Disposition 5: System Security Mechanisms

Mathias Ravn Tversted

December 18, 2019

Table of contents

Trusted computing base and firewalls

Malware and defenses

Security Policies

Trusted Computing Base

A *Trusted Computing Base* is a part or parts of a system that is trusted to perform according to specification. This can be established using hardware, or using software security depending on the threat model. It can also be in several places. An example of this is SGX. A system on chip that is protected is called a *secure enclave*. An enclave can be used to handle keys and or sign things. It can also run code. It will often be a part of the OS.

Firewalls

Even if a computer insists on authentication, it still has to perform untrusted communication momentarily. This opens it up to attack. Generally we have the following roles:

- **Packet filters**: Sits on network, filters communication, blocks packages to machine et cetera. TCP packages have flags that can be recognised etc.
- Proxies:
- Stateful Firewalls

Packet filtering

Packet filters can look at packages can determine if they should be allowed through. There is a tradeoff between being too nazi, and being *too* permissive. The packet filter needs to be smart and do thorough analysis in order to be effective

Proxies

A proxy firewall does not allow clients to make connections, but rather it handles connections *on behalf of the machines*. This hides the local network from the outside. Downside is that programs need to be configured to use proxies. Proxy firewalls are regarded as being quite secure.

Stateful firewalls

A *stateful firewall* keeps track of all the current connections that goes throuh it. It checks to see if the traffic going through it is allowed or not. Flexibility is needed for UDP packets. A stateful firewall can also translate local addresses to something else, this is known as a *masquerading firewall*, can help protect against IP address scanning. It can also scan for suspicious behavior, and block that traffic altogether.

Malware

Malware comes in different flavours, such as

- **Trojan Horses**: Appear to be useful, but have hidden intentions. Such as being spyware or destroying or stealing data
- **Viruses**: Infect programs and spread themselves to other machines
- **Worms**: Similiar to viruses, but are stand-alone programs
- **Ransomware**: Encrypts your files or similiar, and demands money, usually in cryptocurrencies. May initially act as trojans and may want to spread themselves (see WANNACRY)

Virus Scanners

Virus scanners scan your machine for malicious software. Many users do not update their scanners and attacks happen due to old scanners, software/hardware and not necessarily because of new attacks. (See WANNACRY infecting old computers). Malware may try to encrypt itself, or have barebones infrastructure that loads the actual malicious code into memory.

Intrusion Detection

Intrusion Detection is the practice of detecting malicious behaviour rather than particular pieces of malicious software. This happens by monitoring processes, and or users. An example of this, is a stateful firewall. There are two approaches: *Rule-based* and *statistical* intrusion detection. **Rule-based**: System sets up rules for normal behaviour **Statistical**: First gather statistics on normal behaviour, detect outliers. Intrusion detection is also a tradeoff between being eager and being permissive. A system may use *honey pots*.

Security Policies

We usually organise the way we think of security in the following ways

- **Security Policy**: Specification of the system in question. Description of security objects, and maybe high-level strategy for achieving the objectives
- **Threat Model**: Descriptions of the attacks we want to protect ourselves against
- **Security mechanisms**: The technical solutions we use to reach our objectives

Models for Security Policies

- **Lattice approach**