

# Pen Testing

---

ESSAYER DE TROUVER ET D'EXPLOITER LES VULNÉRABILITÉS D'UN  
SYSTÈME INFORMATIQUE

# Structure

---

- Introduction
- DoS / DDoS
- SQL injection
- MITM (man-in-the-middle)
- Isolement pour les tests
- Réflexion
- Conclusion

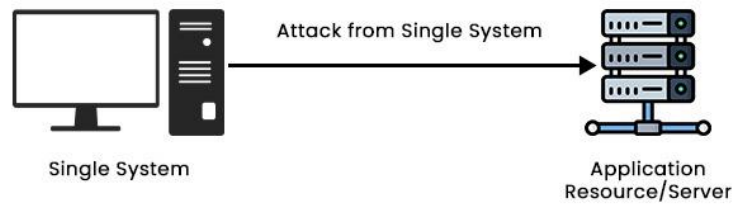
# Introduction

---

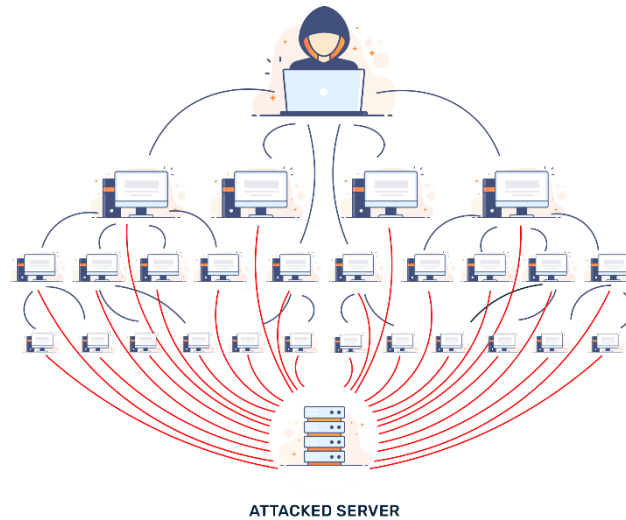
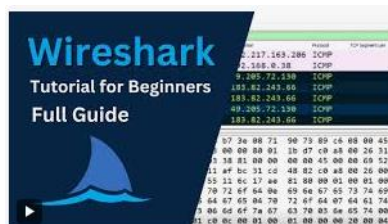
L'importance du pentesting réside dans la capacité à anticiper d'éventuelles failles de sécurité avant qu'elles ne soient exploitées par des attaquants malveillants, ce qui peut entraîner des pertes financières significatives pour une entreprise et même la mener à la faillite.

# DoS/DDoS

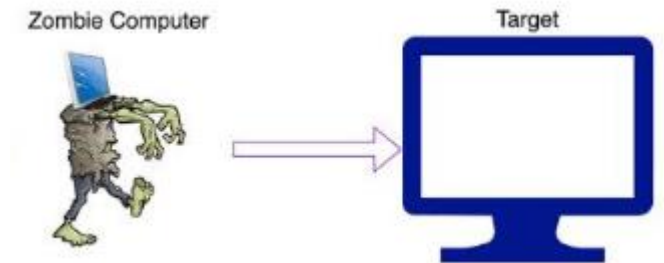
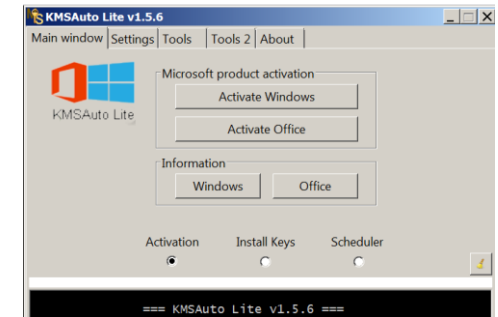
## DoS Attack



Facile à détecter

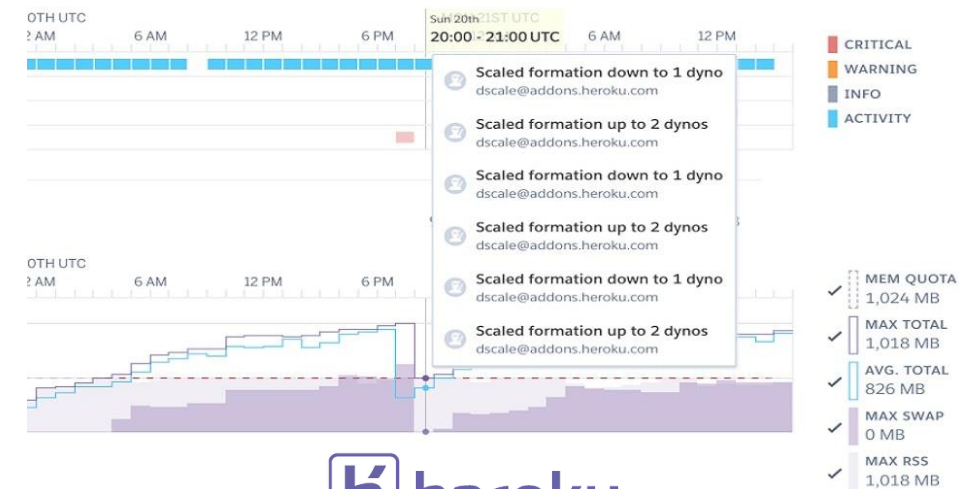


très difficile à détecter



# Test

de nombreux outils



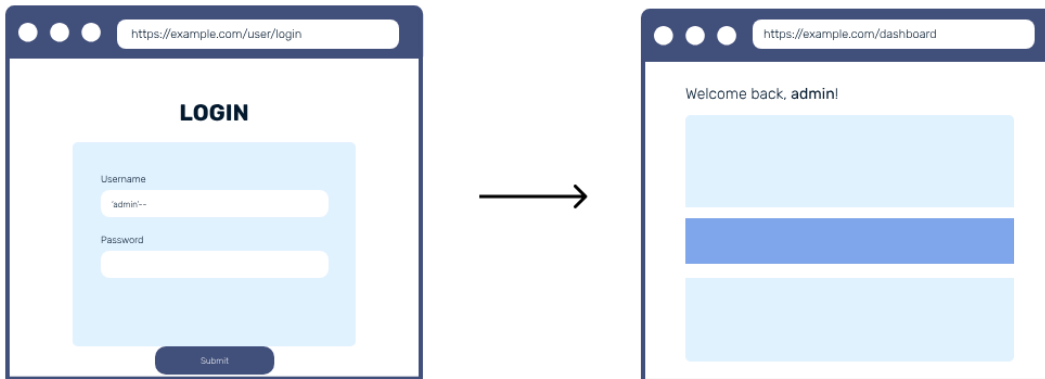
# SQL injection



## Running an SQL Injection Attack – Computerphile

<https://www.youtube.com/watch?v=ciNHn38EyRc>

query= "SELECT \* FROM Users WHERE Username = " + username + " AND Password = " + password + ";"



Create new app

Name of my app:

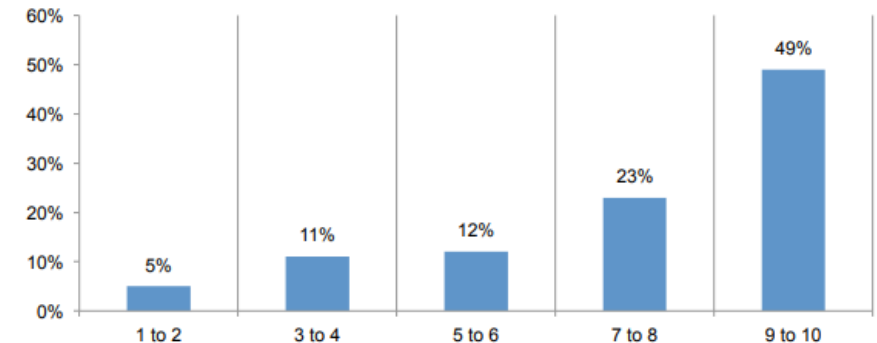
<App Name>

Some characters are not allowed, for example: \ / : \* ? " ' < > |

Cancel Create

## Is SQL injection still a thing?

Figure 2. The SQL injection threat facing your company today  
1 = no threat to 10 = significant threat



## The SQL Injection Threat Study

<https://www.ponemon.org/local/upload/file/DB%20Networks%20Research%20Report%20FINAL5.pdf>

# Tests

## Tests manuels



```
sec@kali:~/sqlmap$ python sqlmap.py --update
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 21:34:51

[21:34:51] [INFO] updating sqlmap to the latest development version from the GitHub repository
[21:34:51] [INFO] update in progress ....
[21:34:55] [INFO] updated to the latest revision '74d0315'

[*] shutting down at 21:34:55

sec@kali:~/sqlmap$ python sqlmap.py
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 21:34:56

[21:34:56] [INFO] updating sqlmap to the latest development version from the GitHub repository
[21:34:56] [INFO] update in progress ....
[21:34:59] [INFO] updated to the latest revision '74d0315'

[*] shutting down at 21:34:59

Usage: python sqlmap.py [options]
```

```
sqlmap -u
```

```
"http://localhost/vulnerable_page.php?id=1" -D
database_name -T table_name --dump --batch
```

# MITM (man-in-the-middle)

---



## Indicateurs d'une attaque MITM :

Connexions non chiffrées :

Trafic HTTP au lieu de HTTPS, en particulier sur des réseaux publics ou inconnus.

Ponts intermédiaires :

Le trafic semble passer par des intermédiaires inattendus (comme un proxy ou une passerelle inconnue).

Changements dans les adresses ARP :

La table ARP montre que plusieurs IP sont associées à la même adresse MAC (indication de spoofing).

Latence anormale :

Des retards inattendus dans le trafic peuvent être un symptôme de redirection ou d'inspection.



# Tests

---

## **Validation des certificats**

Inspecter le certificat délivré par un serveur HTTPS pour vérifier sa validité et sa correspondance avec le domaine.

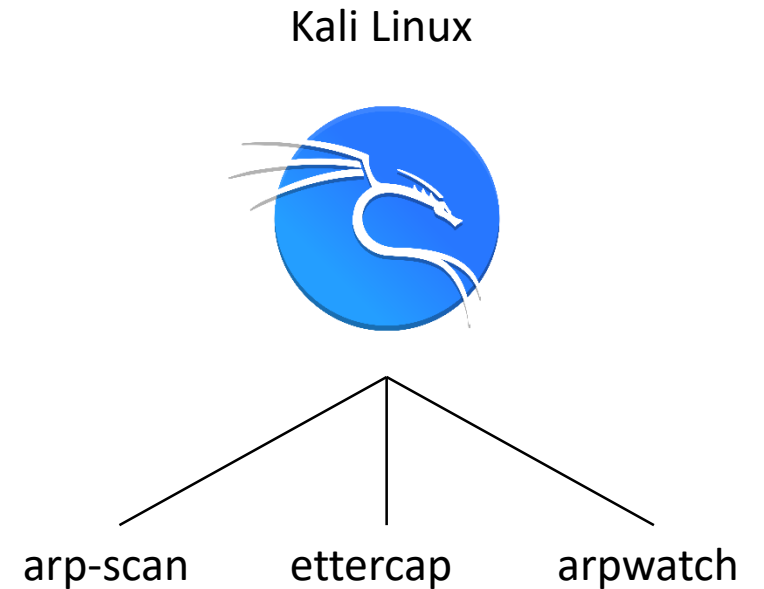
## **Analyse du réseau**

### **Détection du spoofing ARP :**

Détecter les changements dans la table ARP.

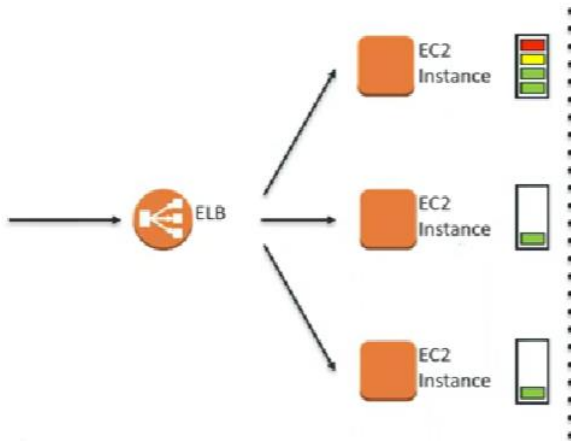
### **Tests de proxy**

Analyser si un proxy intermédiaire altère le trafic :  
Faux proxy pour analyser si les applications autorisent des connexions non sécurisées.



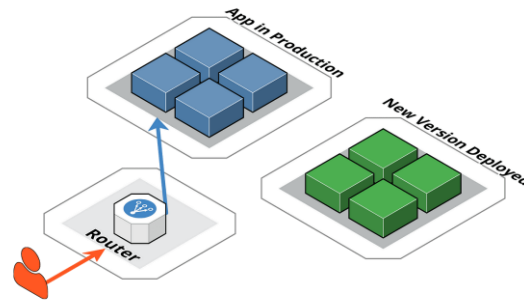
# Isolement pour les tests

Load Balancers with Isolated Traffic  
AWS Elastic Load Balancer

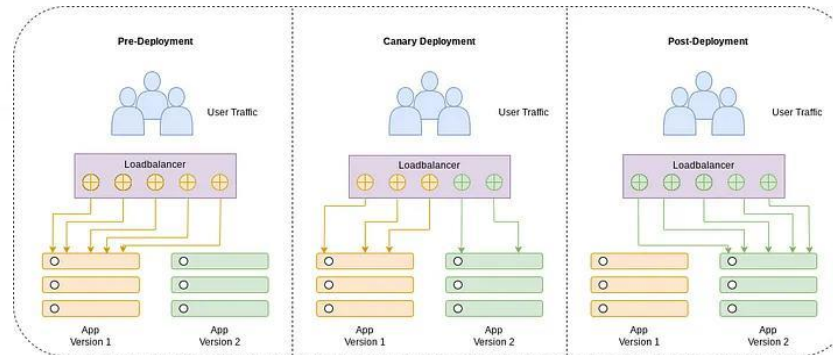


Health checks ensure that request traffic is shifted away from a failed instance.

Blue/Green Deployments



Canary Deployments



Balancement

Round-Robin DNS

Migration site web

```
server {  
    listen 80;  
    server_name oldsite.com;  
    return 301  
    http://newsite.com$request_uri;  
}  
  
location /old-path/ {  
    return 301 /new-path/;  
}
```

# Réflexion

---

Malgré toutes ces précautions, les attaques sont inévitables. Il est donc important de réfléchir à la manière dont nous allons réagir lorsque cela se produira.

# Conclusion

---

Les vulnérabilités seront toujours présentes, il est donc important de réaliser des tests pour les détecter avant que les attaquants ne le fassent et, dans le pire des cas, d'établir une stratégie sur la manière de réagir face à une attaque.