For every computer system and software design, it is imperative that it should address all security concerns and implement required safeguards to enforce security policies. At the same time, it is important to keep a balance since rigorous security measures can not only increase costs but also limit the user-friendliness, usefulness and smooth performance of the system. Hence, system designers have to ensure effective performance without compromising on security. A computer’s operating system must concentrate on delivering a functionally complete and flexible set of security mechanism for security policies to be effectively enforced.

An operating system’s protection and security requires all computer resources such as software, CPU, memory and others to be protected. This can be enforced by ensuring the confidentiality, integrity and availability in the operating system. It must be able to protect against all threats including malware and unauthorized access.

## Threats to Operating Systems

Let’s have a look at the common threats faced by any operating system.

Anything that has a malicious nature and can be harmful for the system is a threat.

### Malware

This category includes viruses, worms, trojan horses and all kinds of malicious software. These are generally small code snippets that can corrupt files, destroy data, replicate to spread further, and even crash a system. Many times, the malware goes unnoticed by the victim user, while the cyber criminals silently extract sensitive information.

### Denial of Service Attacks

DoS attacks don’t actually attempt to damage a system, but rather clog it to make it useless. A tight loop that requests system services repeatedly is an example of a DoS attack.

### Network Intrusion

Network intruders can be classified as masqueraders, misfeasors or a clandestine users. A masquerader is an unauthorized individual who penetrates into a system and exploits an authorized individual’s account. Misfeasor is a legitimate user who accesses and misuses programs, data or resources. Clandestine user takes over supervisory control and tries to evade access controls and audit collection.

**Buffer Overflow**

Also called buffer overrun, buffer overflow is defined in the NIST Glossary of Key information security terms as “A condition at an interface under which more input can be placed into a buffer or data-holding area than the capacity allocated, overwriting other information. Attackers exploit such a condition to crash a system or to insert specially crafted code that allows them to gain control of the system”

Buffer overflow is one of the most common and dangerous security threats. To exploit a buffer overflow, attackers identify a buffer overflow vulnerability in a program and understand how the buffer will store in process memory to finally alter the execution flow of the program.

## Ensuring Operating Systems Security

Operating systems security can be ensured with the following mechanisms.

### Authentication

Authentication identifies every user in a system and ensures that their identity is legitimate. The operating system makes sure that each user is authenticated before they are allowed to access a system. Different ways to ensure their authenticity are:

**Username and Password**

Every user has their distinct username and password that needs to be entered correctly before they are able to access a system

**User Attribution Identification**

These methods usually involve biometrics verification such as finger prints, eye retina scan, etc. This authentication is based on uniqueness of users and is compared with the database samples that already exist in the system. Users can access only in case of a match.

**One-Time Password**

A one-time password is generated exclusively for each time a user wants to log in and enter a system. The same password cannot be used again. Methods include:

* **Random Numbers**

The system may ask you for numbers corresponding to a set of pre-arranged alphabets. The combination is different every time you require a login

* **Secret Key**

This includes a hardware device that generates a secret key for the user id, and changes every time.

**Tokens**

A user is authenticated with something that they physically possess, such as a [smart card](http://www.cardzgroup.com/SIMCard.html) or electronic keycard.

### Access Control

### Access control specifies who can have access to a system resource and what type of access each entity has. A security administrator maintains an authorization database to specify what type of access is allowed to each user. This database is consulted by the access control function for determining whether access should be granted.

### Intrusion Detection Systems

### Intrusion Detection Systems monitor network traffic or events occurring within a host to identify any suspicious activity. IDS helps identify network, transport and application protocols.

### Firewalls

### Firewalls are important to monitor all incoming and outgoing traffic. It enforces local security, thus defining the traffic that is authorized to pass through it. Firewalls are effective means to protect local systems or network of systems from all network-based security threats.

### Buffer Overflow Defense

### Countermeasures to avoid buffer overflow include compile-time defense, that aims to harden a program for resisting an attack to [enhance software security](https://codesigningstore.com/tips-for-software-developers-to-enhance-software-security); or runtime defense, that detects and aborts attacks in an executing program.

## Key Takeaway

### Operating systems security plays a primitive role in protecting memory, files, user authentication and data access protection. Consistent protection means that the system meets standard security requirements and have the required functionality to enforce security practices.