Tech-Evenings
Sécurité des applications Web

Sébastien LEBRETON

Pourquoi revoir la sécurité des applications Web

- **Des technologies omniprésentes**
  - ⇒ Facilité de mise en œuvre et de déploiement.
  - ⇒ Commerce en ligne, services Intranet, Extranet et Internet.
  - ⇒ La montée en puissance du Cloud. (90% du budget R&D chez Microsoft).
  - ⇒ La montée en puissance des « WebOs » et la convergence.
- **Des environnements exposés**
  - ⇒ Des données critiques.
  - ⇒ Cible privilégiée des pirates.
- **Réglementations et responsabilités**
  - ⇒ Référentiels et certifications.
  - ⇒ Une société peut engager sa responsabilité.

Les mythes

- Les développeurs codent désormais de façon sécurisée.
  - ⇒ La sécurité d'une application Web n'est pas innée.
- Il faut avoir un niveau avancé pour exploiter les failles.
  - ⇒ Un navigateur est suffisant et de nombreux outils existent.
- Mon site utilise SSL, il est donc sécurisé.
  - ⇒ Le mythe du cadenas, mais la protection n'est que partielle.
- J'ai un firewall, mon site est donc sécurisé.
  - ⇒ Historique de l'infrastructure mais ne couvre pas l'aspect métier.
- Une faille sur une application Intranet est moins importante.
  - ⇒ Un attaquant externe reste capable d'exploiter les failles.

Les outils - prendre conscience du problème

Ils sont très nombreux, mais voici quelques exemples utiles:

- Orientation HTTP/HTML
  - Fiddler, WebScarab: pré-proxy de débogage;
  - Les outils d'aide au développement Web comme Firebug, WebDeveloper;
  - TamperData;
- Orientation réseau
  - WireShark (ex Ethereal), WinPCap: Sniffing;
  - Nmap: le scanneur réseau; (TCP SYN scan = half open scanning)
- Orientation vulnérabilités
  - Nessus, Paros: scanneur réseau et vulnérabilités;
  - Metasploit: framework et tests de vulnérabilités;

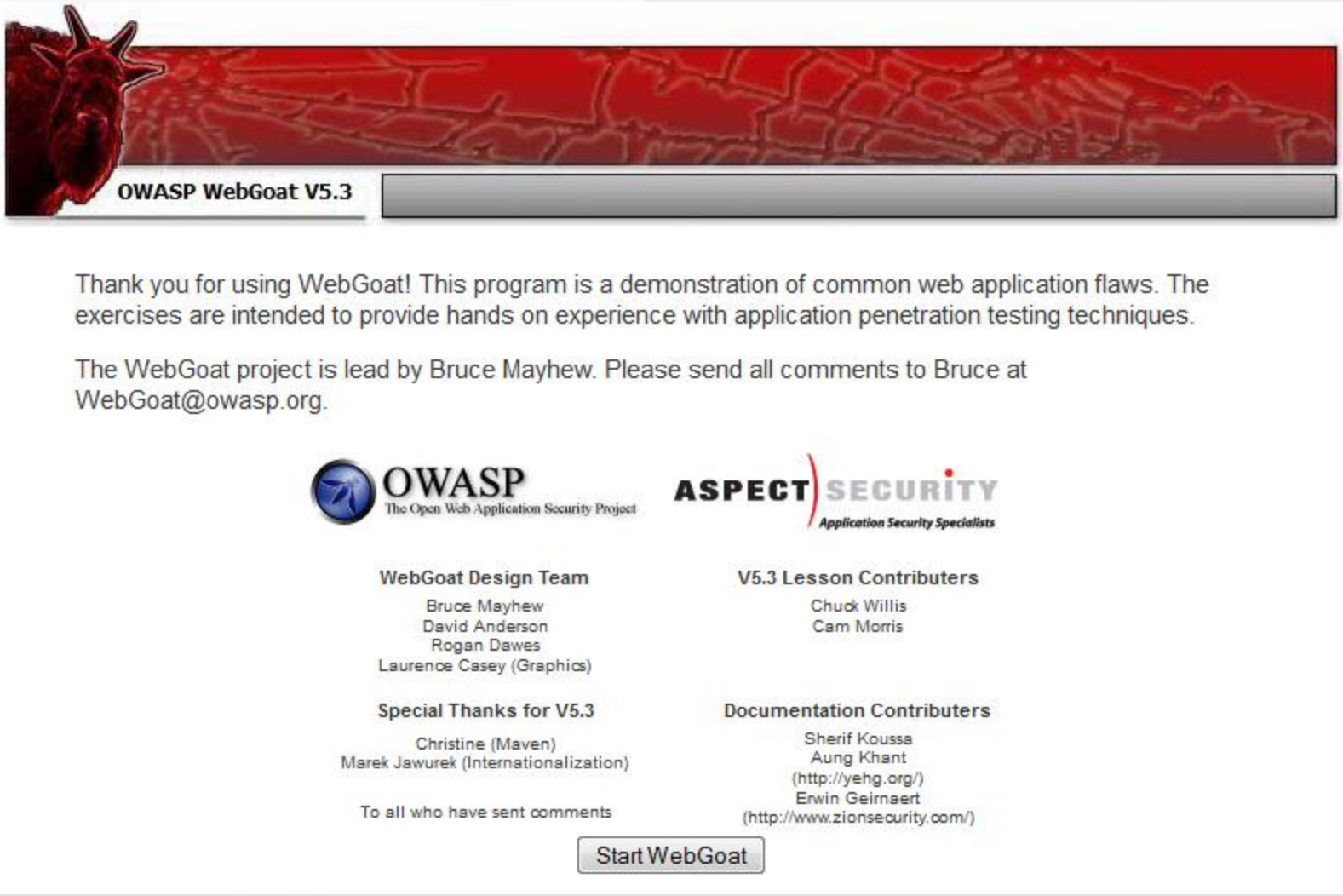
OWASP

- OWASP, pour Open Web Association Security Project.

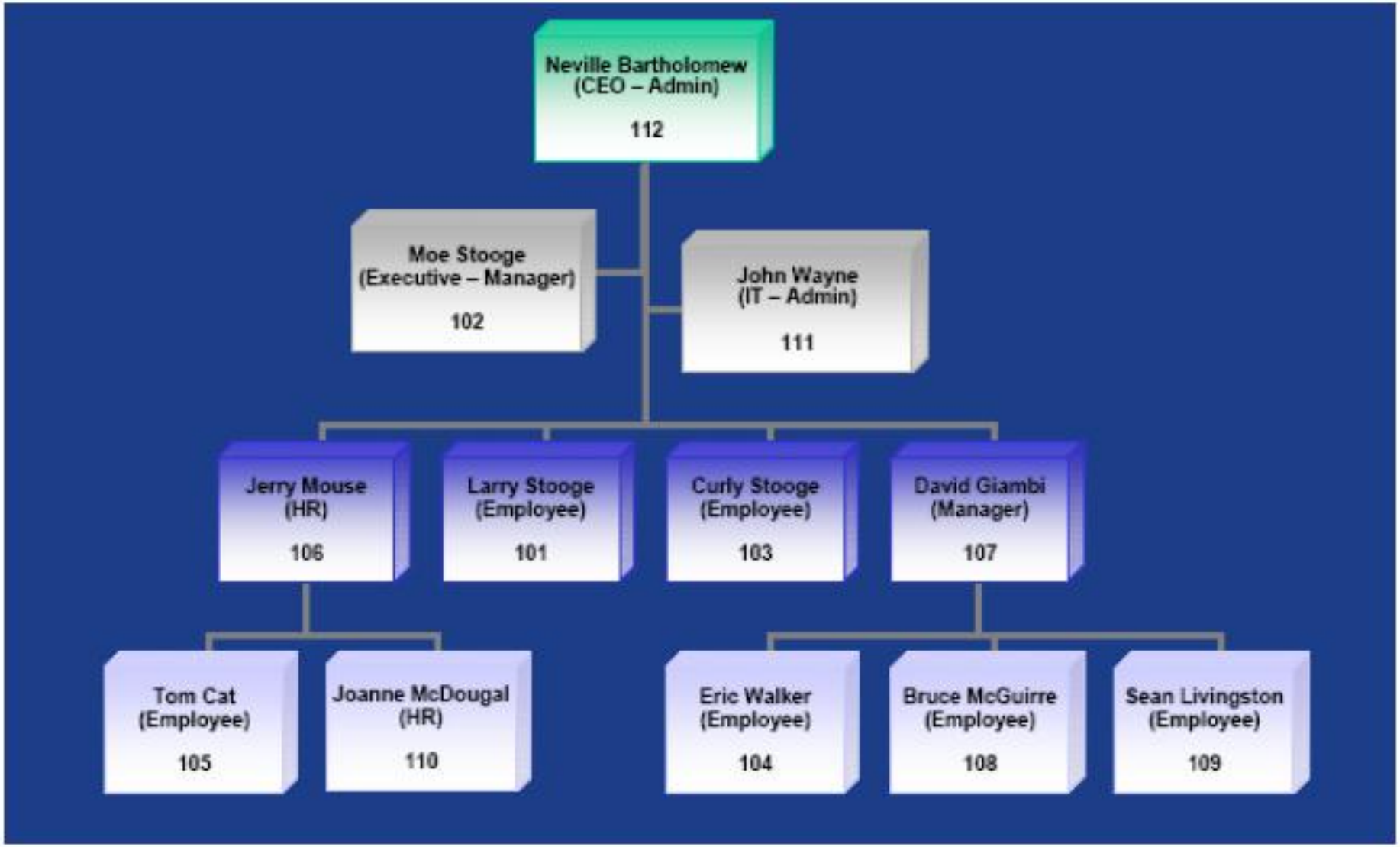
- Présente un Top 10 des risques applicatifs:

A1 – Injection	A2 – Cross-Site Scripting (XSS)	A3 - Violation de Gestion d'Authentification et de Session	A4 - Références directes non sécurisées à un Objet	A5 - Falsification de requête intersite (CSRF)
A6 - Mauvaise configuration Sécurité	A7 - Stockage Cryptographique non Sécurisé	A8 - Manque de Restriction d'Accès URL	A9 - Protection insuffisante de la couche Transport	A10 - Redirections et Renvois non validés

Démonstration OWASP WebGoat



Démonstration OWASP WebGoat



Démonstration OWASP WebGoat

- Overall Policy

Roles \ Assets	Search	List Staff	View Profile	Edit Profile	Create / Delete Profile
Employee	X	X (Self Only)	X	X (Portions)	
Manager	X	X	X		
HR	X	X	X	X (Others Only)	X
Admin	X	X	X	X	X

- Data Access Policy
  - Employees can see their data
  - Employees can edit portions of their data
  - Managers can see their data and their employees' data
  - HR can see and edit all employees. HR cannot edit their data

Définition

- Une faille d'injection se produit quand une donnée non fiable est envoyée à un interpréteur en tant qu'élément d'une commande ou d'une requête.

- Les données hostiles de l'attaquant peuvent tromper l'interpréteur afin de l'amener à exécuter des commandes inattendues ou à accéder à des données non autorisées.

- On parle souvent d'injection SQL mais beaucoup d'autres interpréteurs sont touchés: HTML, LDAP, XPATH, …

Démonstration

A1 – Injection

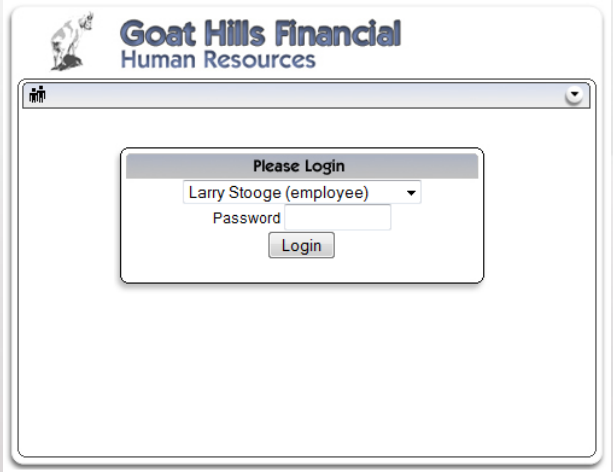
- Comment augmenter la portée d'un résultat avec une Injection SQL (technique du « Select All »);
- Altération des données: (INSERT, UPDATE, DELETE, DROP);
- Recherche dans une table non prévue;

Enter your last name: Your Name Go!

```
SELECT * FROM user_data WHERE last_name = 'Your Name'
```

No results matched. Try Again.

- Comment se connecter sans connaître le mot de passe,
- Comment récupérer le mot de passe



Protection

- Utiliser des ORM ou des systèmes de préparation des requêtes (jamais de concaténation de chaines).
- Encoder toutes les données fournies par l'utilisateur.
- Minimiser les privilèges dans les bases pour limiter l'impact (éviter les failles d'écriture).
- Utiliser des mécanismes de validation côté client ET serveur sur les aspects métier et technique.
- Même si le serveur d'application utilise des filtres, il existe des techniques d'évasion !

Définition

- Les failles XSS se produisent chaque fois qu'une application prend des données non fiables et les envoie à un browser web sans validation appropriée.

- XSS permet à des attaquants d'exécuter du script dans le navigateur de la victime afin de détourner des sessions utilisateur, défigurer des sites web, ou rediriger l'utilisateur vers des sites malveillants.

Démonstration

- Comment voler l'ID de session ou mettre en place un lien vers un site de phishing avec une injection XSS :

Protection

A2 – Cross-Site Scripting (XSS)

- Utiliser des filtres de nettoyage HTML si du HTML est attendu (texte riche).
- Encoder toutes les données fournies par l'utilisateur.
- Utiliser des mécanismes de validation côté client ET serveur sur les aspects métier et technique.
- Les attributs « **ValidateInput** » et « **AllowHtml** » permettent de paramétrer ce comportement dans ASP.NET MVC.

Définition

A3 - Violation de Gestion d'Authentification et de Session

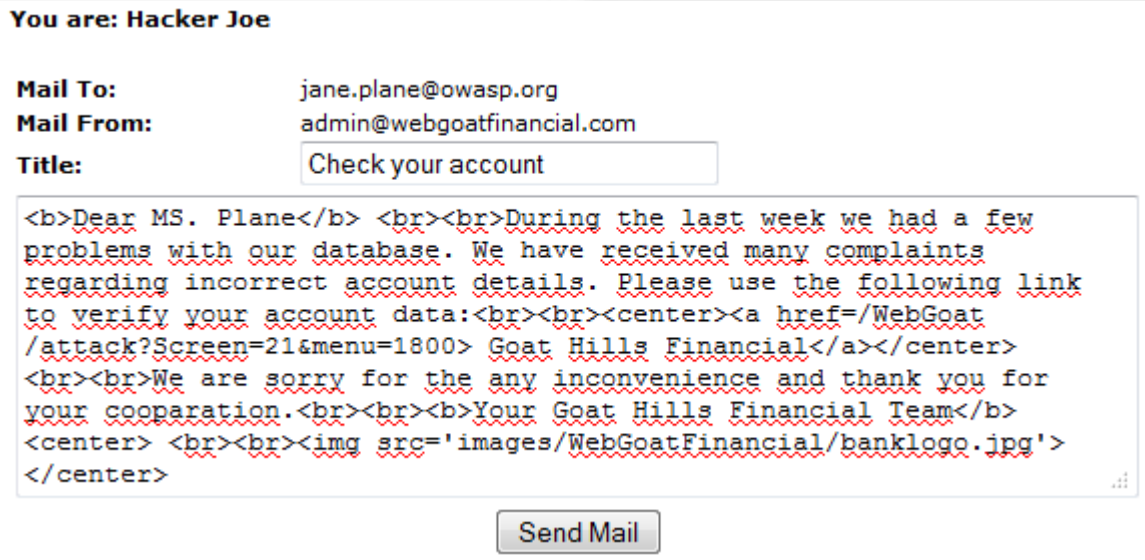
- Les fonctions applicatives relatives à l'authentification et la gestion de session ne sont souvent pas mises en œuvre correctement.

- Ceci permettant aux attaquants de compromettre les mots de passe, clés, jetons de session, ou d'exploiter d'autres failles d'implémentation pour s'approprier les identités d'autres utilisateurs.

Démonstration

A3 - Violation de Gestion d'Authentification et de Session

- Utilisation la technique de « Session Fixation » puis « Session » afin de voler la session d'un utilisateur :



Protection – 1/2

A3 - Violation de Gestion d'Authentification et de Session

- Les jetons de session ne doivent pas apparaitre dans les URL. (Attaques liées aux techniques de « Session Fixation »). Il ne doivent pas être acceptés depuis des variables GET/POST.
- Les sessions doivent expirer et il doit être possible de se déconnecter.
- Utiliser constamment SSL pour protéger les identifiants et les sessions.
- Les mesures de réinitialisation de mots de passe doivent être strictement élaborées pour ne permettre à aucun tiers de prendre le contrôle d'un compte utilisateur.

Protection – 2/2

A3 - Violation de Gestion d'Authentification et de Session

- Les mots de passes doivent être vérifiés suivant une politique de complexité minimale. Ils doivent expirer.

- Les mots de passes éventuellement générés doivent utiliser un algorithme non prédictif.

Définition

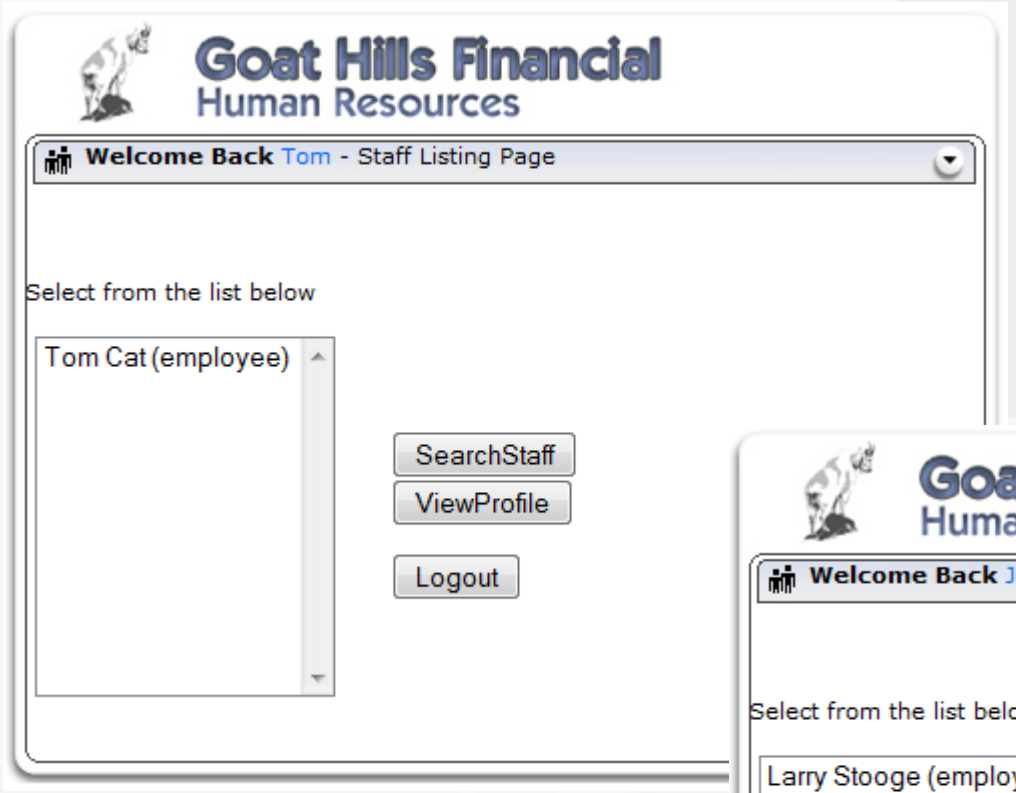
A4 - Références directes non sécurisées à un Objet

- Une référence directe à un objet se produit quand un développeur expose une référence à un objet d'exécution interne, tel un fichier, un dossier, un enregistrement de base de données, ou une clé de base de données.

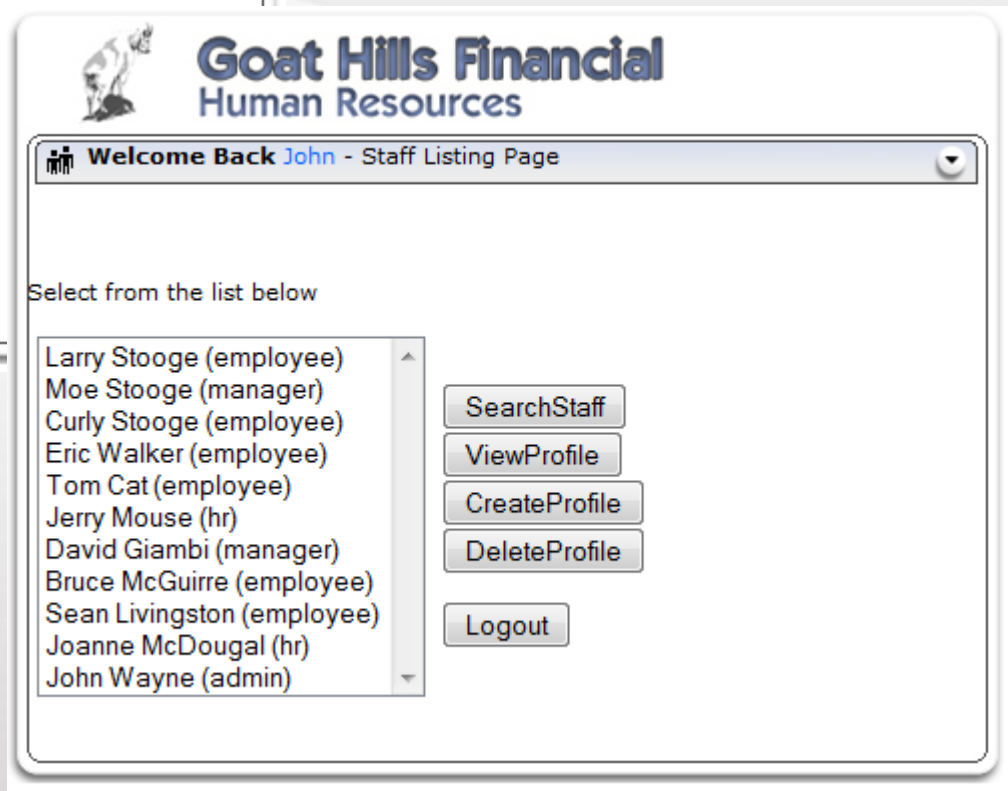
- Sans un contrôle d'accès ou autre protection, les attaquants peuvent manipuler ces références pour accéder à des données non autorisées.

Démonstration

A4 - Références directes non sécurisées à un Objet



- Comment utiliser une fonctionnalité métier non disponible pour un utilisateur.



- Comment consulter le profil d'un autre utilisateur.

Protection – 1/2

A4 - Références directes non sécurisées à un Objet

- Toutes les références aux objets (pages, fichiers, …) doivent être validées : l'utilisateur a-t-il le droit d'accès et dans quel mode (lecture, écriture, suppression, …).

- Idéalement les références directes (clefs primaires en base, etc.) ne sont jamais exposées, on utilise des dictionnaires de références indirectes, par utilisateur ou par session.

- L'association vers la référence directe est faite côté serveur avec un algorithme non prédictif (par exemple une table de hachage avec une clef numérique aléatoire).

- Donner des références directes c'est permettre à un éventuel attaquant de connaitre plus facilement le système.

Protection – 2/2

A4 - Références directes non sécurisées à un Objet

- Canoniser puis vérifier les chemins des fichiers dans le cas de documents téléchargeables avec un nom passé en paramètre d'une URL.

- Ceci permet d'éviter de forger de requêtes visant à récupérer des fichiers sensibles (configuration, code source des pages, etc.)

Définition

A5 - Falsification de requête intersite (CSRF)

- Une attaque CSRF (Cross Site Request Forgery) force le navigateur d'une victime authentifiée à envoyer une requête http forgée, comprenant le cookie de session de la victime ainsi que toute autre information automatiquement inclue, à une application web vulnérable.

- Cette vulnérabilité est causée par la capacité que les navigateurs ont d'envoyer automatiquement des données d'authentification (session, comptes de domaine windows, ..) dans chaque requête.

Démonstration

A5 - Falsification de requête intersite (CSRF)

- Technique CSRF contre une application:

Protection

A5 - Falsification de requête intersite (CSRF)

- Vérifier les pages référentes pour savoir si la requête provient du bon site. (Attention car possibilité de spoof).
- Utiliser un mécanisme de jeton de vérification : « **AntiForgeryToken** » pour ASP.NET MVC ou « **EventValidation** » pour ASP.NET classique.
- Ce système utilise un couple « cookie/hidden input » et le serveur vérifie leur concordance.
- Le cookie utilise le flag « HTTPOnly » pour empêcher une manipulation par scripting.

Définition

A6 - Mauvaise configuration Sécurité

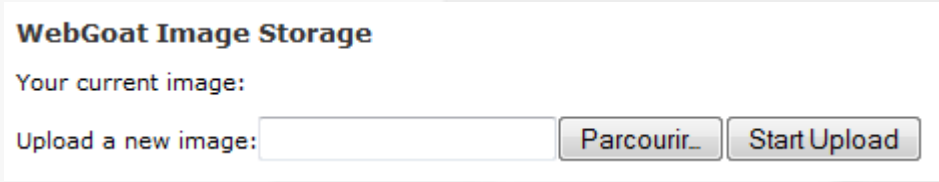
- Une bonne sécurité exige d'avoir une configuration sécurisée définie et déployée pour l'application, les serveurs d'application et de base de données ainsi que la plate-forme.

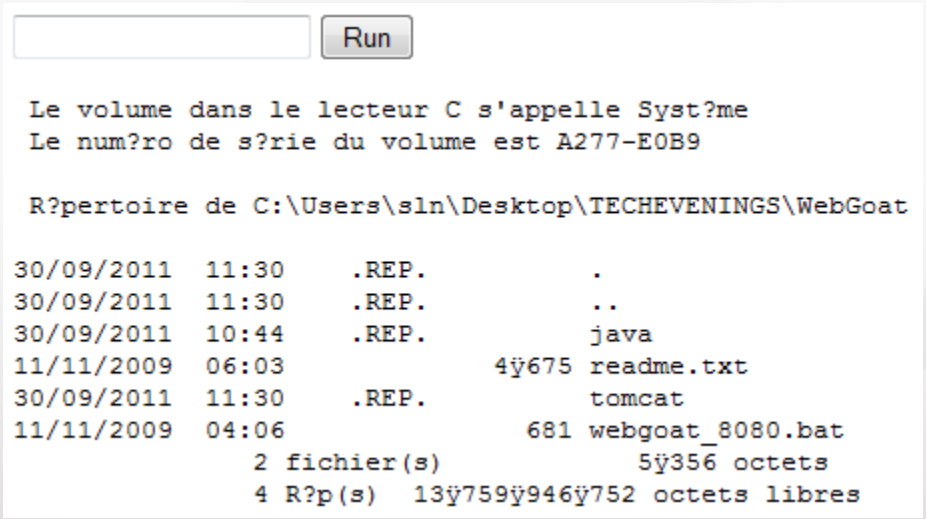
- Attention aux mots de passe de connexion aux bases de données trop souvent disponibles en clair dans les fichiers de configuration.

Démonstration

A6 - Mauvaise configuration Sécurité

- Profiter d'un système d'upload de fichiers pour exécuter du code malicieux côté serveur:





Protection - 1/2

A6 - Mauvaise configuration Sécurité

- Mettre en place un processus itératif de mise à niveau logiciel (système d'exploitation, Serveur Web, bases de données, composants applicatifs).
- Supprimer tous les services non utilisés, restreindre les ports avec un firewall.
- Utiliser des IDS (détecteur d'intrusions) capables de repérer et bloquer l'accès à des attaquants.
- Les mots de passes dans les fichiers de paramétrage type « web.config » sont à proscrire : utilisation des systèmes de sécurité intégrés ou cryptage des chaînes de connexion avec « RsaProtectedConfigurationProvider ».

Protection - 2/2

A6 - Mauvaise configuration Sécurité

- Tracer les attaques « métier » et bloquer par palier les utilisateurs et IP associées.
- Modifier tous les mots de passes par défaut.
- Le système de gestion des erreurs ne doit jamais exposer la pile aux utilisateurs.
- Signer avec un certificat les assemblages pour leur conférer une identité et garantir leur non usurpation par un tiers.
- Utiliser des scanners de vulnérabilités.

Définition

A7 - Stockage Cryptographique non Sécurisé

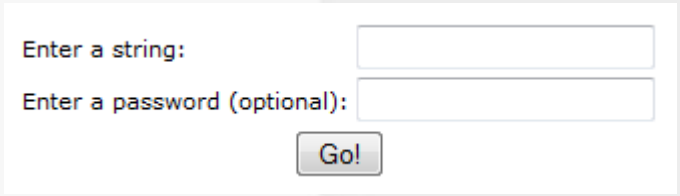
- Beaucoup d'applications web ne protègent pas correctement les données sensibles comme les informations d'authentification avec un chiffrement ou un hash approprié.

- Les attaquants peuvent voler ou modifier ces données faiblement protégées afin d'usurper une identité.

Démonstration

A7 - Stockage Cryptographique non Sécurisé

- Sensibilisation au « hashing » et au « salting »;



MD5HASHCRACKER

PLAINTEXT	MD5	SHA1
hotheaded	d4f26cca6f2a2c89514a28f31fb5e74f	119efb725d608cf11562e82594855cef132cdeda
hotheadedly	ed199302ee1a8df596baa711fb0bb828	b6f853479c70b0990679a05418896263d8c62319
hotheadedness	79871453ac2ceb43b6ec8e9aefcfd4a9	1a1c0b78516bf3cd5d7fc342e692278eb4aaecce
hothearted	ff98fd14abc2fc681a572d95d7e7290e	f433ec374ca52e6c2cf8b363687f6affa72c6397
hotheartedly	53b48808ce9846d77b2e770b6c9372c0	17fd9eef88cbcb3b50969ff48d43f56ae026d00e
hotheartedness	a02f704f4362bb6ebf426d34ec6a496a	479a8bc929f3260379541b42b4dfe29b33f6e3a7
hothir	7ff3cf32a0484e685ccd4d66084ff9a5	82d50e52c68b77c9a4380a78d0663a652dd126a9
hothouse	489a31042eb1cb80df59b8eeda0013d8	6db9c4e8e670b3a716c7108b44bd5fa4dc38b4a7
hoti	0f360f13ce55a64f07d5e55fb1767044	8adf0e8abf49cbe8d90728c32504d58e668e3220

Protection

A7 - Stockage Cryptographique non Sécurisé

- Utiliser des hashs forts pour la gestion des mots de passe (SHA-1, SHA-2). Utiliser des salts pour créer des variations.
- Chiffrer toutes les données sensibles (fichiers, etc.).
- Les sauvegardes doivent être également chiffrées.
- Une clef forte de chiffrement est utilisée. Cette dernière doit être protégée et un processus de changement de la clef est défini puis exécuté périodiquement.

Définition

A8 - Manque de Restriction d'Accès URL

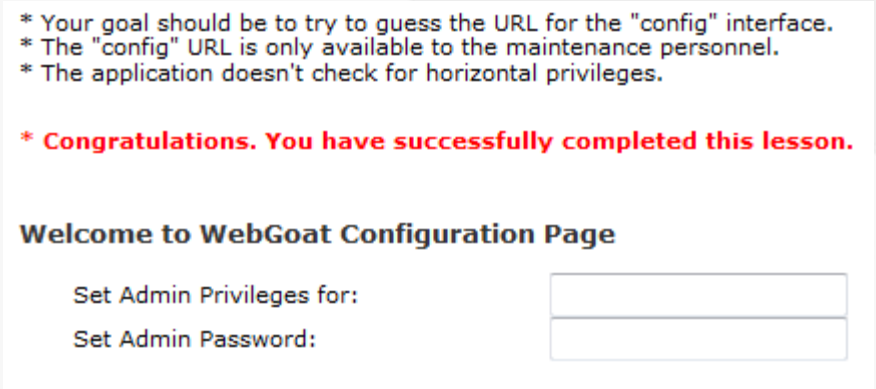
- Beaucoup d'applications web vérifient les droits d'accès URL avant de rendre les liens protégés.

- Cependant, les applications doivent effectuer des contrôles d'accès similaires chaque fois que ces pages sont accédées, ou les attaquants seront en mesure de forger des URL pour accéder à ces pages cachées de toute façon.

Démonstration

A8 - Manque de Restriction d'Accès URL

- Technique du « Forced Browsing » :



Protection

A8 - Manque de Restriction d'Accès URL

- Tout accès à une page entraine la vérification du contexte, de la session et des droits.
- Les politiques d'authentification et d'autorisation doivent être basées sur les rôles, afin de minimiser l'effort nécessaire lors de leur maintenance.
- Les politiques doivent être hautement configurables, afin de minimiser les aspects codés en dur.
- Si la page est impliquée dans un workflow, assurez-vous que toutes les conditions soient réunies pour permettre l'accès.

Définition

A9 - Protection insuffisante de la couche Transport

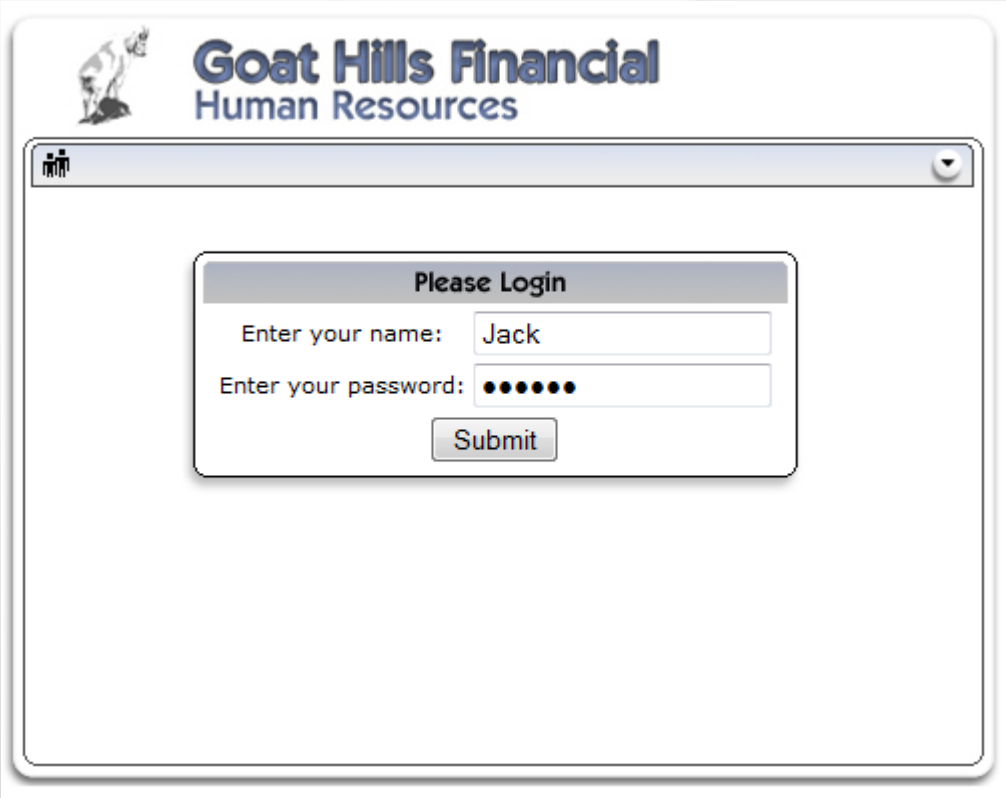
- Les applications ont souvent du mal à authentifier, chiffrer et protéger la confidentialité et l'intégrité d'un trafic réseau sensible.

- Quand elles le font, elles supportent parfois des algorithmes faibles ou utilisent des certificats expirés ou invalides.

Démonstration

A9 - Protection insuffisante de la couche Transport

- Comment sniffer le trafic réseau pour récupérer un id de session ou un mot de passe.
- Il existe des techniques pour sniffer un réseau « switché », par exemple avec « l'ARP Spoofing » et « l'ARP Poisoning »



Protection

A9 - Protection insuffisante de la couche Transport

- Utiliser le protocole SSL pour toutes les pages sensibles et forcer les requêtes non SSL à basculer sur la version sécurisée des pages.
- Utiliser le « secure flag » sur les cookies sensibles, ce qui va forcer leur utilisation dans le cadre d'une connexion sécurisée.
- Configurer SSL pour n'utiliser que des algorithmes forts.
- Vérifier les certificats.
- La couche de présentation n'est pas le seul point d'entrée, sécuriser la couche de service également avec du SSL et de l'authentification.

Définition

A10 - Redirections et Renvois non validés

- Les applications web réorientent et font suivre fréquemment les utilisateurs vers d'autres pages et sites web, et utilisent des données non fiables pour déterminer les pages de destination.

- Sans validation appropriée, les attaquants peuvent rediriger les victimes vers des sites de phishing, ou utiliser les renvois pour accéder à des pages non autorisées.

Démonstration

A10 - Redirections et Renvois non validés

- Avec les redirections (Redirect)

  http://site.fr/redirect.aspx?url=jesuismechant.com

  Attention au phishing!

- Avec les renvois (Forward / Transfer)

  http://site.fr/visiteur.jsp?fwd=admin.jsp

  Attention à la vérification de droits.

- Utilisation des techniques de « HTTP Splitting » et « Cache Poisoning ».
  - Ce sont des attaques pointues qui ne sont pas détectées par les IDS !
  - Tout repose sur le protocole HTTP 1.1 permet d'échanger de multiples requêtes dans la même session TCP.

Protection

A10 - Redirections et Renvois non validés

- Limiter les redirections et les transferts.
- Ne pas utiliser des paramètres pour définir l'URL ou la fonction cible.
- Sinon valider ces paramètres en fonction des rôles et des droits de l'utilisateur.
- Valider les URL cibles après construction pour vérifier qu'elles pointent bien vers un site autorisé.

Questions ?