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

GS-Days 2009

Les Webshells, véritable menace pour les SI ?

Renaud Dubourguais
<renaud.dubourguais@hsc.fr>

Mise en situation

Retours d'expérience HSC

- **Problème essentiel ses dernières années: applicatif.**
  - 90% des tests intrusifs : applicatif
  - ≃ 100% des cas : présence de vulnérabilités exploitables
  - Injections SQL, XSS, CSRF, inclusion de fichiers ...
- **Pourquoi ?**
  - Domaine en forte évolution (Web 2.0, Web Services ...)
  - Encore trop peu de sensibilisation des développeurs à la sécurité
  - Traitement des aspects sécurité trop tardif lors des développements

⇒ Introduction de vulnérabilités exploitables.

Impacts de ces vulnérabilités

- **Attaques simples: atteinte à l'image de la cible**
  - Défiguration
  - Récupération et diffusion d'informations client

  ⇒ Exploitations successives des vulnérabilités

- **Attaques complexes: serveur web = point d'entrée au SI**
  - Prise de contrôle du serveur web
  - Rebond sur le serveur web pour atteindre le réseau interne
  - Exploitations successives des vulnérabilités: pas réaliste

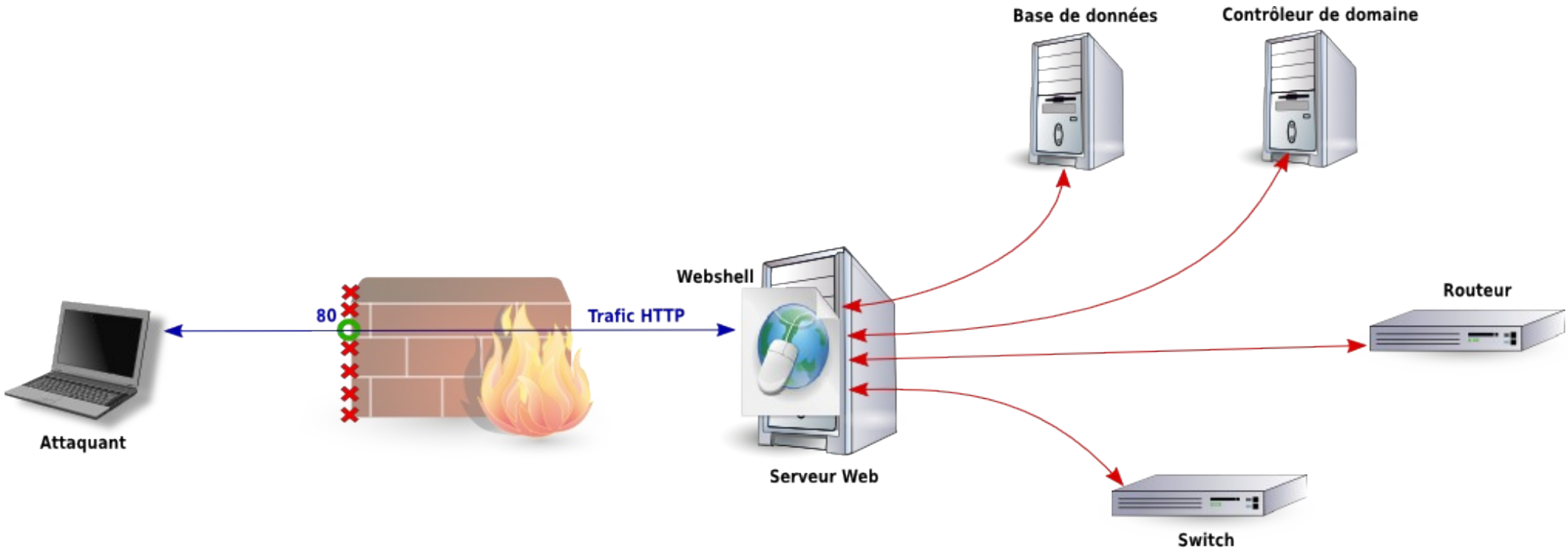
  ⇒ Déploiement de nouvelles applications: Webshells

Webshell ?

Webshell ?

- **Porte dérobée au sein d'un(e) application/serveur web:**
  - Accessible via une URL particulière
  - S'exécutant sous l'identité du serveur web (parfois *root*)
  - Ouvrant une porte sur le réseau interne via le port HTTP(S)
- **Quelques exemples de fonctionnalités:**
  - Téléchargement / Envoi de fichiers sur le serveur web
  - Exécution de commandes systèmes sur le serveur web
  - Interrogation de bases de données internes
  - 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 ...)
  - Mauvais filtrages de ports (JBoss et son port 4444)
- **Souvent simple:**
  - Inclusion de fichiers
  - 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**
- **Utiliser le langage serveur à son avantage:**
  - Exécution de commandes systèmes
  - *Binding* de ports
  - Téléchargement / Envoi de fichiers
- **Solution la plus simple et la plus commune:**

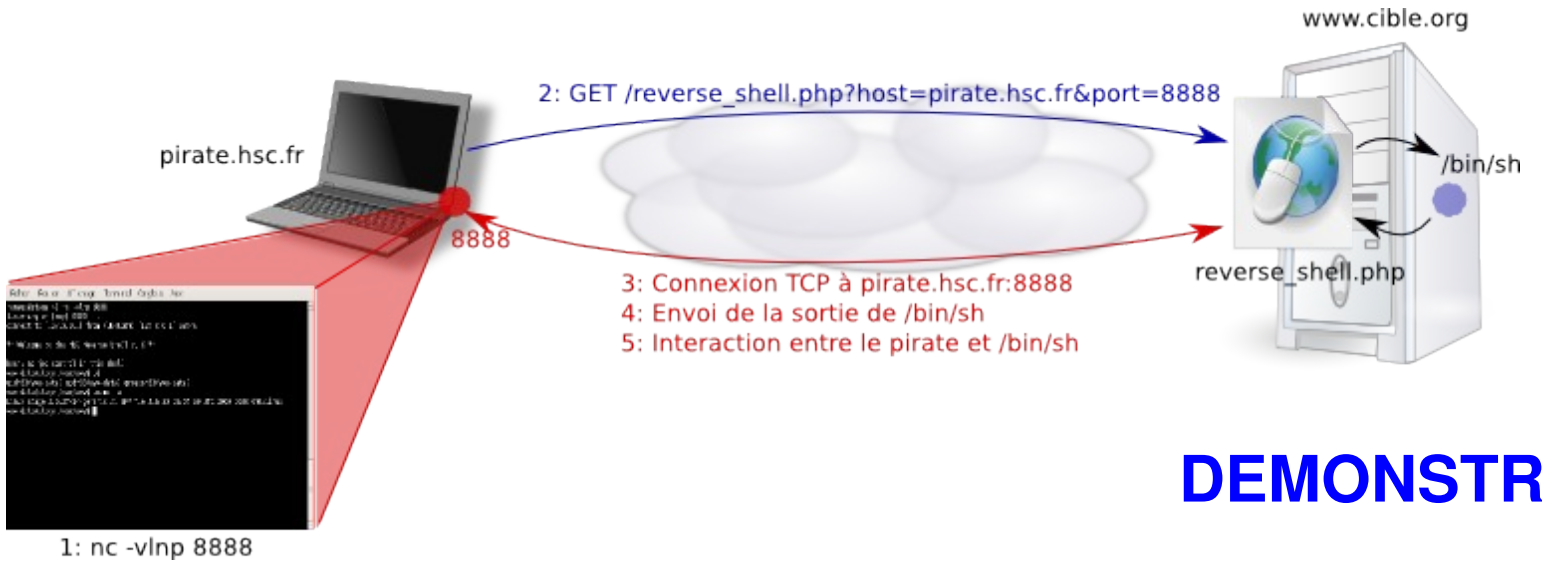
  ⇒ Utilisation du navigateur web comme interface

DEMONSTRATION

Prise de contrôle du serveur web (2/3)

- **Mais:**
  - Manque d'interactivité
  - Pas très agréable à l'utilisation
  - Possibilités limitées

- **Exploiter la faiblesse du filtrage en sortie: *reverse shell***

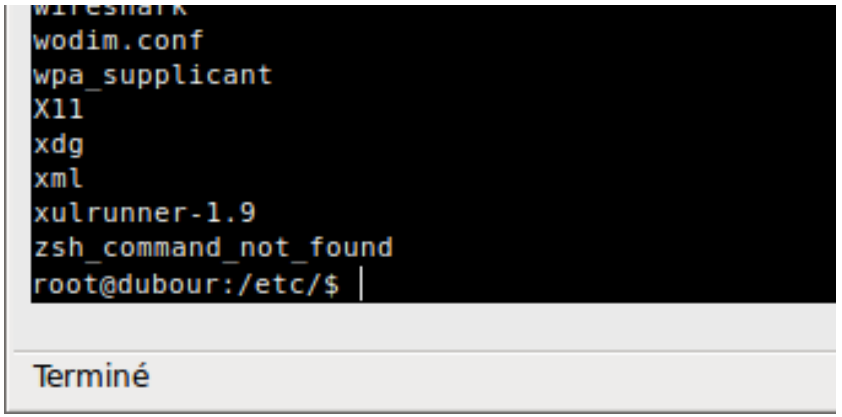


DEMONSTRATION

Prise de contrôle du serveur web (3/3)

- **Et si le filtrage est trop restrictif ?**
  - Ne marche pas dans toutes les situations ...
- **Compromis: mettre le Web 2.0 au profit de l'attaquant**
  - Simulation d'un *shell* dans le navigateur web: interactivité
  - Utilisation de HTTP de bout en bout: pas de problème de filtrage
  - Réduction du trafic HTTP généré via Ajax

DEMONSTRATION



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**

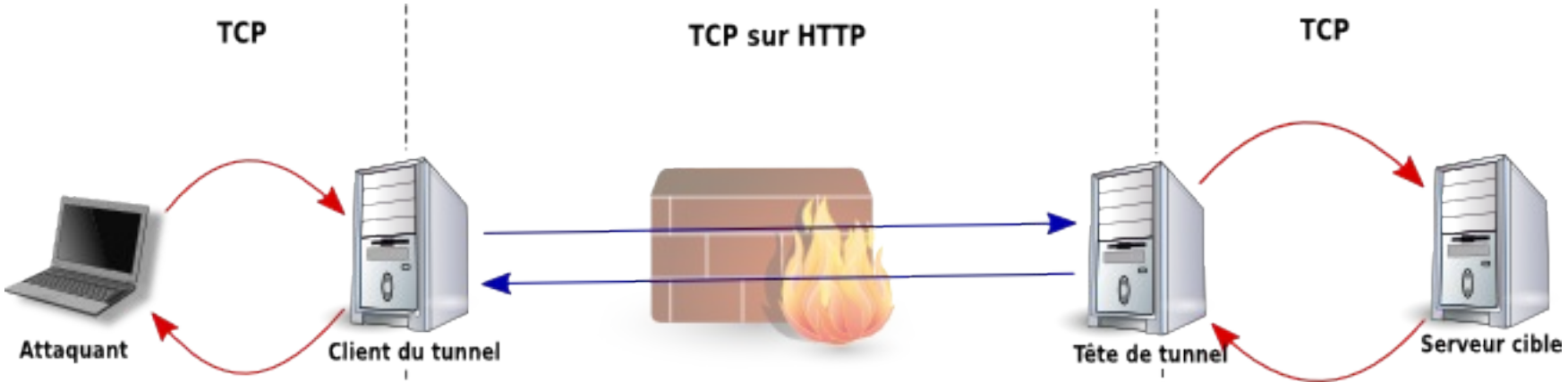
⇒ Intégration d'un scanner TCP au sein du Webshell

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
  - Ouverture d'une véritable invite sur le serveur web sans restrictions
  - Compromission des serveurs d'applications internes (JBoss ...)
  - Interaction directe avec les bases de données internes

Voire même …

- **Prise de contrôle totale du réseau interne ...**
- **Attaques d'autres sites depuis le serveur web**

Comment s'en prémunir ?

Les modes de sécurité (1/2)

- **Les plus connus:**
  - PHP: Safe Mode (disparaît avec PHP 6)
  - JAVA: Security Manager
- **Parfois très limitant:**
  - 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)

- **Délégation de la sécurité à ces modes de sécurité :**
  - 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 = compléments**
- **Application « bien faite » = pas de mode de sécurité.**

Quelques pistes ...

- **Applicatif: faire de la sécurité en amont.**
  - Sensibiliser les développeurs à la sécurité applicative
  - Réaliser des audits de sécurité applicatifs avec le code source **pendant** la phase de développement
- **Configurations serveurs:**
  - **Ne pas faire confiance aux configurations par défaut !**
  - Minimiser les services / applications lancés
  - N'exposer aux utilisateurs que ce qu'ils ont besoin de voir
  - Mettre en place un relai inverse
  - Modifier / Retirer les comptes par défaut
  - Ne pas utiliser de mots de passe triviaux

Conclusion

Conclusion

- **Les attaques applicatives sont bien réelles:**
  - **Parfois facile à mettre en œuvre**
  - Outils existants (C99, R57 ...)
- **Les attaques ne proviennent pas toujours de l'extérieur:**
  - Les serveurs web peuvent être atteints depuis l'interne
  - Les attaquants ne sont pas toujours ceux que l'on croit (MACARON)

Questions ?