

Techniques d'attaques

Une introduction aux problèmes de sécurité, aux exploits et à leur prévention

Luca SAIU

<http://ageinghacker.net>

Département Réseaux et Télécommunications
IUT de Villetaneuse, Université Paris 13

Mai 2019

Sommaire

- 1 Introduction
 - Sécurité
 - Sources de failles
 - Méthodologie des attaquants
 - Recherche des informations
- 2 Techniques d'attaque et de défense
 - Attaque
 - Défense
- 3 Annexes
 - Remerciements
 - Autres ressources

Les quatre propriétés de sécurité

- *Disponibilité* : un service doit toujours être accessible quand les utilisateurs souhaitent y accéder
- *Intégrité* : les informations manipulées par un service doivent être conservées sous leur forme originelle. Elles ne doivent être modifiées que par des tiers licites
- *Confidentialité* : les informations doivent rester inaccessibles aux yeux des tiers non autorisés
- *Imputabilité/Non répudiation* : prévenir le refus, le démenti qu'un message ait été émis ou reçu, ou qu'une action/transaction ait eu lieu

Les quatre propriétés de sécurité

- *Disponibilité* : un service doit toujours être accessible quand les utilisateurs souhaitent y accéder
- *Intégrité* : les informations manipulées par un service doivent être conservées sous leur forme originelle. Elles ne doivent être modifiées que par des tiers licites
- *Confidentialité* : les informations doivent rester inaccessibles aux yeux des tiers non autorisés
- *Imputabilité/Non répudiation* : prévenir le refus, le démenti qu'un message ait été émis ou reçu, ou qu'une action/transaction ait eu lieu

Les quatre propriétés de sécurité

- *Disponibilité* : un service doit toujours être accessible quand les utilisateurs souhaitent y accéder
- *Intégrité* : les informations manipulées par un service doivent être conservées sous leur forme originelle. Elles ne doivent être modifiées que par des tiers licites
- *Confidentialité* : les informations doivent rester inaccessibles aux yeux des tiers non autorisés
- *Imputabilité/Non répudiation* : prévenir le refus, le démenti qu'un message ait été émis ou reçu, ou qu'une action/transaction ait eu lieu

Les quatre propriétés de sécurité

- *Disponibilité* : un service doit toujours être accessible quand les utilisateurs souhaitent y accéder
- *Intégrité* : les informations manipulées par un service doivent être conservées sous leur forme originelle. Elles ne doivent être modifiées que par des tiers licites
- *Confidentialité* : les informations doivent rester inaccessibles aux yeux des tiers non autorisés
- *Imputabilité/Non répudiation* : prévenir le refus, le démenti qu'un message ait été émis ou reçu, ou qu'une action/transaction ait eu lieu

Sources de failles

- *Humain*
- *Topologie du réseau*
- *Services*
 - système
 - réseau
- *Mauvaises configurations*
- *Sécurité physique*
- *Système d'exploitation*

Sources de failles

- *Humain*
- *Topologie du réseau*
- *Services*
 - système
 - réseau
- *Mauvaises configurations*
- *Sécurité physique*
- *Système d'exploitation*

Sources de failles

- *Humain*
- *Topologie du réseau*
- *Services*
 - système
 - réseau
- *Mauvaises configurations*
- *Sécurité physique*
- *Système d'exploitation*

Sources de failles

- *Humain*
- *Topologie du réseau*
- *Services*
 - système
 - réseau
- *Mauvaises configurations*
- *Sécurité physique*
- *Système d'exploitation*

Sources de failles

- *Humain*
- *Topologie du réseau*
- *Services*
 - système
 - réseau
- *Mauvaises configurations*
- *Sécurité physique*
- *Système d'exploitation*

Sources de failles

- *Humain*
- *Topologie du réseau*
- *Services*
 - système
 - réseau
- *Mauvaises configurations*
- *Sécurité physique*
- *Système d'exploitation*

Méthodologie des attaquants

Quatre phases :

- *Recherche d'informations*
- *Masquer / se masquer*
- *Obtenir un accès/escalader les privilèges*
- *Paralyser une machine*

Méthodologie des attaquants

Quatre phases :

- *Recherche d'informations*
- *Masquer / se masquer*
- *Obtenir un accès/escalader les privilèges*
- *Paralyser une machine*

Méthodologie des attaquants

Quatre phases :

- *Recherche d'informations*
- *Masquer / se masquer*
- *Obtenir un accès/escalader les privilèges*
- *Paralyser une machine*

Méthodologie des attaquants

Quatre phases :

- *Recherche d'informations*
- *Masquer / se masquer*
- *Obtenir un accès/escalader les privilèges*
- *Paralyser une machine*

Recherche des informations

- *Footprinting* : on cherche à recenser toute machine / adresse IP de la cible
- *Scanning* : déterminer les services réseaux hébergés sur la cible
- *Fingerprinting* : déterminer le système d'exploitation et les logiciels spécifiques
- *Sniffing* : récupérer les données transitant sur le réseau
- *Social engineering* : contacter une personne par téléphone, mail, messagerie afin d'obtenir des informations

Recherche des informations

- *Footprinting* : on cherche à recenser toute machine / adresse IP de la cible
- *Scanning* : déterminer les services réseaux hébergés sur la cible
- *Fingerprinting* : déterminer le système d'exploitation et les logiciels spécifiques
- *Sniffing* : récupérer les données transitant sur le réseau
- *Social engineering* : contacter une personne par téléphone, mail, messagerie afin d'obtenir des informations

Recherche des informations

- *Footprinting* : on cherche à recenser toute machine / adresse IP de la cible
- *Scanning* : déterminer les services réseaux hébergés sur la cible
- *Fingerprinting* : déterminer le système d'exploitation et les logiciels spécifiques
- *Sniffing* : récupérer les données transitant sur le réseau
- *Social engineering* : contacter une personne par téléphone, mail, messagerie afin d'obtenir des informations

Recherche des informations

- *Footprinting* : on cherche à recenser toute machine / adresse IP de la cible
- *Scanning* : déterminer les services réseaux hébergés sur la cible
- *Fingerprinting* : déterminer le système d'exploitation et les logiciels spécifiques
- *Sniffing* : récupérer les données transitant sur le réseau
- *Social engineering* : contacter une personne par téléphone, mail, messagerie afin d'obtenir des informations

Recherche des informations

- *Footprinting* : on cherche à recenser toute machine / adresse IP de la cible
- *Scanning* : déterminer les services réseaux hébergés sur la cible
- *Fingerprinting* : déterminer le système d'exploitation et les logiciels spécifiques
- *Sniffing* : récupérer les données transitant sur le réseau
- *Social engineering* : contacter une personne par téléphone, mail, messagerie afin d'obtenir des informations

Exploits

Profiter d'un *bug* dans du logiciel (ou matériel) pour obtenir un comportement non prévu par le développeur et une *privilege escalation* ou *denial of service*.

- *local* exploit
- *remote* exploit
- *injection de code*
- un *buffer overflow* est souvent un *vecteur*
- Autres vecteurs : *double free*, *use-after-free*
- *Mauvaise configuration*

Prévention : techniques de programmation sûre, isolation.
...Mais ce ne suffit pas.

Exploits

Profiter d'un *bug* dans du logiciel (ou matériel) pour obtenir un comportement non prévu par le développeur et une *privilege escalation* ou *denial of service*.

- *local* exploit
- *remote* exploit
- *injection de code*
- un *buffer overflow* est souvent un *vecteur*
- Autres vecteurs : *double free*, *use-after-free*
- *Mauvaise configuration*

Prévention : techniques de programmation sûre, isolation.
...Mais ce ne suffit pas.

Exploits

Profiter d'un *bug* dans du logiciel (ou matériel) pour obtenir un comportement non prévu par le développeur et une *privilege escalation* ou *denial of service*.

- *local* exploit
- *remote* exploit
- *injection de code*
 - un *buffer overflow* est souvent un *vecteur*
 - Autres vecteurs : *double free*, *use-after-free*
 - *Mauvaise configuration*

Prévention : techniques de programmation sûre, isolation.
...Mais ce ne suffit pas.

Exploits

Profiter d'un *bug* dans du logiciel (ou matériel) pour obtenir un comportement non prévu par le développeur et une *privilege escalation* ou *denial of service*.

- *local* exploit
- *remote* exploit
- *injection de code*
- un *buffer overflow* est souvent un *vecteur*
 - Autres vecteurs : *double free*, *use-after-free*
 - *Mauvaise configuration*

Prévention : techniques de programmation sûre, isolation.
...Mais ce ne suffit pas.

Exploits

Profiter d'un *bug* dans du logiciel (ou matériel) pour obtenir un comportement non prévu par le développeur et une *privilege escalation* ou *denial of service*.

- *local* exploit
- *remote* exploit
- *injection de code*
- un *buffer overflow* est souvent un *vecteur*
- Autres vecteurs : *double free*, *use-after-free*
- *Mauvaise configuration*

Prévention : techniques de programmation sûre, isolation.
...Mais ce ne suffit pas.

Exploits

Profiter d'un *bug* dans du logiciel (ou matériel) pour obtenir un comportement non prévu par le développeur et une *privilege escalation* ou *denial of service*.

- *local* exploit
- *remote* exploit
- *injection de code*
- un *buffer overflow* est souvent un *vecteur*
- Autres vecteurs : *double free*, *use-after-free*
- *Mauvaise configuration*

Prévention : techniques de programmation sûre, isolation.
...Mais ce ne suffit pas.

Exploits

Profiter d'un *bug* dans du logiciel (ou matériel) pour obtenir un comportement non prévu par le développeur et une *privilege escalation* ou *denial of service*.

- *local* exploit
- *remote* exploit
- *injection de code*
- un *buffer overflow* est souvent un *vecteur*
- Autres vecteurs : *double free*, *use-after-free*
- *Mauvaise configuration*

Prévention : **techniques de programmation sûre, isolation.**

...Mais ce ne suffit pas.

Exploits

Profiter d'un *bug* dans du logiciel (ou matériel) pour obtenir un comportement non prévu par le développeur et une *privilege escalation* ou *denial of service*.

- *local* exploit
- *remote* exploit
- *injection de code*
- un *buffer overflow* est souvent un *vecteur*
- Autres vecteurs : *double free*, *use-after-free*
- *Mauvaise configuration*

Prévention : techniques de programmation sûre, isolation.
...**Mais ce ne suffit pas.**

Autres types de vulnérabilités et d'attaques

- *brute-force* attacks
- *backdoor* (une méthode « secrète » de contourner l'authentification)
- *Denial of Service (DoS)* (Distributed DoS ou DDoS, quand on prend le contrôle d'*autres machines qu'on a déjà compromis* pour lancer des DoS en parallèle)
- *eavesdropping* (analyse de paquets si vous avez accès au réseau physique ; les programmes des services secrets américains Carnivore, NarusInSight)
- *privilege escalation*
- *spoofing* Ex : MAC spoofing, IP address spoofing, attaques au DNS
- *social engineering*
- *phishing* (une forme d'ingénierie sociale)
- *direct-access attacks* (ex. : accès aux disques, installation de keyloggers)

Autres types de vulnérabilités et d'attaques

- *brute-force* attacks
- *backdoor* (une méthode « secrète » de contourner l'authentification)
- *Denial of Service (DoS)* (Distributed DoS ou DDoS, quand on prend le contrôle d'autres machines qu'on a déjà compromis pour lancer des DoS en parallèle)
- *eavesdropping* (analyse de paquets si vous avez accès au réseau physique ; les programmes des services secrets américains Carnivore, NarusInSight)
- *privilege escalation*
- *spoofing* Ex : MAC spoofing, IP address spoofing, attaques au DNS
- *social engineering*
- *phishing* (une forme d'ingénierie sociale)
- *direct-access attacks* (ex. : accès aux disques, installation de keyloggers)

Autres types de vulnérabilités et d'attaques

- *brute-force* attacks
- *backdoor* (une méthode « secrète » de contourner l'authentification)
- *Denial of Service (DoS)* (Distributed DoS ou DDoS, quand on prend le contrôle d'*autres machines qu'on a déjà compromis* pour lancer des DoS en parallèle)
- *eavesdropping* (analyse de paquets si vous avez accès au réseau physique ; les programmes des services secrets américains Carnivore, NarusInSight)
- *privilege escalation*
- *spoofing* Ex : MAC spoofing, IP address spoofing, attaques au DNS
- *social engineering*
- *phishing* (une forme d'ingénierie sociale)
- *direct-access attacks* (ex. : accès aux disques, installation de keyloggers)

Autres types de vulnérabilités et d'attaques

- *brute-force* attacks
- *backdoor* (une méthode « secrète » de contourner l'authentification)
- *Denial of Service (DoS)* (Distributed DoS ou DDoS, quand on prend le contrôle d'*autres machines qu'on a déjà compromis* pour lancer des DoS en parallèle)
- *eavesdropping* (analyse de paquets si vous avez accès au réseau physique ; les programmes des services secrets américains Carnivore, NarusInSight)
- *privilege escalation*
- *spoofing* Ex : MAC spoofing, IP address spoofing, attaques au DNS
- *social engineering*
- *phishing* (une forme d'ingénierie sociale)
- *direct-access attacks* (ex. : accès aux disques, installation de keyloggers)

Autres types de vulnérabilités et d'attaques

- *brute-force* attacks
- *backdoor* (une méthode « secrète » de contourner l'authentification)
- *Denial of Service (DoS)* (Distributed DoS ou DDoS, quand on prend le contrôle d'*autres machines qu'on a déjà compromis* pour lancer des DoS en parallèle)
- *eavesdropping* (analyse de paquets si vous avez accès au réseau physique ; les programmes des services secrets américains Carnivore, NarusInSight)
- *privilege escalation*
 - *spoofing* Ex : MAC spoofing, IP address spoofing, attaques au DNS
 - *social engineering*
 - *phishing* (une forme d'ingénierie sociale)
 - *direct-access attacks* (ex. : accès aux disques, installation de keyloggers)

Autres types de vulnérabilités et d'attaques

- *brute-force* attacks
- *backdoor* (une méthode « secrète » de contourner l'authentification)
- *Denial of Service (DoS)* (Distributed DoS ou DDoS, quand on prend le contrôle d'*autres machines qu'on a déjà compromis* pour lancer des DoS en parallèle)
- *eavesdropping* (analyse de paquets si vous avez accès au réseau physique ; les programmes des services secrets américains Carnivore, NarusInSight)
- *privilege escalation*
- *spoofing* Ex : MAC spoofing, IP address spoofing, attaques au DNS
- *social engineering*
- *phishing* (une forme d'ingénierie sociale)
- *direct-access attacks* (ex. : accès aux disques, installation de keyloggers)

Autres types de vulnérabilités et d'attaques

- *brute-force* attacks
- *backdoor* (une méthode « secrète » de contourner l'authentification)
- *Denial of Service (DoS)* (Distributed DoS ou DDoS, quand on prend le contrôle d'*autres machines qu'on a déjà compromis* pour lancer des DoS en parallèle)
- *eavesdropping* (analyse de paquets si vous avez accès au réseau physique ; les programmes des services secrets américains Carnivore, NarusInSight)
- *privilege escalation*
- *spoofing* Ex : MAC spoofing, IP address spoofing, attaques au DNS
- *social engineering*
 - *phishing* (une forme d'ingénierie sociale)
 - *direct-access attacks* (ex. : accès aux disques, installation de keyloggers)

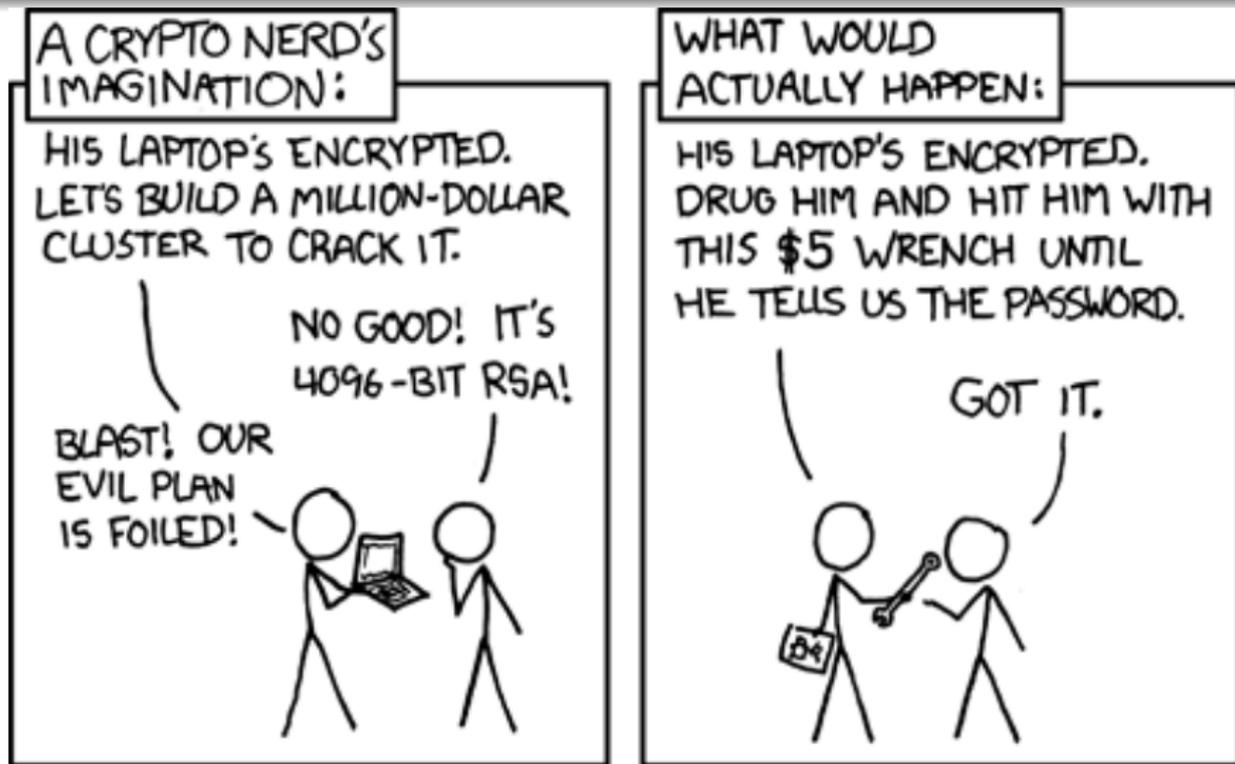
Autres types de vulnérabilités et d'attaques

- *brute-force* attacks
- *backdoor* (une méthode « secrète » de contourner l'authentification)
- *Denial of Service (DoS)* (Distributed DoS ou DDoS, quand on prend le contrôle d'*autres machines qu'on a déjà compromis* pour lancer des DoS en parallèle)
- *eavesdropping* (analyse de paquets si vous avez accès au réseau physique ; les programmes des services secrets américains Carnivore, NarusInSight)
- *privilege escalation*
- *spoofing* Ex : MAC spoofing, IP address spoofing, attaques au DNS
- *social engineering*
- *phishing* (une forme d'ingénierie sociale)
- *direct-access attacks* (ex. : accès aux disques, installation de keyloggers)

Autres types de vulnérabilités et d'attaques

- *brute-force* attacks
- *backdoor* (une méthode « secrète » de contourner l'authentification)
- *Denial of Service (DoS)* (Distributed DoS ou DDoS, quand on prend le contrôle d'*autres machines qu'on a déjà compromis* pour lancer des DoS en parallèle)
- *eavesdropping* (analyse de paquets si vous avez accès au réseau physique ; les programmes des services secrets américains Carnivore, NarusInSight)
- *privilege escalation*
- *spoofing* Ex : MAC spoofing, IP address spoofing, attaques au DNS
- *social engineering*
- *phishing* (une forme d'ingénierie sociale)
- *direct-access attacks* (ex. : accès aux disques, installation de keyloggers)

Social-engineering or direct-access attacks



<https://www.xkcd.com/538/> XKCD, Copyright © 2009 Randall Munroe. Licensed under a Creative Commons Attribution-NonCommercial 2.5 License.

Pour moi-même : "wrench" en français : « clé à molette ».

Détection de vulnérabilité

Port scanning :

- *nmap*
- *openvas*

Outils plus spécialisés :

- *nikto*

Fuzzing tester les programmes en donnant des entrées aléatoires

Rootkits déjà prêts, appliqués systématiquement et automatiquement (regardez `/var/log/auth.log` sur une machine exposée au public)

Outils spécialisés pour la détection (et l'exploitation) des vulnérabilités :

- *hydra*
- *sqlmap*
- *metasploit*

Détection de vulnérabilité

Port scanning :

- *nmap*
- *openvas*

Outils plus spécialisés :

- *nikto*

Fuzzying tester les programmes en donnant des entrées aléatoires

Rootkits déjà prêts, appliqués systématiquement et automatiquement (regardez `/var/log/auth.log` sur une machine exposée au public)

Outils spécialisés pour la détection (et l'exploitation) des vulnérabilités :

- *hydra*
- *sqlmap*
- *metasploit*

Détection de vulnérabilité

Port scanning :

- *nmap*
- *openvas*

Outils plus spécialisés :

- *nikto*

Fuzzing tester les programmes en donnant des entrées aléatoires

Rootkits déjà prêts, appliqués systématiquement et automatiquement (regardez `/var/log/auth.log` sur une machine exposée au public)

Outils spécialisés pour la détection (et l'exploitation) des vulnérabilités :

- *hydra*
- *sqlmap*
- *metasploit*

Détection de vulnérabilité

Port scanning :

- *nmap*
- *openvas*

Outils plus spécialisés :

- *nikto*

Fuzzing tester les programmes en donnant des entrées aléatoires

Rootkits déjà prêts, appliqués systématiquement et automatiquement (regardez `/var/log/auth.log` sur une machine exposée au public)

Outils spécialisés pour la détection (et l'exploitation) des vulnérabilités :

- *hydra*
- *sqlmap*
- *metasploit*

Détection de vulnérabilité

Port scanning :

- *nmap*
- *openvas*

Outils plus spécialisés :

- *nikto*

Fuzzing tester les programmes en donnant des entrées aléatoires

Rootkits déjà prêts, appliqués systématiquement et automatiquement (regardez `/var/log/auth.log` sur une machine exposée au public)

Outils spécialisés pour la détection (et l'exploitation) des vulnérabilités :

- *hydra*
- *sqlmap*
- *metasploit*

Être au courant des vulnérabilités déjà connues

Common Vulnerabilities and Exposures (CVE)

- références de la forme « CVE-AAAA-NNNN » (AAAA est l'année de publication et NNNN un numéro d'identifiant).

Une base de données publique des vulnérabilités connues, gérée par MITRE et financée par le département de la Sécurité intérieure des États-Unis

- *CERT-FR* est l'équivalent français

Les fournisseurs de logiciels s'occupent de publier des versions corrigées, et l'on peut tracer ces identifiants. Par exemple :

- <https://www.debian.org/security/cve-compatibility>

Mettre à jour le logiciels.

Si vous découvrez une vulnérabilité...

Si vous découvrez une vulnérabilité...

- Un *zero-day attack* devient possible
- **contactez les auteurs du logiciel** (ou les administrateurs du système, si c'est un problème de configuration), **en privé**

Prévention d'attaques : bonnes pratiques

- isoler du réseau ce qui ne doit pas être accessible à distance
- ne pas négliger *social engineering attacks*
- un nouveau logiciel qui n'a pas encore été testé longtemps peut contenir des bugs
 - (mais on trouve aussi parfois des vulnérabilités dans des logiciels très anciens (ex. : *Shellshock*) ou même des matériels (ex. : *Meltdown/Spectre*))
- mises à jours pour corriger les vulnérabilités *connues*
- authentification plus forte, par exemple multi-factor authentication
- un système simple est souvent plus sûr qu'un système complexe
 - **n'activer que les services nécessaires**
 - chercher un balance : on ne peut pas rendre la vie impossible aux utilisateurs

On peut **suivre des bonnes pratiques**, mais se protéger contre un attaquant ayant une grande disponibilité de ressources (ex. : un état) est très difficile.

Protection

Le *penetration testing* peut *mitiguer* les vulnérabilités.

Monitoring

Remerciements

Nicolas Grenèche m'a fourni une copie de ses transparents, que j'ai réutilisé en partie—en particulier, l'introduction.

Les experts de sécurité réunis à l'IUT de Roanne en occasion de l'Assemblée des Chefs de Département R&T m'ont donné plusieurs bonnes idées, en particulier le *Capture the Flag* pour débutants.

Les erreurs sont à moi.

Bibliography



Saiu, L. (2019). La page web de mes cours.

<http://ageinghacker.net/teaching>

La page web officielle du cours contient des pointeurs à des ressources web, et une copie des mes transparents.

Image credits and licenses

- <https://www.xkcd.com/538/>
XKCD, by Randall Munroe. Licensed under a Creative Commons Attribution-NonCommercial 2.5 License.