HERVÉ SCHAUER CONSULTANTS
Cabinet de Consultants en Sécurité Informatique depuis 1989
Spécialisé sur Unix, Windows, TCP/IP et Internet

JSSI 2010

Les webshells,

ou comment ouvrir les portes de son réseau ?

Renaud Dubourguais
<renaud.dubourguais@hsc.fr>

Retours d'expérience

Retours d'expérience HSC

- **Problème essentiel ses dernières années: applicatif**
  - Sécurité périmétrique maîtrisée
  - 90% des tests intrusifs : applicatif
  - ≃ 100% des cas : présence de vulnérabilités exploitables
- **Pourquoi ?**
  - Domaine en forte évolution (Web 2.0, Web Services ...)
  - Trop peu de sensibilisation des développeurs à la sécurité
  - Traitement des aspects sécurité trop tardif
  - Manque de temps et de budget

⇒ Mise en production avec des vulnérabilités exploitables.

Impacts de ces vulnérabilités

- **Classiques: atteinte à l'image de la cible**
  - Défiguration du site web
  - Récupération et diffusion d'informations client

⇒ Exploitations itératives des vulnérabilités

- **Plus vicieux: serveur web = serveur rebond**
  - Prise de contrôle du serveur web
  - Rebond sur des équipements internes ou tierces

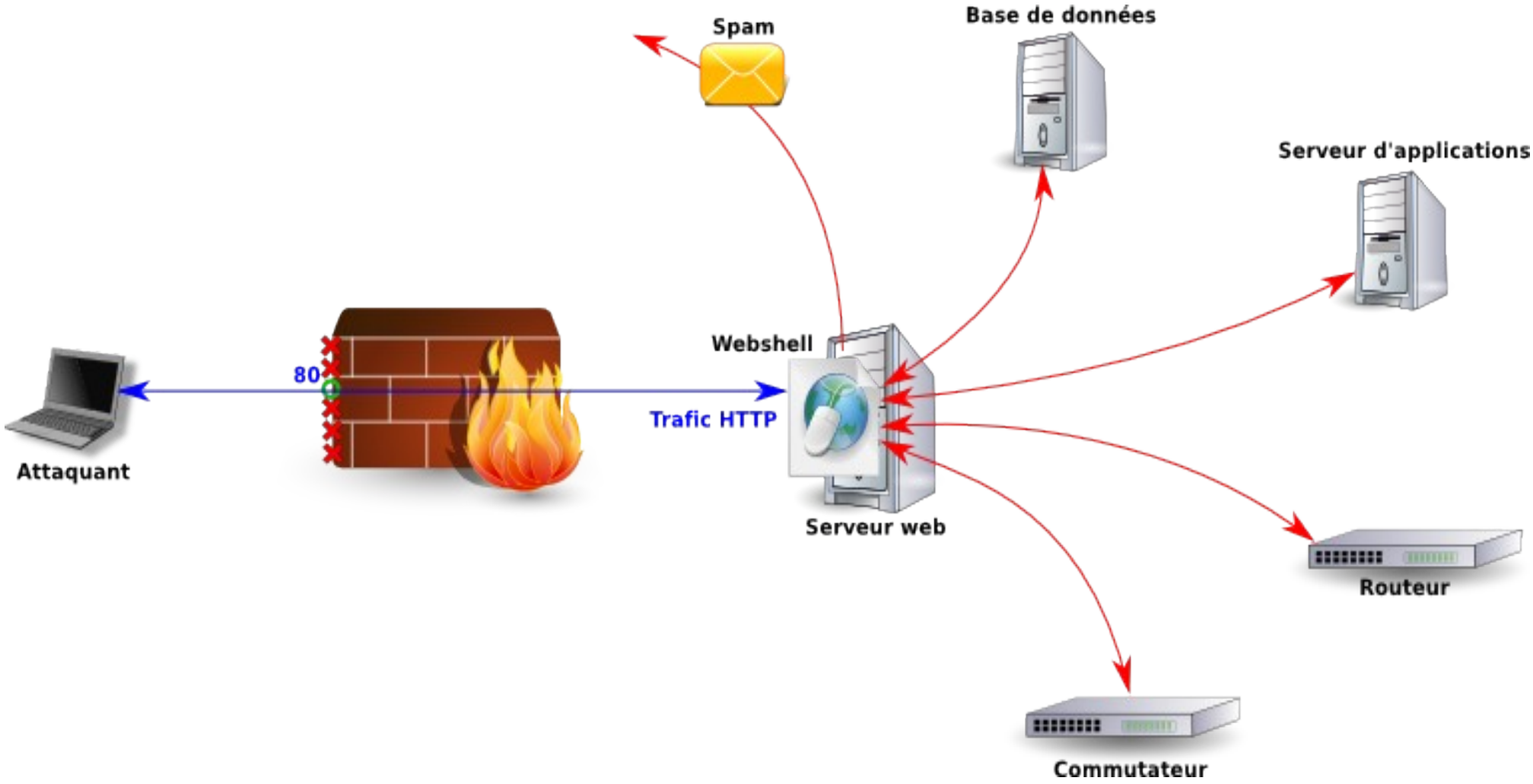
⇒ Déploiement de nouvelles applications: webshells

Webshell ?

Webshell ?

- **Porte dérobée au sein d'un(e) application/serveur web:**
  - Accessible via une URL particulière à connaître
  - Exécutant des actions sous l'identité du serveur web (parfois *root* !)
  - Ouvrant une porte sur le réseau interne via HTTP(S)
  - Permettant l'attaque par rebond d'un site tierce
- **Quelques exemples de fonctionnalités:**
  - Exécution de commandes systèmes sur le serveur web
  - Interrogation de bases de données
  - *Spam relay*
  - Encapsulation de TCP dans HTTP

Un webshell en image ...



Déploiement d'un webshell

Déploiement d'un webshell

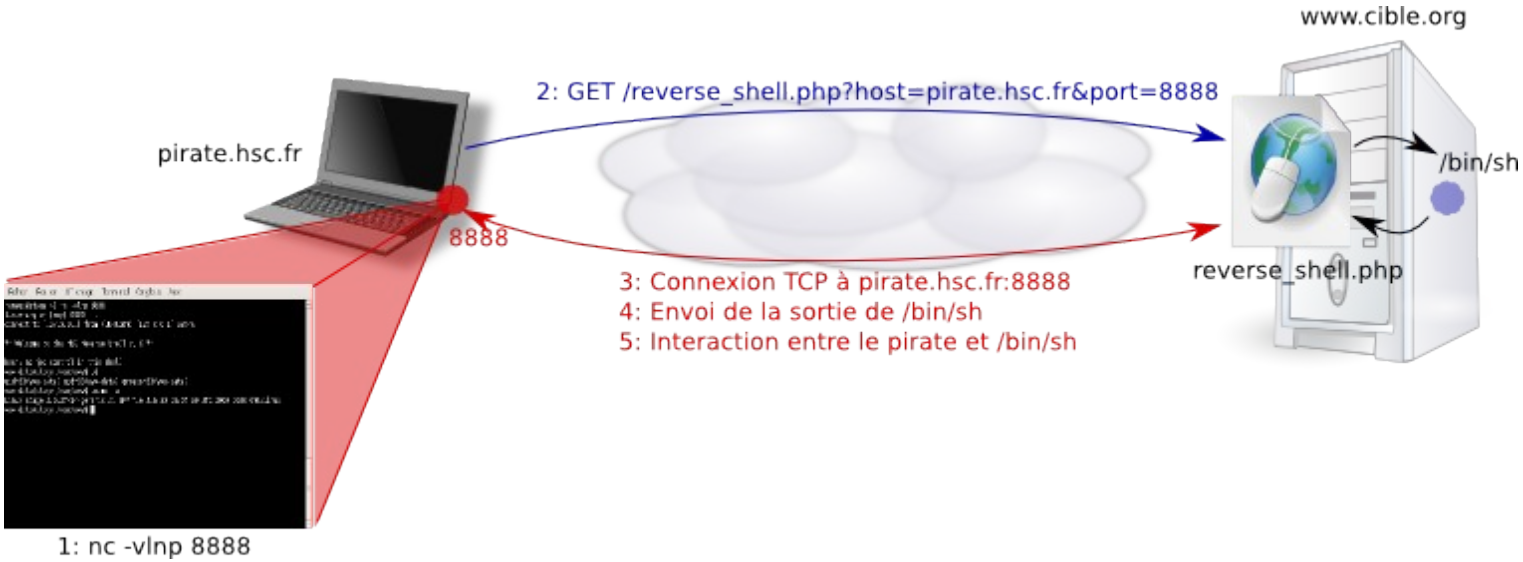
- **Pas du tout marginal !**
- **Parfois complexe:**
  - Vulnérabilités dans les socles applicatifs
  - Mauvais filtrages des entrées utilisateurs (injection SQL …)
  - Exposition de fonctionnalités à risque (JMXInvokerServlet sur JBoss)
- **Souvent simple:**
  - Mauvais filtrage des extensions de fichiers pour les *uploads*
  - Compromission de l'interface d'administration

DEMONSTRATION

Prise de contrôle du serveur web

Prise de contrôle du serveur web (1/3)

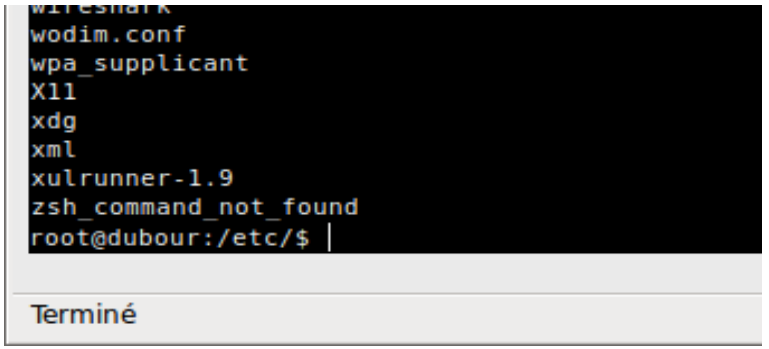
- **Objectif: interagir avec l'OS et le système de fichiers**
- **Solution de facilité :**
  - Connexion TCP entre une invite de commande et le pirate
  - Exploiter la faiblesse du filtrage en sortie : ***Reverse Shell***



DEMONSTRATION

Prise de contrôle du serveur web (2/3)

- **Et si le filtrage est trop restrictif ?**
  - Ne marche pas toujours !
- **Utiliser le canal HTTP**
- **Utiliser le langage serveur à son avantage :**
  - Version du moteur et des API
  - Librairies chargées
  - …



DEMONSTRATION

⇒ Utilisation du navigateur web comme interface

Prise de contrôle du serveur web (3/3)

- **Mais :**
  - Pas de réelle invite de commandes
  - Pas d'élévation de privilèges possible
- **Finalement, pas de solutions magiques …**
  - Shell Ajax + *polling*
  - Encapsulation HTTP des entrées/sorties d'un *shell*
  - Accès au système de fichiers du serveur web via FUSE
  - …

⇒ Combiner les solutions pour maximiser ses chances

Reconnaissance du réseau interne

Reconnaissance du réseau interne

- **Un serveur web n'est généralement pas seul …**
  - Routeurs
  - Serveurs d'applications
  - Serveurs de bases de données
  - …
- **Un serveur web possède une meilleure vue que l'attaquant**

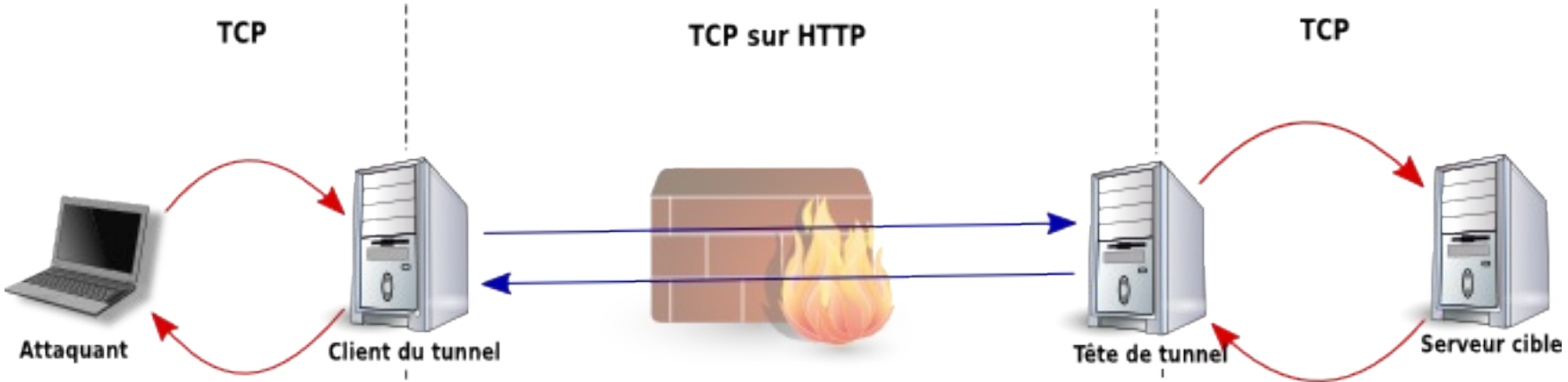
⇒ Intégration d'un scanner TCP au sein du webshell

DEMONSTRATION

Rebond au sein du SI

Rebond au sein du SI (1/2)

- **Atteindre le réseau interne malgré:**
  - Filtrage intermédiaire
  - Diversité des protocoles
- **Utiliser la même porte d'entrée: TCP over HTTP**



DEMONSTRATION

Rebond au sein du SI (2/2)

- **TCP over HTTP: Graal de l'intrusion web**
  - Contournement total du filtrage d'entrée de site
  - Compromission de serveurs internes
  - Interaction directe avec les bases de données internes

Voire même …

- **Prise de contrôle des équipements clés du réseau interne ...**
- **Attaques de sites tierce depuis le serveur web en *Full TCP***

Comment s'en prémunir ?

Les modes de sécurité (1/2)

- **Les plus connus:**
  - PHP: Safe Mode (disparaît avec PHP 6)
  - JAVA: Security Manager
- **Souvent très limitant (*sandboxing*):**
  - Restrictions sur les répertoires
  - Interdictions d'interagir avec l'OS
  - Cloisonnement totale des applications
- **Parfois vulnérables et contournables:**
  - Safe Mode: cf. Stefan Esser

Les modes de sécurité (2/2)

- **Efficaces … si bien configurés !**
- **Mais:**
  - Dé-responsabilisation des développeurs
  - Code applicatif truffé d'injections en tout genre
  - Applications dépendantes des modes de sécurité

⇒ Manque de « portabilité » de la sécurité

- **Modes de sécurité = « rustines au cas où ... »**
- **Idéalement:**
  - application « bien faite » = pas de mode de sécurité.

Quelques pistes ...

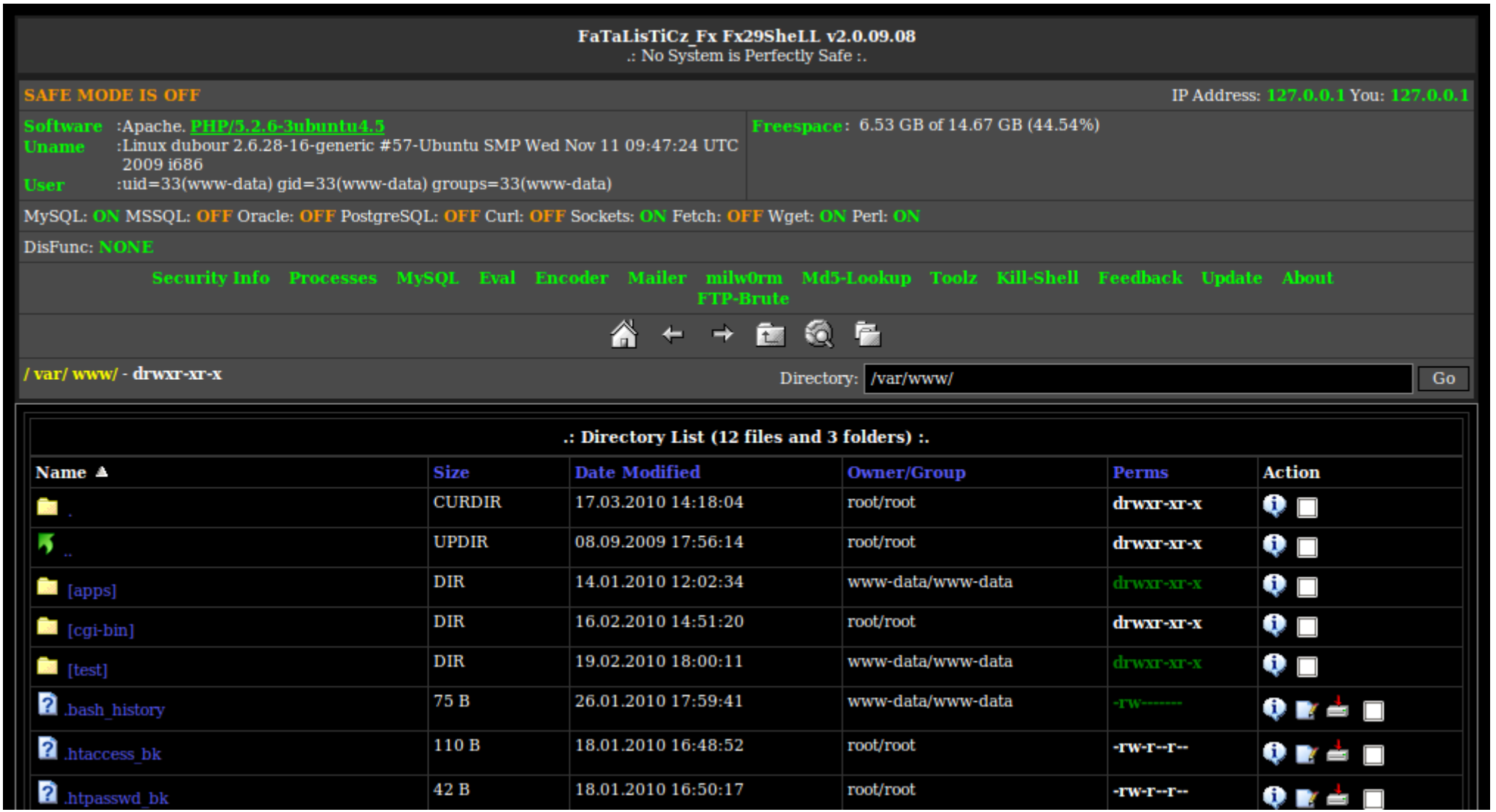
- **Applicatif: faire de la sécurité en amont.**
  - Ne pas s'appuyer **que** sur les modes de sécurité
  - Inclure la sécurité dans le cycle de développement
  - Réaliser des audits de code **pendant** la phase de développement
- **Configurations serveurs:**
  - Ne diffuser que ce que les clients ont besoin de voir
  - Considérer le serveur web comme une réelle porte d'entrée/sortie !
  - Mettre en place un relai inverse ?
  - Mettre en place un WAF ?

Conclusion

Conclusion

- **Les attaques applicatives sont bien réelles:**
  - **Parfois facile à mettre en œuvre**
  - Depuis l'externe ou l'interne
- **Les outils existent et sont de plus en plus évolués:**
  - C99, R57, FX29 …

Conclusion



Questions ?