Programmation orientée sécurité : introduction

Olivier Levillain

4INFOS8ProSec @ INSA 2018-2019

Avant-propos

- Les supports de cours seront disponibles sur `http://paperstreet.picty.org/POS/`
- En cas de question, n'hésitez-pas
  - pendant le cours, à m'interrompre,
  - plus tard, à m'envoyer un courrier électronique à `cours-POS@picty.org`

Introduction

Qui suis-je ?

Parcours

- stage de DEA en cryptographie sur une fonction de hachage
- membre du laboratoire « système » de l'ANSSI (2007-2012)
- responsable du laboratoire « réseau » de l'ANSSI (2012-2015)
- responsable du centre de formation de l'ANSSI (2015-2018)
- maître de conférences à Télécom SudParis (2018-)

Qui suis-je ?

Parcours

- stage de DEA en cryptographie sur une fonction de hachage
- membre du laboratoire « système » de l'ANSSI (2007-2012)
- responsable du laboratoire « réseau » de l'ANSSI (2012-2015)
- responsable du centre de formation de l'ANSSI (2015-2018)
- maître de conférences à Télécom SudParis (2018-)

Recherche

- participation aux travaux sur les mécanismes bas-niveau x86
- travaux sur SSL/TLS
- études sur les langages
- travaux sur les *parsers*

Description du cours

Objectifs

- Comprendre les enjeux de la programmation sécurisée
- Décrire les vulnérabilités classiques
- Comprendre comment s'en protéger
- Adopter une attitude proactive et responsable

Demandez le programme !

Six matinées (dont une allongée) :

- 27/02 : introduction, vulnérabilités classiques
- 28/02 : autres vulnérabilités et bonnes pratiques
- 13/03 : MyL et *buffer overflow*
- 14/03 : TP noté
- 27/03 : web / crypto, correction du TP noté
- 28/03 : soutenances projets

Demandez le programme !

Six matinées (dont une allongée) :

- 27/02 : introduction, vulnérabilités classiques
- 28/02 : autres vulnérabilités et bonnes pratiques
- 13/03 : MyL et *buffer overflow*
- 14/03 : TP noté
- 27/03 : web / crypto, correction du TP noté
- 28/03 : soutenances projets

Politique d'évaluation du module :

- un TP noté (le 14 mars)
- un projet bibliographique en binôme (soutenances en mars)
- un contrôle écrit (en avril ?)

Quelques questions

- Quels langages de programmation connaissez-vous ?
- Quel est votre plus gros projet de programmation ?
- Quels sont vos projets pour le stage à venir ?

Projets bibliographiques (1/2)

Objectifs

- chercher de l'information sur une faille
- présenter votre démarche de recherche
- expliquer la vulnérabilité et son impact
- analyser les réponses à apporter (court et long terme)

Modalités

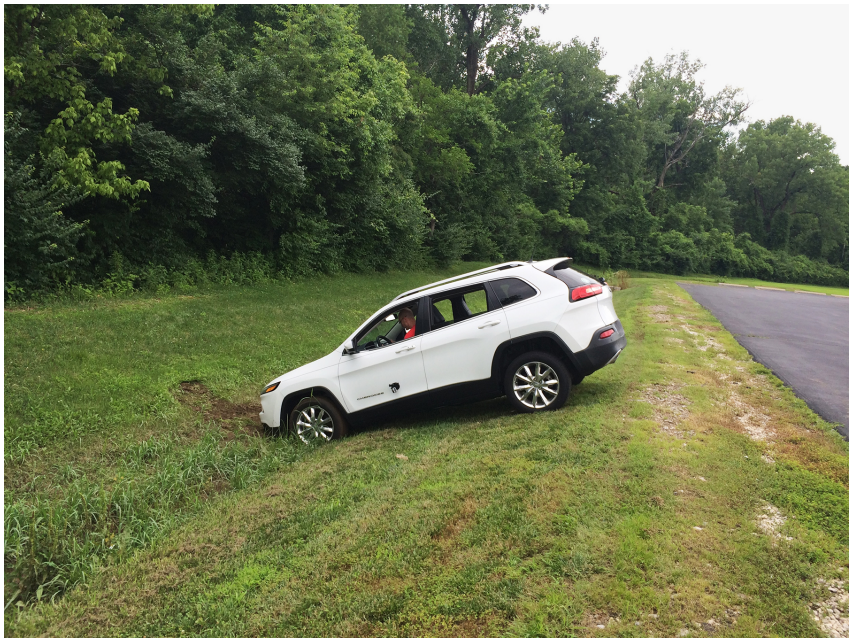
- travail en binôme
- préparer une présentation d'ici le 15 mars (20 + 15 min)
- exemple de présentation à venir

Projets bibliographiques (2/2)

Sujets

- accès non autorisé dans OpenSSH
- biais RC4
- exécution de code arbitraire dans `tnftp`
- faille Java « *Calendar* »
- Déréférencement de pointeur nul dans Linux
- Oracle de *padding* dans SSH
- SKIP-TLS
- BERserk
- *Packets in packets*

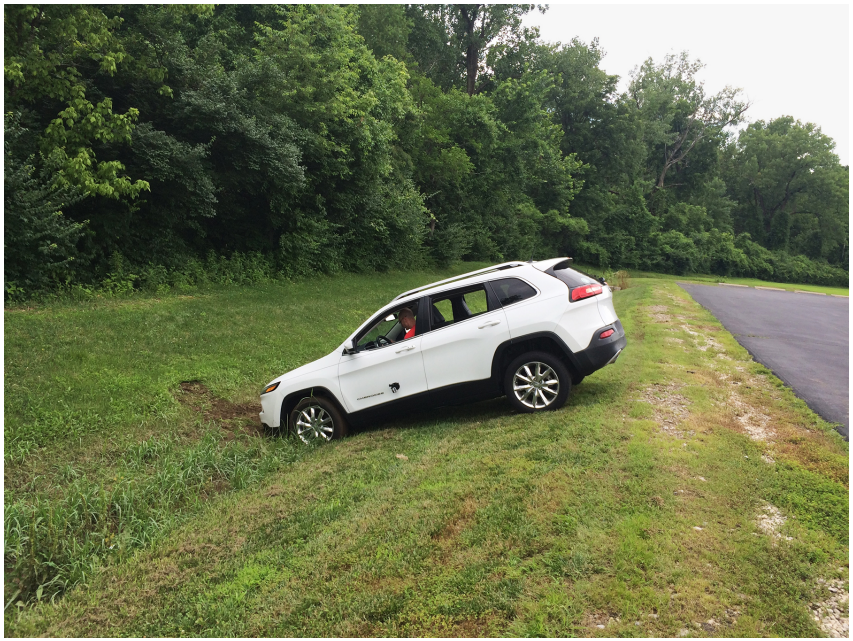
Une voiture connectée



Charlie Miller et Chris Valasek (BlackHat2015) : prise de contrôle à distance d'une Jeep

- Cause : plein de services non sécurisés en écoute sur internet

Une voiture connectée



Charlie Miller et Chris Valasek (BlackHat2015) : prise de contrôle à distance d'une Jeep

- Cause : plein de services non sécurisés en écoute sur internet
- Combien de voitures (avions, usines...) reposent de manière critique sur du logiciel pour fonctionner ?

Heartbleed

Heartbleed (8 avril 2014)

- Fuite de données HTTP accessibles à tous de manière discrète
- Une faille très médiatisée
- Cause : un débordement de tampon dans OpenSSL...

Heartbleed

Heartbleed (8 avril 2014)

- Fuite de données HTTP accessibles à tous de manière discrète
- Une faille très médiatisée
- Cause : un débordement de tampon dans OpenSSL...
- dans une fonctionnalité inutilisée (mais activée par défaut)...

Heartbleed

Heartbleed (8 avril 2014)

- Fuite de données HTTP accessibles à tous de manière discrète
- Une faille très médiatisée
- Cause : un débordement de tampon dans OpenSSL...
- dans une fonctionnalité inutilisée (mais activée par défaut)...
- qui permet de récupérer des bouts de la mémoire du serveur

Heartbleed

Heartbleed (8 avril 2014)

- Fuite de données HTTP accessibles à tous de manière discrète
- Une faille très médiatisée
- Cause : un débordement de tampon dans OpenSSL...
- dans une fonctionnalité inutilisée (mais activée par défaut)...
- qui permet de récupérer des bouts de la mémoire du serveur
- Connaissez-vous l'autre information SSI du 8 avril 2014 ?

Stuxnet

(Source : Office of the Presidency of the Islamic Republic of Iran)

Attaque de systèmes industriels (centrifugeuses) en Iran (2010)

- Manipulation du logiciel des automates pour saboter les équipements...
- tout en remontant des informations fausses à la supervision
- Cause : des logiciels propriétaires sans aucune défense (limites de la sécurité par l'obscurité)

Stuxnet

(Source : Office of the Presidency of the Islamic Republic of Iran)

Attaque de systèmes industriels (centrifugeuses) en Iran (2010)

- Manipulation du logiciel des automates pour saboter les équipements...
- tout en remontant des informations fausses à la supervision
- Cause : des logiciels propriétaires sans aucune défense (limites de la sécurité par l'obscurité)
- Les installations en question étaient sur un réseau dédié (limites de l'*air gap*)

Santé et sécurité

Les *pacemakers* ont aujourd'hui des interfaces sans fil pour permettre un suivi en temps réel des patients

- Dick Cheney (vice-président des USA sous Georges W. Bush) a été convaincu par la NSA de désactiver ces interfaces
- Quelques publications académiques sur le sujet

Santé et sécurité

Les *pacemakers* ont aujourd'hui des interfaces sans fil pour permettre un suivi en temps réel des patients

- Dick Cheney (vice-président des USA sous Georges W. Bush) a été convaincu par la NSA de désactiver ces interfaces
- Quelques publications académiques sur le sujet
- Scénario utilisé dans une série américaine

Santé et sécurité

Les *pacemakers* ont aujourd'hui des interfaces sans fil pour permettre un suivi en temps réel des patients

- Dick Cheney (vice-président des USA sous Georges W. Bush) a été convaincu par la NSA de désactiver ces interfaces
- Quelques publications académiques sur le sujet
- Scénario utilisé dans une série américaine

Erreurs logicielles et santé

- Erreur de calcul dans les doses de radiation à Epinal

Les caméras à l'assaut d'OVH

En septembre 2016, un réseau de caméras infectées par un logiciel malveillant (un *botnet*) a servi à monter plusieurs attaques DDoS

- DDoS = *Distributed Denial of Service*
- La bande passante reçue par OVH a été estimée à 1 Tbps

Les caméras à l'assaut d'OVH

En septembre 2016, un réseau de caméras infectées par un logiciel malveillant (un *botnet*) a servi à monter plusieurs attaques DDoS

- DDoS = *Distributed Denial of Service*
- La bande passante reçue par OVH a été estimée à 1 Tbps

Les causes de l'attaque

- de nombreux équipements connectés avec des mots de passe par défaut
- les caméras ont souvent une bonne connectivité

Les caméras à l'assaut d'OVH

En septembre 2016, un réseau de caméras infectées par un logiciel malveillant (un *botnet*) a servi à monter plusieurs attaques DDoS

- DDoS = *Distributed Denial of Service*
- La bande passante reçue par OVH a été estimée à 1 Tbps

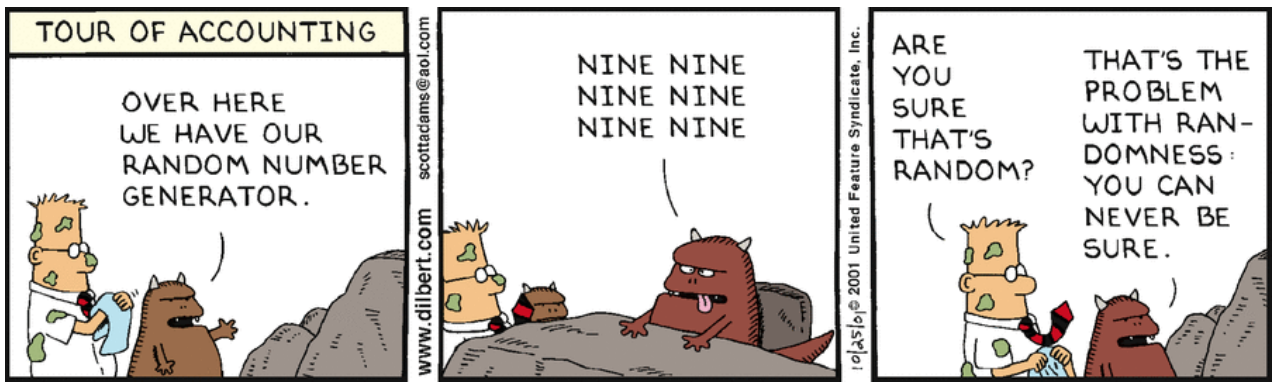
Les causes de l'attaque

- de nombreux équipements connectés avec des mots de passe par défaut
- les caméras ont souvent une bonne connectivité

Pour aller plus loin

- un précédent intéressant : l'Internet Census 2012...
- au fait, qui est responsable ?

Le problème avec l'aléa... (1/2)



(Source : Dilbert Comic Strip by Scott Adams)

Le problème avec l'aléa... (2/2)

Erreurs dans les implémentations de générateurs aléatoires

- 2009 : *bug* dans OpenSSL Debian
- Usenix Security 2012 : *Mining Your Ps and Qs : Detection of Widespread Weak Keys in Network Devices* par Heninger et al. (`factorable.net`)
- Clés cryptographiques prédictibles ou cassables
- Cause : incompréhension des hypothèses cryptographiques par les développeurs/intégrateurs

Le logiciel est littéralement *partout*

- Les terminaux de paiement intègrent aujourd'hui des piles logicielles *complètes* qui vont jusqu'au lecteur Flash pour la publicité !

Le logiciel est littéralement *partout*

- Les terminaux de paiement intègrent aujourd'hui des piles logicielles *complètes* qui vont jusqu'au lecteur Flash pour la publicité !
  - la complexité est l'ennemi de la sécurité

Le logiciel est littéralement *partout*

- Les terminaux de paiement intègrent aujourd'hui des piles logicielles *complètes* qui vont jusqu'au lecteur Flash pour la publicité !
  - la complexité est l'ennemi de la sécurité
- *Internet of Things* : les frigos et ampoules connectées !

Le logiciel est littéralement *partout*

- Les terminaux de paiement intègrent aujourd'hui des piles logicielles *complètes* qui vont jusqu'au lecteur Flash pour la publicité !
  - la complexité est l'ennemi de la sécurité
- *Internet of Things* : les frigos et ampoules connectées !
  - brève insolite il y a quelques temps : un grille-pain *spammeur*

Le logiciel est littéralement *partout*

- Les terminaux de paiement intègrent aujourd'hui des piles logicielles *complètes* qui vont jusqu'au lecteur Flash pour la publicité !
  - la complexité est l'ennemi de la sécurité
- *Internet of Things* : les frigos et ampoules connectées !
  - brève insolite il y a quelques temps : un grille-pain *spammeur*
- Les télévisions connectées, un nouveau point d'entrée
  - Thèse d'Yann Bachy (LAAS, Toulouse)

Sûreté de fonctionnement et sécurité

Dans certains secteurs existe la notion de *sûreté de fonctionnement* :

- recherche de la fiabilité
- prise en compte des pannes
- mesures classiques : redondance des équipements, restriction du périmètre logiciel, tests, preuves et autres méthodes formelles

Sûreté de fonctionnement et sécurité

Dans certains secteurs existe la notion de *sûreté de fonctionnement* :

- recherche de la fiabilité
- prise en compte des pannes
- mesures classiques : redondance des équipements, restriction du périmètre logiciel, tests, preuves et autres méthodes formelles

Les propriétés classiques de la sécurité :

- confidentialité
- intégrité
- disponibilité

Sûreté de fonctionnement et sécurité

Dans certains secteurs existe la notion de *sûreté de fonctionnement* :

- recherche de la fiabilité
- prise en compte des pannes
- mesures classiques : redondance des équipements, restriction du périmètre logiciel, tests, preuves et autres méthodes formelles

Les propriétés classiques de la sécurité :

- confidentialité
- intégrité
- disponibilité

Les outils de la sûreté peuvent être utilisés ici aussi, mais ne suffisent généralement pas

Différence entre sûreté et sécurité

Dans les deux cas, l'objectif passe par

- une meilleure maîtrise du code
- une meilleure qualité logicielle

Cependant, une différence fondamentale

- la sûreté s'intéresse aux *pannes* et considère la *probabilité* de certains événements
- la sécurité prend en compte un *attaquant* qui s'adapte (le *coût* devient alors un indicateur plus pertinent)

Sécurité ? (1/3)

Parmi les commandes suivantes, lesquelles sont susceptibles (sans redirection) de provoquer la destruction des données d'un fichier ?

☐ `ls` ☐ `cd` ☐ `cp` ☐ `cat` ☐ `rm` ☐ `mv`

Sécurité ? (1/3)

Parmi les commandes suivantes, lesquelles sont susceptibles (sans redirection) de provoquer la destruction des données d'un fichier ?

☐ `ls` ☐ `cd` ☐ `cp` ☐ `cat` ☐ `rm` ☐ `mv`

Fonctionnel

- Question ré-interprétée « *Comment détruire les données d'un fichier ?* », seule la commande `rm` est mentionnée

Sécurité ? (1/3)

Parmi les commandes suivantes, lesquelles sont susceptibles (sans redirection) de provoquer la destruction des données d'un fichier ?

☐ `ls` ☐ `cd` ☐ `cp` ☐ `cat` ☐ `rm` ☐ `mv`

Fonctionnel

- Question ré-interprétée « *Comment détruire les données d'un fichier ?* », seule la commande `rm` est mentionnée

Sécurité

- Si on cherche à protéger les données, les commandes dangereuses sont `rm` mais aussi `cp` et `mv`, qui écrasent les fichiers cibles

Sécurité ? (2/3)

Un paquet IP comporte dans son entête deux champs adresse pour sa source et sa destination

Fonctionnel

- Rendre le service consiste à acheminer l'information jusqu'à l'adresse destination
- La diffusion convient dans certains contextes

Sécurité ? (2/3)

Un paquet IP comporte dans son entête deux champs adresse pour sa source et sa destination

Fonctionnel

- Rendre le service consiste à acheminer l'information jusqu'à l'adresse destination
- La diffusion convient dans certains contextes

Sécurité

- L'adresse source n'est probablement ni exploitée ni vérifiée lors du transit, ce qui permet le *spoofing* d'adresses IP
- En diffusion, chaque récepteur voit tous les paquets

Sécurité ? (3/3)

Spécification de deux fonctions pour la compression (`Zip`) et la décompression (`Unzip`) de fichiers

Sécurité ? (3/3)

Spécification de deux fonctions pour la compression (`Zip`) et la décompression (`Unzip`) de fichiers

Fonctionnel

- $\forall\,(f : \mathsf{File}),\ \mathsf{Unzip}(\mathsf{Zip}\,f) = f$

Sécurité ? (3/3)

Spécification de deux fonctions pour la compression (`Zip`) et la décompression (`Unzip`) de fichiers

Fonctionnel

- $\forall\,(f : \mathsf{File}),\ \mathsf{Unzip}(\mathsf{Zip}\,f) = f$

Sécurité

- $\forall\,(c : \mathsf{File}),\ (\neg\exists\,(f : \mathsf{File}),\ \mathsf{Zip}\,f = c) \Rightarrow \mathsf{Unzip}\,c = \mathsf{Error}$
- En particulier, ne pas faire confiance à un champ annonçant à l'avance la taille du fichier décompressé

L'esprit de sécurité

Comment résumer cet état d'esprit nécessaire pour « penser sécurité » ?

L'esprit de sécurité

Comment résumer cet état d'esprit nécessaire pour « penser sécurité » ?

- il faut penser *au-delà du fonctionnel*

L'esprit de sécurité

Comment résumer cet état d'esprit nécessaire pour « penser sécurité » ?

- il faut penser *au-delà du fonctionnel*
- pour défendre, il faut couvrir tous les chemins d'attaques possibles de manière cohérente (principe du maillon faible)

L'esprit de sécurité

Comment résumer cet état d'esprit nécessaire pour « penser sécurité » ?

- il faut penser *au-delà du fonctionnel*
- pour défendre, il faut couvrir tous les chemins d'attaques possibles de manière cohérente (principe du maillon faible)
- aucune défense n'est parfaite, il faut combiner des mécanismes pour créer des lignes de défense multiples et complémentaires (principe de la défense en profondeur)

L'esprit de sécurité

Comment résumer cet état d'esprit nécessaire pour « penser sécurité » ?

- il faut penser *au-delà du fonctionnel*
- pour défendre, il faut couvrir tous les chemins d'attaques possibles de manière cohérente (principe du maillon faible)
- aucune défense n'est parfaite, il faut combiner des mécanismes pour créer des lignes de défense multiples et complémentaires (principe de la défense en profondeur)
- de manière générale, en sécurité, on se pose beaucoup de questions...

L'esprit de sécurité

Comment résumer cet état d'esprit nécessaire pour « penser sécurité » ?

- il faut penser *au-delà du fonctionnel*
- pour défendre, il faut couvrir tous les chemins d'attaques possibles de manière cohérente (principe du maillon faible)
- aucune défense n'est parfaite, il faut combiner des mécanismes pour créer des lignes de défense multiples et complémentaires (principe de la défense en profondeur)
- de manière générale, en sécurité, on se pose beaucoup de questions...
- sans pouvoir apporter de réponses génériques valables à tous les coups (importance du contexte)

L'esprit de sécurité

Comment résumer cet état d'esprit nécessaire pour « penser sécurité » ?

- il faut penser *au-delà du fonctionnel*
- pour défendre, il faut couvrir tous les chemins d'attaques possibles de manière cohérente (principe du maillon faible)
- aucune défense n'est parfaite, il faut combiner des mécanismes pour créer des lignes de défense multiples et complémentaires (principe de la défense en profondeur)
- de manière générale, en sécurité, on se pose beaucoup de questions...
- sans pouvoir apporter de réponses génériques valables à tous les coups (importance du contexte)

Attention cependant aux conséquences néfastes de cet état d'esprit poussé à l'extrême : paranoïa et immobilisme.

Les dangers d'un logiciel non sûr

An unreliable programming language generating unreliable programs constitutes a far greater risk to our environment and to our society than unsafe cars, toxic pesticides, or accidents at nuclear power stations. Be vigilant to reduce that risk, not to increase it.

C.A.R Hoare

Les dangers d'un logiciel non sûr

An unreliable programming language generating unreliable programs constitutes a far greater risk to our environment and to our society than unsafe cars, toxic pesticides, or accidents at nuclear power stations. Be vigilant to reduce that risk, not to increase it.

C.A.R Hoare (1980)

Les dangers d'un logiciel non sûr

> *An unreliable programming language generating unreliable programs constitutes a far greater risk to our environment and to our society than unsafe cars, toxic pesticides, or accidents at nuclear power stations. Be vigilant to reduce that risk, not to increase it.*
>
> *C.A.R Hoare (1980)*

Notes

- Dans ce discours (prix Türing), Hoare pensait sans doute plus à la sûreté de fonctionnement qu'à la sécurité

Les dangers d'un logiciel non sûr

> *An unreliable programming language generating unreliable programs constitutes a far greater risk to our environment and to our society than unsafe cars, toxic pesticides, or accidents at nuclear power stations. Be vigilant to reduce that risk, not to increase it.*
>
> *C.A.R Hoare (1980)*

Notes

- Dans ce discours (prix Türing), Hoare pensait sans doute plus à la sûreté de fonctionnement qu'à la sécurité
- À l'époque, les voitures et centrales nucléaires dépendait *nettement moins* du logiciel

Les dangers d'un logiciel non sûr

> *An unreliable programming language generating unreliable programs constitutes a far greater risk to our environment and to our society than unsafe cars, toxic pesticides, or accidents at nuclear power stations. Be vigilant to reduce that risk, not to increase it.*
>
> *C.A.R Hoare (1980)*

Notes

- Dans ce discours (prix Türing), Hoare pensait sans doute plus à la sûreté de fonctionnement qu'à la sécurité
- À l'époque, les voitures et centrales nucléaires dépendait *nettement moins* du logiciel
- Le langage décrié par Hoare était... Ada

Conclusion de cette (longue) introduction

- Le logiciel est *partout*
- De nombreux processus critiques en dépendent dans différents domaines
  - santé
  - transport
  - énergie
  - militaire
  - économique

Conclusion de cette (longue) introduction

- Le logiciel est *partout*
- De nombreux processus critiques en dépendent dans différents domaines
  - santé
  - transport
  - énergie
  - militaire
  - économique
- Les développeurs de demain ne peuvent ignorer la sécurité
  - il faut penser au-delà du fonctionnel
  - *simple is beautiful*
  - défendre en profondeur

Questions

?