Server-side Security

How to secure a server?

- Server security is a major issue for companies

- Indeed, being a central element in the functioning of all the components of an information system (applications, network, infrastructure, employees, etc.), servers are often the prime targets of attacks

- Furthermore, server-side vulnerabilities can have severe consequences

How to secure a server?

- In the event of misconfiguration or lack of control,
  - these flaws can be exploited and lead to the compromise of the data in transit, or
  - even to the server being taken over by malicious persons.

Why is server security important?

- Attacks on servers are a daily occurrence because, very often, too many loopholes exist

- Indeed,
  - applications vulnerable to SQL injections hosted on a server,
  - users unaware of social engineering risks, or
  - simply poor practices in terms of updates and patch management of OS and server services,

Why is server security important?

- easily allow attackers to achieve their goals:
  - data theft,
  - access to sensitive information,
  - paralysis of a company's activity, etc

- Securing a server is therefore vital and necessarily involves implementing best practices in terms of
  - configuration,
  - control,
  - monitoring and
  - security testing

Implement an update and patch management policy for servers

- Implementing an update management policy for operating systems and services is essential to maintain a good level of security

- Indeed, new vulnerabilities are discovered and published regularly

- And if security patches are not applied in time, the risk of attacks and server compromise increases.

Implement an update and patch management policy for servers

- The numerous attacks suffered by companies, via malware

- It is therefore important to define and implement an update and patch management policy or servers and all software and hardware components of the IS

- This involves documenting procedures and continuous monitoring of the various patch releases or new versions

Disable or remove unnecessary services

- All software and components installed on an OS increase the attack surface and therefore the risk of compromise

- However, strengthening server security requires reducing the attack surface

- To do this, it is necessary to disable or even remove (as far as possible) all services, applications, network protocols, third-party components, etc. that are not essential to the operation of your server

Disable or remove unnecessary services

- Removing or disabling unnecessary systems enhances the security of a server in several ways

- Firstly, they cannot be compromised, nor can they be used as attack vectors to alter the services that are essential for the server to operate

- Indeed, it should be kept in mind that each component added to a server increases the risk of compromising it

Disable or remove unnecessary services

- Furthermore, the server can be configured to better meet the requirements of a particular service, while improving performance and the overall level of security

- Finally, reducing services means limiting the number of log entries, which makes it easier to monitor and detect unusual events

Controlling and securing server access: Focus on SSH

- Presentation of SSH: an essential tool to ensure proper management and secure administration of the server

- SSH (Secure Shell) is both a communication protocol and a computer program allowing local and remote administration of a server

- Its most common implementation is OpenSSH, which can be found on many systems, including servers (Unix, Linux, Windows) as well as workstations and network equipment

Controlling and securing server access: Focus on SSH

- Indeed, OpenSSH is a suite of tools offering many features, including:
  - a server (sshd),
  - several clients – remote shell connection (ssh) / file transfer and download (scp and sftp),
  - a key generation tool (ssh-keygen),
  - a keychain service (ssh-agent and ssh-add), etc

Controlling and securing server access: Focus on SSH

- SSH currently exists in two versions: SSHv1 and SSHv2
- SSHv1 contains vulnerabilities that have been fixed in the second version
- Therefore, for enhanced security, only version 2 of the SSH protocol should be authorized
- The most widely used feature of SSH is remote administration, which consists of connecting to a remote machine and launching a shell session following authentication

Public key / certificate authentication

- Not ensuring the authenticity of a server can have several security impacts, including
  - the inability to verify that you are communicating with the correct server,
  - with consequent risks of spoofing and exposure to Man in The Middle attacks.

- SSH relies on asymmetric encryption for authentication, and thus ensures the legitimacy of the server being contacted before access is granted

Public key / certificate authentication

- Moreover, this control is done in several ways with OpenSSH

- Either by
  - ensuring that the public key fingerprint obtained previously with ssh-keygen and presented by the server is the correct one, or
  - by verifying the signature of the certificate presented by the server with a certification authority known to the client

Security of authentication and user access control

- Users of a system always have rights, no matter how small

- It is therefore important to protect their access via a secure authentication mechanism and to thwart brute force attacks as much as possible

- Firstly, each user must have his or her own account to ensure better traceability and the allocation of access rights must always follow the principle of least privilege

Security of authentication and user access control

- Furthermore, access to a service should be restricted to users who have a justified need and are explicitly authorized

- Regarding users authorized to configure the operating system of servers, they should be limited to a small number of designated administrators

Security of authentication and user access control

- Furthermore, it is strongly discouraged to use authentication mechanisms in which authentication information is transmitted in the clear over a network, as this information can be intercepted via Man in the Middle attacks and used by an attacker to impersonate an authorized user

Security of authentication and user access control

- Finally, to ensure secure user authentication, the following measures should be implemented:
- **Removing default accounts**
  - because the default configuration of the OS often includes guest accounts, administrator accounts and accounts associated with services.
  - The names and passwords of these accounts are known to attackers.

Security of authentication and user access control

- **Implement a proper password policy**
  - All passwords should be complex and difficult to guess.
  - Above all, they should be long (more than 15 characters).
  - To achieve this, nothing is better than the implementation of a password manager

Logging and monitoring of server events

- Logging is a central aspect of a security strategy

- It is a control mechanism for
  - monitoring the network, systems and servers; and
  - it ensures the traceability of all normal and suspicious events.

- Indeed, log files can be used to track the activities of an attacker and thus detect unusual events or failed or successful intrusion attempts.

Logging and monitoring of server events

- To facilitate the management and exploitation of logs, they should be centralized on a dedicated server

- This is even more important because, in the event of a machine being compromised, it is likely that the logs will be destroyed or altered by the attacker

- Centralizing, regularly backing up and duplicating logs will ensure that a copy is always kept. And given their importance, it is essential to restrict access to logs to authorized users only

Securing APIs, websites and web applications hosted on a server

- APIs, websites and web applications hosted on a server must also be secured

- These can be used as attack vectors to compromise the server if they are vulnerable to SQL injections for example