La sécurité des extensions

The Mozilla Art Of War

David Teller

Laboratoire d'Informatique Fondamentale d'Orléans

20 septembre 2008

Firefox, c'est sûr

Firefox, c'est sûr

"Donc je vais faire mes extensions sous Firefox, elles seront en sécurité."

Firefox, c'est sûr

"Donc je vais faire mes extensions sous Firefox, elles seront en sécurité."
Hein ?

Firefox, c'est sûr

"Donc je vais faire mes extensions sous Firefox, elles seront en sécurité."

Hein ?

Il est temps de faire peur à ces développeurs d'extensions.

Attention

Attention

Écouter cet exposé peut vous amener en prison.

Attention

Écouter cet exposé peut vous amener en prison.
(merci Dadvsi)

En fait, Firefox, c'est...

Que sait faire Firefox ?

En fait, Firefox, c'est...

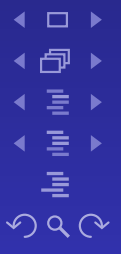
Que sait faire Firefox ?

1. Résoudre des noms, négocier des connexions.
2. Télécharger des trucs.
3. Installer des extensions.

En fait, Firefox, c'est...

Que sait faire Firefox ?

1. Résoudre des noms, négocier des connexions.
2. Télécharger des trucs.
3. Installer des extensions.
4. Exécuter du JavaScript.
5. Exécuter des templates.



En fait, Firefox, c'est...

Que sait faire Firefox ?

1. Résoudre des noms, négocier des connexions.
2. Télécharger des trucs.
3. Installer des extensions.
4. Exécuter du JavaScript.
5. Exécuter des templates.
6. Afficher des images, des animations, jouer des musiques.
7. Afficher des pages web, des interfaces xul.

En fait, Firefox, c'est...

Que sait faire Firefox ?

1. Résoudre des noms, négocier des connexions.
2. Télécharger des trucs.
3. Installer des extensions.
4. Exécuter du JavaScript.
5. Exécuter des templates.
6. Afficher des images, des animations, jouer des musiques.
7. Afficher des pages web, des interfaces xul.
8. Stocker des cookies, des relations, des triplets...
9. Recharger des cookies, des relations, des triplets...

En d'autres termes, Firefox, c'est...

Que sait faire Firefox ?

En d'autres termes, Firefox, c'est...

Que sait faire Firefox ?

1 – 3 Installer et charger des applications (locales et client-serveur).

En d'autres termes, Firefox, c'est...

Que sait faire Firefox ?

1 – 3 Installer et charger des applications (locales et client-serveur).

4 – 5 Exécuter des applications.

En d'autres termes, Firefox, c'est...

Que sait faire Firefox ?

1 – 3 Installer et charger des applications (locales et client-serveur).

4 – 5 Exécuter des applications.

6 – 7 Proposer des interfaces graphiques.

En d'autres termes, Firefox, c'est...

Que sait faire Firefox ?

1 – 3 Installer et charger des applications (locales et client-serveur).

4 – 5 Exécuter des applications.

6 – 7 Proposer des interfaces graphiques.

8 – 9 Stocker et relire des informations.

En d'autres termes, Firefox, c'est...

Que sait faire Firefox ?

1 – 3 Installer et charger des applications (locales et client-serveur).

4 – 5 Exécuter des applications.

6 – 7 Proposer des interfaces graphiques.

8 – 9 Stocker et relire des informations.

Ça ne vous rappelle rien ?

The browser is the OS

The browser is the OS

Avec de grands pouvoirs, viennent quelques questions :

- Les applications peuvent-elles être compromises ?
- Le système peut-il être compromis par une application locale ?
- Le système peut-il être compromis à distance ?

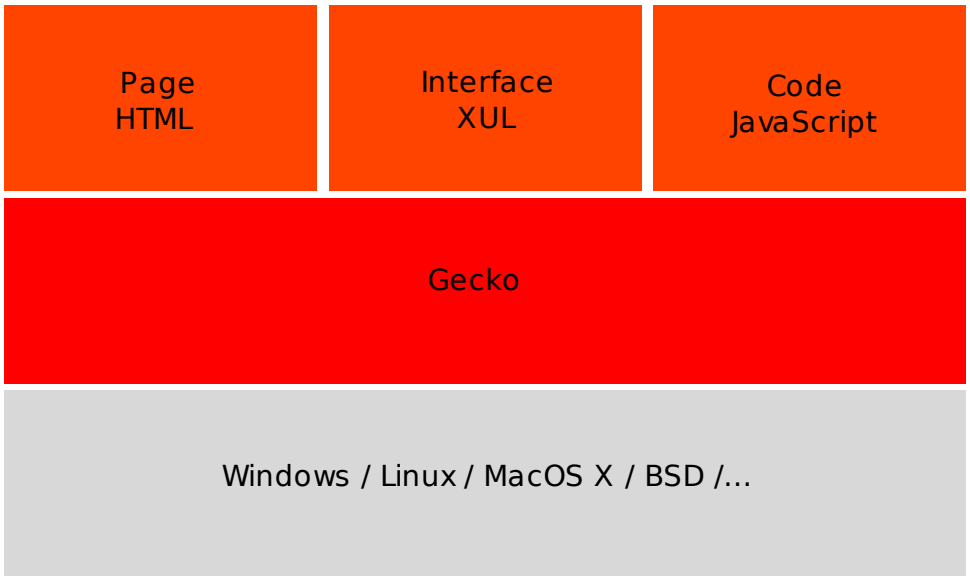
The browser is the OS

Avec de grands pouvoirs, viennent quelques questions :

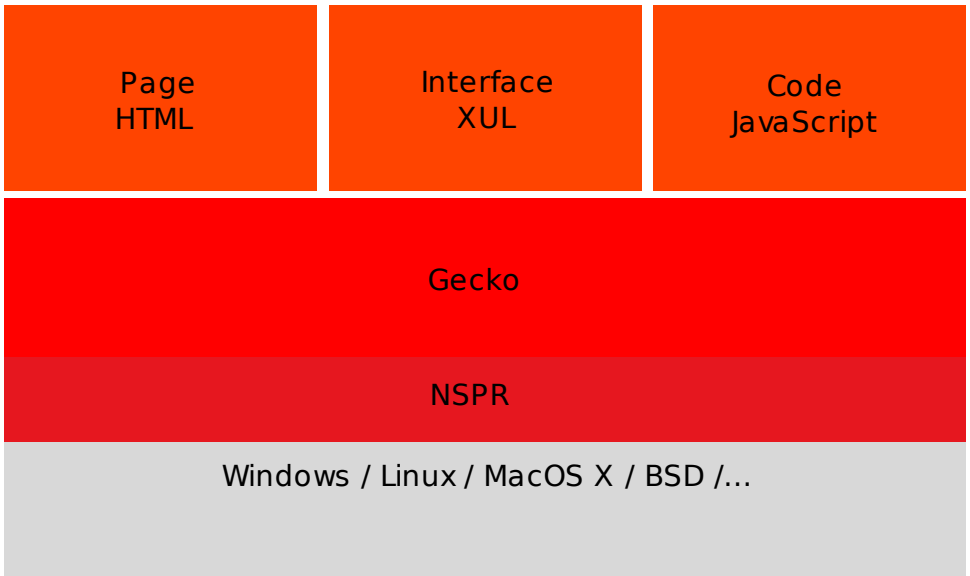
- Les applications peuvent-elles être compromises ?
- Le système peut-il être compromis par une application locale ?
- Le système peut-il être compromis à distance ?

Note Nous parlons du modèle de sécurité, pas d'éventuels bugs.

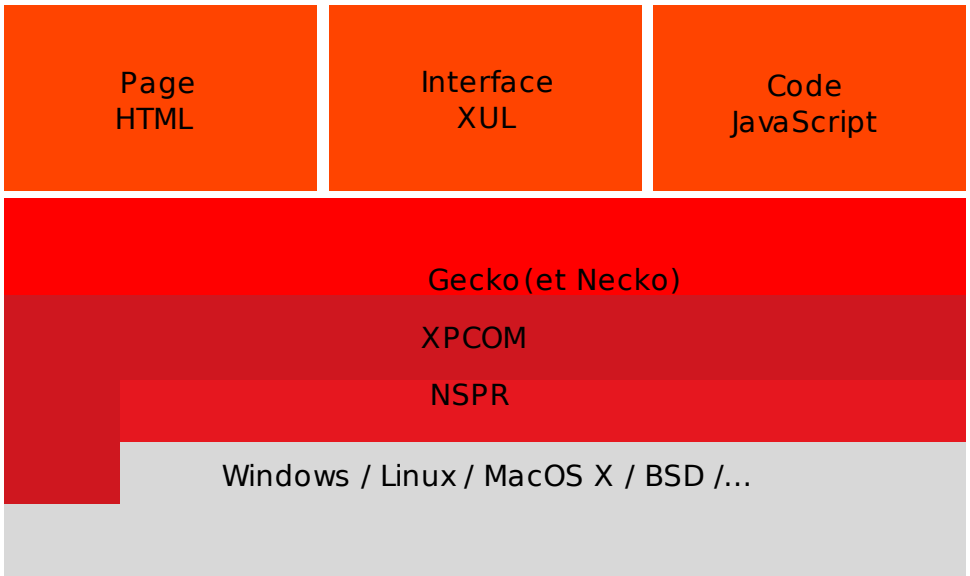
Firefox pour les adultes



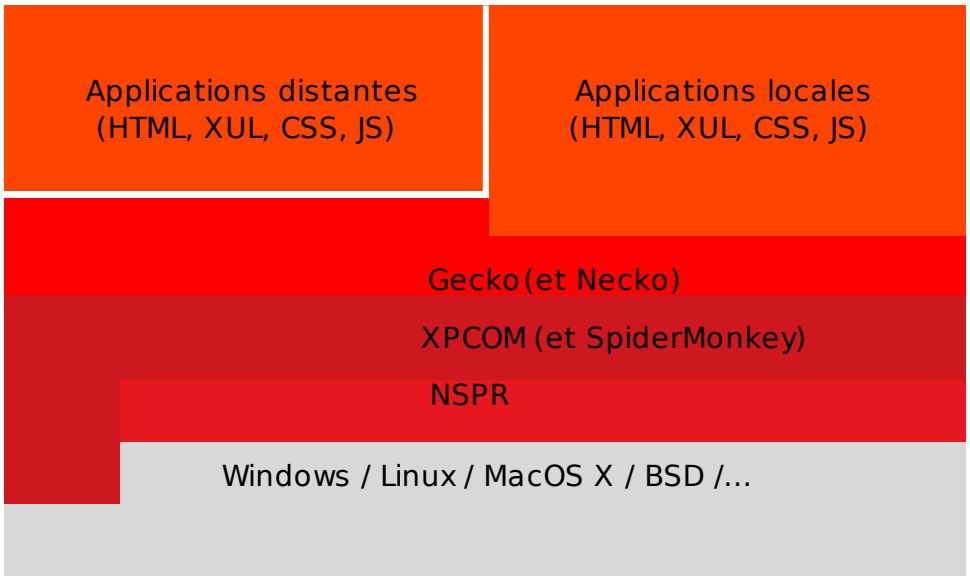
Firefox pour les adultes



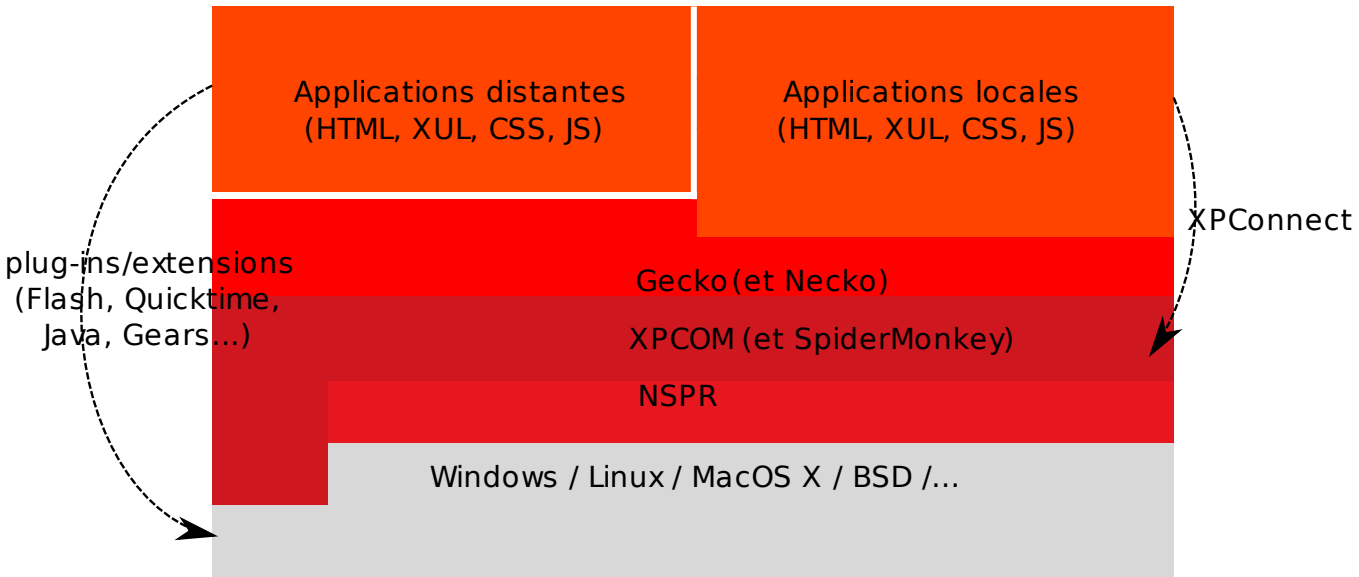
Firefox pour les adultes



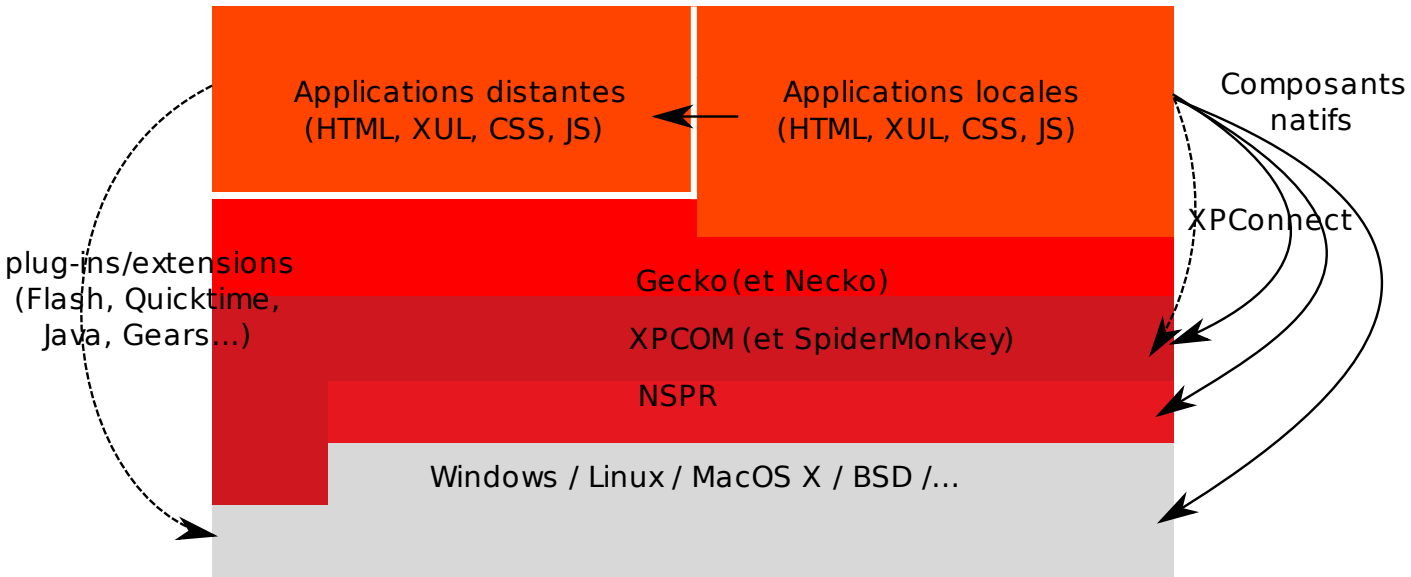
Firefox pour les adultes



Firefox pour les adultes



Firefox pour les adultes



Quelques critères de sécurité

Définition Un système est sécurisé si ce qui n'est pas autorisé ne peut pas arriver.

Quelques critères de sécurité

Définition Un système est sécurisé si ce qui n'est pas autorisé ne peut pas arriver.

Confidentialité Une application peut lire les informations secrètes d'une autre ?

Intégrité Une application peut-elle modifier le fonctionnement d'une autre ?

Quelques critères de sécurité

Définition Un système est sécurisé si ce qui n'est pas autorisé ne peut pas arriver.

Confidentialité Une application peut lire les informations secrètes d'une autre ?

Intégrité Une application peut-elle modifier le fonctionnement d'une autre ?

Échanges Une application peut-elle communiquer avec une autre sans risque ?

Limites Où s'arrête le modèle ?

Le modèle Unix

Confidentialité Applications isolées par défaut. Mémoire partagée sur demande.

Intégrité Applications isolées par défaut. Mémoire partagée sur demande.

Le modèle Unix

Confidentialité Applications isolées par défaut. Mémoire partagée sur demande.

Intégrité Applications isolées par défaut. Mémoire partagée sur demande.

Échanges Bibliothèques de communication, avec contrôle.

Le modèle Unix

Confidentialité	Applications isolées par défaut. Mémoire partagée sur demande.
Intégrité	Applications isolées par défaut. Mémoire partagée sur demande.
Échanges	Bibliothèques de communication, avec contrôle.
Limites	Accès aux ressources pas réellement isolé

Le modèle Unix

Confidentialité	Applications isolées par défaut. Mémoire partagée sur demande.
Intégrité	Applications isolées par défaut. Mémoire partagée sur demande.
Échanges	Bibliothèques de communication, avec contrôle.
Limites	Accès aux ressources pas réellement isolé

$\Rightarrow$ Interactions lourdes mais plutôt robustes.

Le modèle Mozilla

Confidentialité Informations globales cachées uniquement *par capacité* ou pas du tout.

Les clôtures permettent de cacher des choses.

Intégrité Un objet connu peut généralement être modifié..

Le modèle Mozilla

Confidentialité Informations globales cachées uniquement *par capacité* ou pas du tout.

Les clôtures permettent de cacher des choses.

Intégrité Un objet connu peut généralement être modifié..

Échanges `window`, évènements.

Le modèle Mozilla

Confidentialité Informations globales cachées uniquement *par capacité* ou pas du tout.

Les clôtures permettent de cacher des choses.

Intégrité Un objet connu peut généralement être modifié..

Échanges `window`, évènements.

Limites Accès à XPCOM pas réellement isolé.

Le modèle Mozilla

Confidentialité Informations globales cachées uniquement *par capacité* ou pas du tout.

Les clôtures permettent de cacher des choses.

Intégrité Un objet connu peut généralement être modifié..

Échanges `window`, évènements.

Limites Accès à XPCOM pas réellement isolé.

C++ omnipotent.

Le modèle Mozilla

Confidentialité Informations globales cachées uniquement *par capacité* ou pas du tout.

Les clôtures permettent de cacher des choses.

Intégrité Un objet connu peut généralement être modifié..

Échanges `window`, évènements.

Limites Accès à XPCOM pas réellement isolé.

C++ omnipotent.

$\Rightarrow$ Interactions riches mais dangereuses.

Filtre anti-phishing ?

Ce que l'extension devrait faire Vérifier si l'utilisateur visite uniquement des sites sûrs :

```
function AntiPhishing() {
}

AntiPhishing.prototype = {
        displaySafety : function(e)
        {
                var browser = getBrowser();
                if(checkSafetySomehow(browser.currentURI))
                        //laisser passer
                else
                        //refuser l'adresse

        }
}

getBrowser().addEventListener("onload", function(e){antiPhising.displaySafety(e)});
```


Ou filtre pro-phishing ?

Ou après piratage... Faire croire que `"http://break.your-computer.xxx"` est sûr :

```
var oldDisplaySafety = AntiPhishing.prototype.displaySafety;

AntiPhishing.prototype.displaySafety = function(e) {
        var browser = getBrowser();
        if(browser.currentURI.host == "break.your-computer.xxx")
                //accepter l'adresse
        else
                oldDisplaySafety.apply(this, [e]);
}
```

Recette générale

1. explorer `window`, surcharger `Array.push`, `String.indexOf`...

Recette générale

1. explorer `window`, surcharger `Array.push`, `String.indexOf`...
2. trouver un objet de l'extension
3. trouver son prototype

Recette générale

1. explorer `window`, surcharger `Array.push`, `String.indexOf`...
2. trouver un objet de l'extension
3. trouver son prototype
4. lire les propriétés éventuellement secrètes
   - 4.1 directement
   - 4.2 subtilement, avec `Object.toSource` / `Object.watch`
   - 4.3 violemment, à coups de débogueur ou C++

Recette générale

1. explorer `window`, surcharger `Array.push`, `String.indexOf`...
2. trouver un objet de l'extension
3. trouver son prototype
4. lire les propriétés éventuellement secrètes
   - 4.1 directement
   - 4.2 subtilement, avec `Object.toSource` / `Object.watch`
   - 4.3 violemment, à coups de débogueur ou C++
5. si nécessaire,
   - 5.1 remplacer les méthodes de manière transparente
   - 5.2 remplacer les propriétés par des accesseurs transparents
   - 5.3 s'en servir pour trouver d'autres objets
   - 5.4 reprendre à 2

Recette générale

1. explorer `window`, surcharger `Array.push`, `String.indexOf`...
2. trouver un objet de l'extension
3. trouver son prototype
4. lire les propriétés éventuellement secrètes
   - 4.1 directement
   - 4.2 subtilement, avec `Object.toSource` / `Object.watch`
   - 4.3 violemment, à coups de débogueur ou C++
5. si nécessaire,
   - 5.1 remplacer les méthodes de manière transparente
   - 5.2 remplacer les propriétés par des accesseurs transparents
   - 5.3 s'en servir pour trouver d'autres objets
   - 5.4 reprendre à 2
6. si nécessaire
   - 6.1 remonter au contexte à l'aide de `__parent__`
   - 6.2 reprendre à 2

Pour s'en prémunir

- cloisonner le code à coups de
  - clôtures, `const`, `__defineGetter__`, `__defineSetter__`, `__noSuchMethod__`
  - composants XPCom
  - modules `jsm`
- s'en servir pour couper les chemins depuis le domaine partagé vers les objets internes au module
- surcharger `toSource` et `watch`

Pour s'en prémunir

- cloisonner le code à coups de
  - clôtures, `const`, `__defineGetter__`, `__defineSetter__`, `__noSuchMethod__`
  - composants XPCom
  - modules `jsm`
- s'en servir pour couper les chemins depuis le domaine partagé vers les objets internes au module
- surcharger `toSource` et `watch`
- ne passer aucun objet important en argument à du JavaScript public ou du XPCom

Pour s'en prémunir

- cloisonner le code à coups de
  - clôtures, `const`, `__defineGetter__`, `__defineSetter__`, `__noSuchMethod__`
  - composants XPCOM
  - modules `jsm`
- s'en servir pour couper les chemins depuis le domaine partagé vers les objets internes au module
- surcharger `toSource` et `watch`
- ne passer aucun objet important en argument à du JavaScript public ou du XPCOM
- pour plus de paranoïa, vous pouvez aussi
  - désactiver le débogage *régulièrement*
  - vérifier avec `caller` qui invoque vos méthodes critiques.

Bilan du round 1

Non Le modèle de sécurité n'est pas prévu pour protéger une application locale d'une autre.

Bilan du round 1

Non Le modèle de sécurité n'est pas prévu pour protéger une application locale d'une autre.

Oui Sous JS, vos valeurs privées peuvent rester privées.

Bilan du round 1

Non Le modèle de sécurité n'est pas prévu pour protéger une application locale d'une autre.

Oui Sous JS, vos valeurs privées peuvent rester privées.

Non Votre ennemi juré pourra toujours supprimer votre extension ou l'empêcher de fonctionner.

Non Vous ne pourrez pas vous défendre contre C++.

Bilan du round 1

Non	Le modèle de sécurité n'est pas prévu pour protéger une application locale d'une autre.
Oui	Sous JS, vos valeurs privées peuvent rester privées.
Non	Votre ennemi juré pourra toujours supprimer votre extension ou l'empêcher de fonctionner.
Non	Vous ne pourrez pas vous défendre contre C++.
Attention	Votre ennemi juré pourra *toujours* s'enfoncer plus profondément dans le système.

Le modèle Mozilla

Confidentialité Un script ne peut pas consulter des informations d'un script d'origine différente.

Intégrité Deux scripts d'origine différente ne peuvent pas interagir.

Le modèle Mozilla

Confidentialité Un script ne peut pas consulter des informations d'un script d'origine différente.

Intégrité Deux scripts d'origine différente ne peuvent pas interagir.

Échanges Deux scripts d'origine différente ne peuvent pas interagir.

Le modèle Mozilla

Confidentialité Un script ne peut pas consulter des informations d'un script d'origine différente.

Intégrité Deux scripts d'origine différente ne peuvent pas interagir.

Échanges Deux scripts d'origine différente ne peuvent pas interagir.

Limites Le serveur ou le site peut prétendre que deux scripts ont la même origine.

Le modèle Mozilla

Confidentialité Un script ne peut pas consulter des informations d'un script d'origine différente.

Intégrité Deux scripts d'origine différente ne peuvent pas interagir.

Échanges Deux scripts d'origine différente ne peuvent pas interagir.

Limites Le serveur ou le site peut prétendre que deux scripts ont la même origine.

$\Rightarrow$ Sans hack, pas d'interactions du tout.

Le modèle Mozilla (suite)

Une fois que les interactions sont possibles. . .

Confidentialité Comme applications locales.

Intégrité Comme applications locales.

Le modèle Mozilla (suite)

Une fois que les interactions sont possibles. . .

Confidentialité Comme applications locales.

Intégrité Comme applications locales.

Échanges Comme applications locales.

Le modèle Mozilla (suite)

Une fois que les interactions sont possibles. . .

Confidentialité Comme applications locales.

Intégrité Comme applications locales.

Échanges Comme applications locales.

$\Rightarrow$ Avec hack, aucune protection.

Recette générale

1. Se débrouiller pour créer une page qui mélange l'application cible et le code pirate.
2. Trifouiller comme applications locales.

Recette générale

1. Se débrouiller pour créer une page qui mélange l'application cible et le code pirate.
2. Trifouiller comme applications locales.

Cross-Site Scripting (XSS) : 70% des alertes de sécurité, ces jours-ci.

Pour s'en protéger

En résumé :

- Oublier `eval`.
- Vérifier que le seul code JS sur vos pages est bien votre code.
- Clôtures, etc.

Pour s'en protéger

En résumé :

- Oublier `eval`.
- Vérifier que le seul code JS sur vos pages est bien votre code.
- Clôtures, etc.

Plus de détails une autre fois, aujourd'hui, on parle d'extensions.

Le modèle Mozilla

Applications locales

Confidentialité Une application locale peut consulter toutes les informations du système.

Intégrité Une application locale peut agir sur le système sans limites.

Sans limites ?

Une extension suffisamment malicieuse (ou un plug-in) peut :

- accéder au système d'exploitation
- donner accès au système d'exploitation à une application distante
- se faire passer pour une application locale, une page web...

Sans limites ?

Une extension suffisamment malicieuse (ou un plug-in) peut :

- accéder au système d'exploitation
- donner accès au système d'exploitation à une application distante
- se faire passer pour une application locale, une page web...

Applications directes :

Phishing Rediriger un site bancaire vers un site pirate identique.

Sabotage Oublier vos mots de passe.

Vol Envoyer vos mots de passe à un pirate.

Sans limites ?

Une extension suffisamment malicieuse (ou un plug-in) peut :

- accéder au système d'exploitation
- donner accès au système d'exploitation à une application distante
- se faire passer pour une application locale, une page web...

Applications directes :

Phishing	Rediriger un site bancaire vers un site pirate identique.
Sabotage	Oublier vos mots de passe.
Vol	Envoyer vos mots de passe à un pirate.
Sabotage	Supprimer vos fichiers.
Vol	Envoyer vos fichiers à un pirate.

Sans limites ?

Une extension suffisamment malicieuse (ou un plug-in) peut :

- accéder au système d'exploitation
- donner accès au système d'exploitation à une application distante
- se faire passer pour une application locale, une page web...

Applications directes :

Phishing	Rediriger un site bancaire vers un site pirate identique.
Sabotage	Oublier vos mots de passe.
Vol	Envoyer vos mots de passe à un pirate.
Sabotage	Supprimer vos fichiers.
Vol	Envoyer vos fichiers à un pirate.
Escalade	S'emparer d'un mot de passe `sudoer`.

Sans limites ?

Une extension suffisamment malicieuse (ou un plug-in) peut :

- accéder au système d'exploitation
- donner accès au système d'exploitation à une application distante
- se faire passer pour une application locale, une page web...

Applications directes :

Phishing	Rediriger un site bancaire vers un site pirate identique.
Sabotage	Oublier vos mots de passe.
Vol	Envoyer vos mots de passe à un pirate.
Sabotage	Supprimer vos fichiers.
Vol	Envoyer vos fichiers à un pirate.
Escalade	S'emparer d'un mot de passe `sudoer`.

Note C'est *exactement* le syndrome ActiveX.

Le modèle Mozilla

Applications distantes

Confidentialité Une application distante ne peut pas consulter les informations du système.

Intégrité Une application distante ne peut pas agir sur le système.

Le modèle Mozilla

Applications distantes

Confidentialité Une application distante ne peut pas consulter les informations du système.

Intégrité Une application distante ne peut pas agir sur le système.

Limites Une application distante peut presque se faire passer pour une application locale.

Le modèle Mozilla

Applications distantes

Confidentialité Une application distante ne peut pas consulter les informations du système.

Intégrité Une application distante ne peut pas agir sur le système.

Limites Une application distante peut presque se faire passer pour une application locale.

Limites Flash, Java, Quicktime, Google Gears...

Le modèle Mozilla

Applications distantes

Confidentialité Une application distante ne peut pas consulter les informations du système.

Intégrité Une application distante ne peut pas agir sur le système.

Limites Une application distante peut presque se faire passer pour une application locale.

Limites Flash, Java, Quicktime, Google Gears...

et votre extension.

Les limites

Une application locale peut :

- modifier une page web
- échanger des évènements avec des pages web

Les limites

Une application locale peut :

- modifier une page web
- échanger des évènements avec des pages web
- passer des références à une page web

Les limites

Une application locale peut :

- modifier une page web
- échanger des évènements avec des pages web
- passer des références à une page web
- étendre la bibliothèque JavaScript.

Les limites

Une application locale peut :

- modifier une page web
- échanger des évènements avec des pages web
- passer des références à une page web
- étendre la bibliothèque JavaScript.

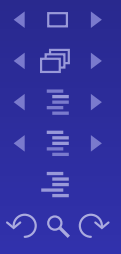
Quelqu'un a pensé à tester la résistance de Google Gears à de l'injection SQL ?

Les limites

Une application locale peut :

- modifier une page web
- échanger des évènements avec des pages web
- passer des références à une page web
- étendre la bibliothèque JavaScript.

Quelqu'un a pensé à tester la résistance de Google Gears à de l'injection SQL ?
Ou à du XSS ?



Les limites

Une application locale peut :

- modifier une page web
- échanger des évènements avec des pages web
- passer des références à une page web
- étendre la bibliothèque JavaScript.

Quelqu'un a pensé à tester la résistance de Google Gears à de l'injection SQL ?
Ou à du XSS ?
Une bibliothèque privilégiée pour *un* site web peut se retrouver utilisée par un autre site malicieux – et ainsi donner accès à tout le système.

Pour se protéger

- Protéger l'application locale comme d'habitude.
- Protéger l'application distante comme d'habitude.

Pour se protéger

- Protéger l'application locale comme d'habitude.
- Protéger l'application distante comme d'habitude.
- Ne pas laisser traîner des références locales.
- Ne jamais avoir confiance en une page web.

Alors ?

Alors ?

Bilan

- Une application locale a tous les droits.
- Une application distante n'a aucun droit dangereux.
- Une application locale malicieuse / boguée peut donner tous les droits à une application distante.

Bilan

- Une application locale a tous les droits.
- Une application distante n'a aucun droit dangereux.
- Une application locale malicieuse / boguée peut donner tous les droits à une application distante.
- On ne peut pas se protéger contre C++.

Solutions techniques

- Protégez votre code.
- Pour une application non-browser, préférez XULRunner à Firefox.
- Pour une application hybride desktop/web, préférez Prism à Firefox.
- Faites attention aux bibliothèques.

Solution communautaire

La vraie raison de la sécurité de Firefox s'appelle AMO (`http://addons.mozilla.org`).

Solution communautaire

La vraie raison de la sécurité de Firefox s'appelle AMO (`http://addons.mozilla.org`).
Quelques critères de rejet :

- Code JS téléchargé dynamiquement.
- Code XUL téléchargé dynamiquement.
- Code binaire téléchargé et installé dynamiquement.
- Mises-à-jour non contrôlées.

Solution communautaire

La vraie raison de la sécurité de Firefox s'appelle AMO (`http://addons.mozilla.org`).
Quelques critères de rejet :

- Code JS téléchargé dynamiquement.
- Code XUL téléchargé dynamiquement.
- Code binaire téléchargé et installé dynamiquement.
- Mises-à-jour non contrôlées.
- . . .ou si une faille est trouvée !

Solution communautaire

La vraie raison de la sécurité de Firefox s'appelle AMO (`http://addons.mozilla.org`).
Quelques critères de rejet :

- Code JS téléchargé dynamiquement.
- Code XUL téléchargé dynamiquement.
- Code binaire téléchargé et installé dynamiquement.
- Mises-à-jour non contrôlées.
- . . .ou si une faille est trouvée !

En pratique, la solution communautaire marche pas mal du tout.

Solution communautaire

La vraie raison de la sécurité de Firefox s'appelle AMO (`http://addons.mozilla.org`).
Quelques critères de rejet :

- Code JS téléchargé dynamiquement.
- Code XUL téléchargé dynamiquement.
- Code binaire téléchargé et installé dynamiquement.
- Mises-à-jour non contrôlées.
- . . .ou si une faille est trouvée !

En pratique, la solution communautaire marche pas mal du tout.
⇒ gardez votre code simple si vous voulez qu'il soit accepté !

L'avenir de JavaScript

JavaScript 2 arrive.

L'avenir de JavaScript

JavaScript 2 arrive.

- Les modules entrent en force.
- Exit `eval` (ou presque).
- Typage statique + programmation défensive ?
- Un typage dynamique plus puissant.
- Valeurs privées/protégées contre les modifications.
- Meilleure gestion des erreurs.

Alternatives

D'autres techniques sont possibles :

Restriction de privilèges

Séparation des extensions

Analyse statique

Compilateurs sûrs

Alternatives

D'autres techniques sont possibles :

Restriction de privilèges cf. `nsScriptSecurityManager`

Séparation des extensions

Analyse statique

Compilateurs sûrs

Alternatives

D'autres techniques sont possibles :

Restriction de privilèges cf. `nsScriptSecurityManager`

Séparation des extensions mais ça va tout casser

Analyse statique

Compilateurs sûrs

Alternatives

D'autres techniques sont possibles :

Restriction de privilèges cf. `nsScriptSecurityManager`

Séparation des extensions mais ça va tout casser

Analyse statique typage/interprétation abstraite

Compilateurs sûrs

Alternatives

D'autres techniques sont possibles :

Restriction de privilèges cf. `nsScriptSecurityManager`

Séparation des extensions mais ça va tout casser

Analyse statique typage/interprétation abstraite

Compilateurs sûrs Caja, etc.

Alternatives

D'autres techniques sont possibles :

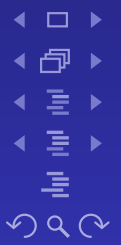
Restriction de privilèges cf. `nsScriptSecurityManager`

Séparation des extensions mais ça va tout casser

Analyse statique typage/interprétation abstraite

Compilateurs sûrs Caja, etc.

Toutes ces techniques nécessitent de rejeter les extensions qui contiennent du code binaire.



Derniers mots

- C'est bon, je vous ai fait peur ?

Derniers mots

- C'est bon, je vous ai fait peur ?
- Stagiaires wanted !

Merci pour votre attention

Des questions ?