry attack is one that uses only system call sequences matching a signed graph.

```
int main(int argc *char argv[])
{
  int fd, n = 0;
  char buf[1];

  fd = open("data", 0);
  if (fd < 0) {
      printf("Bad data file\n");
      exit(1);
  } else {
      while (1) {
          read(fd, buf, 1);
          if (buf[0] == 0) {
            close(fd);
            printf("n = %d\n", n);
            exit(0);
          }
          n = n + 1;
      }
  }
}
```

(a)



(b)

Figure 9-37. (a) A program. (b) System-call graph for (a).

Mobile Code

- Applets, agents, PostScript files are examples of mobile code that are foreign programs running on your machine (with your permission).
- Sandboxes can be assigned to applets (one for data and one for code to prevent self-modifying code).
- A reference monitor handles all system calls from the mobile code.

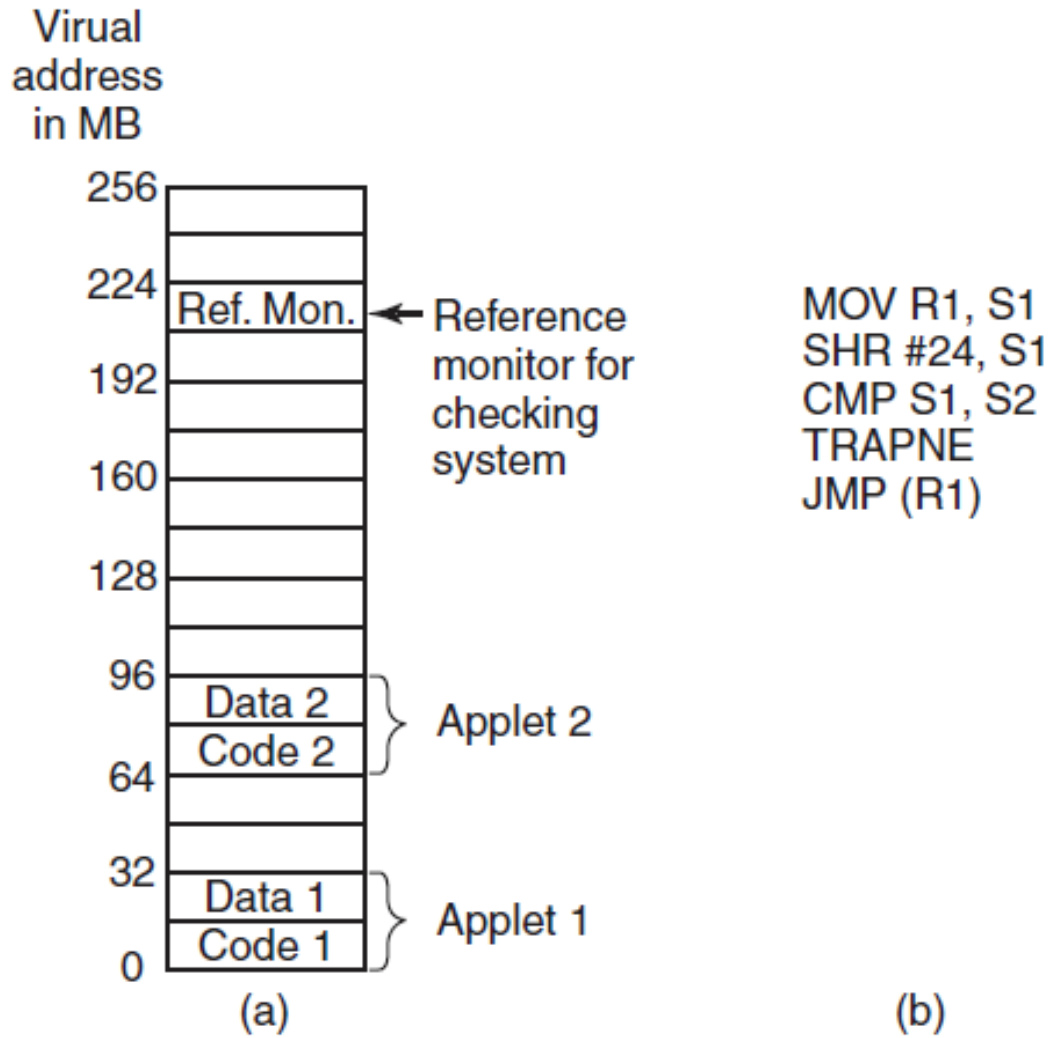


Figure 9-38. (a) Memory divided into 16-MB sandboxes. (b) One way of checking an instruction for validity.

Interpretation of Mobile Code

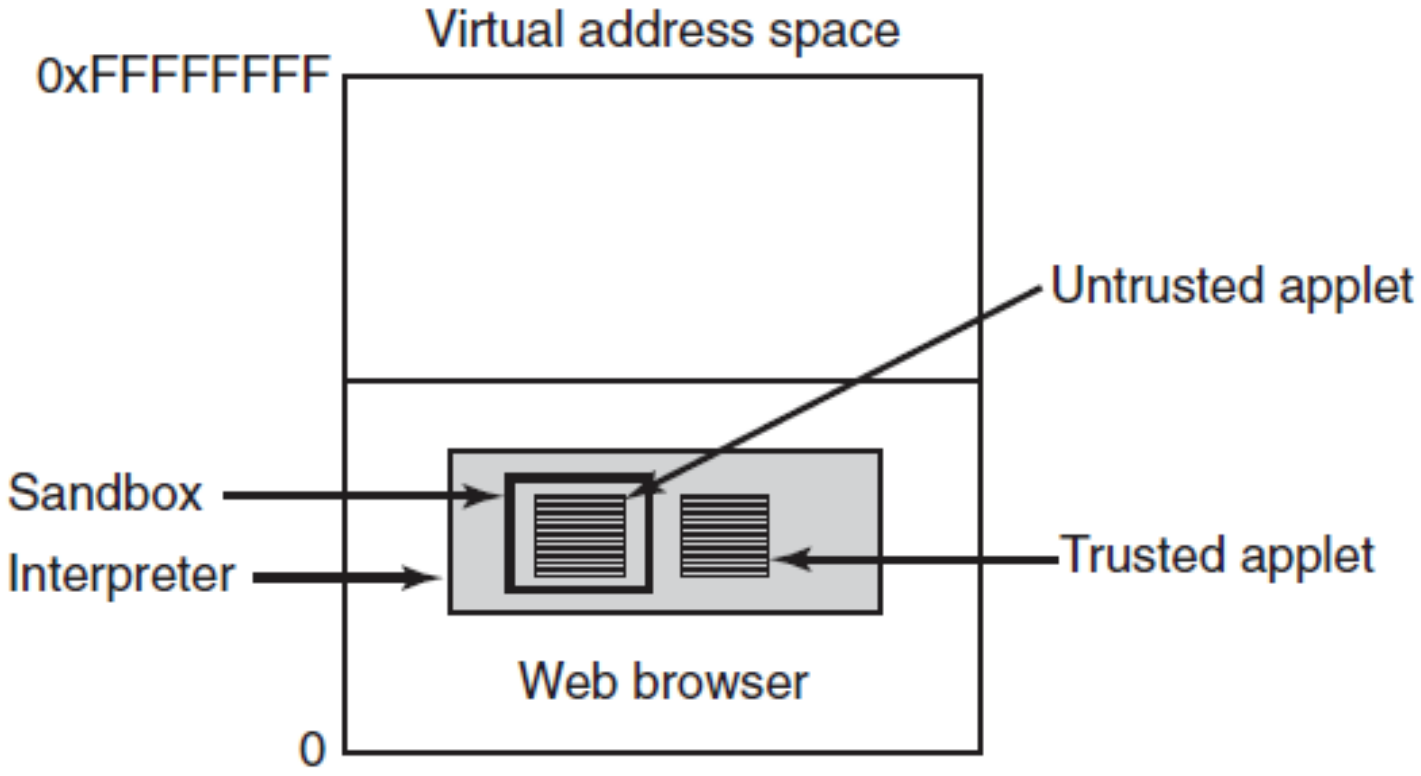


Figure 9-39. Applets can be interpreted by a Web browser.

Java Virtual Machine and Runtime Environment

1. Does the applet attempt to forge pointers?
2. Does it violate access restrictions on private-class members?
3. Does it try to use a variable of one type as another type?
4. Does it generate stack overflows or underflows?
5. Does it illegally convert variables of one type to another?

I/O Protection in Java

URL	Signer	Object	Action
www.taxprep.com	TaxPrep	/usr/susan/1040.xls	Read
*		/usr/tmp/*	Read, Write
www.microsoft.com	Microsoft	/usr/susan/Office/–	Read, Write, Delete

Figure 9-40. Some examples of protection that can be specified with JDK 1.2.