Spoofing and DoS Attack on WordPress Server

Summary of the Spoofing and DoS Attack

We simulated a **Spoofing** and **Denial of Service (DoS)** attack on a WordPress server. The goal was to overload the server with fake requests and observe its behavior.

- **Spoofing**: In this type of attack, the attacker hides their true IP address by pretending to be someone else. This makes it harder to track and block malicious traffic.
- **DoS**: The attack flooded the WordPress server with requests, causing it to slow down or become unavailable to real users.

Monitoring the Attack

We used tools like `htop` in **Kali Linux** to observe how the server responded:

- **CPU Usage**: The CPU on the server spiked, with core 0 reaching over 80% usage.
- **Memory and Network**: Network traffic and memory usage increased as the attack progressed, due to the server handling more requests than usual.
- **Server Response**: The WordPress site became slower and eventually stopped responding, leading to error messages and timeouts for real users.

Defending Against Spoofing and DoS Attacks

To protect a WordPress server from these kinds of attacks, we can use several strategies:

1. **Firewalls**:
   - A **firewall** can block suspicious traffic or IP addresses that send too many requests.
   - **Rate limiting** helps to stop a single IP address from overwhelming the server.
2. **IP Filtering and Blocking Spoofed Traffic**:
   - Use firewall rules to detect and block traffic with spoofed IP addresses.
3. **Load Balancing**:
   - **Load balancers** help spread traffic across multiple servers so that no single server gets overwhelmed.
   - This can also prevent the server from crashing during an attack.
4. **Monitoring Tools**:
   - Tools like `htop` can help detect unusual traffic spikes and server load early, so you can act before the server goes down.

Best Practices for WordPress Security

To avoid these types of attacks in the future, it’s important to follow some security best practices:

- **Update Regularly**: Keep WordPress, plugins, and the server software up to date to avoid known vulnerabilities.
- **Use Strong Authentication**: Implement strong passwords and two-factor authentication (2FA) to prevent unauthorized access.
- **Regular Backups**: In case of an attack, having a recent backup will help you recover quickly.