SLO

Sécurité logicielle

Chapitre 1.
Introduction

Sylvain Pasini
sylvain.pasini@heig-vd.ch

Mars - Juillet 2014

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Atelier

- Selon vous, qu'est-ce que la « sécurité logicielle » ?
  - 3 minutes - réflexion individuelle
  - 3 minutes - échange par groupe (3-5 étudiants)
  - Mise en commun

Atelier - Résultats

- Contrôle efficace entrées utilisateurs (anti-exploit)
- Anti-reverse
- Obfuscation
- Exploitation de failles
- Connexion chiffrée
- Identité/authentification
- Protection de licence
- Protection contre la copie
- Pas dépendant des fichiers éditables
- DRM
  Digital Right Management

L'importance de la sécurité logicielle

- REPORT TO THE PRESIDENT
- Cyber Security: A Crisis of Prioritization
- President's Information Technology Advisory Committee
- [source] http://www.nitrd.gov/pitac/reports/20050301_cybersecurity/cybersecurity.pdf

Cyber Security Research Priorities 37
1. Authentication Technologies 37
2. Secure Fundamental Protocols 38
3. Secure Software Engineering and Software Assurance 39
4. Holistic System Security 40
5. Monitoring and Detection 41
6. Mitigation and Recovery Methodologies 42
7. Cyber Forensics: Catching Criminals and Deterring Criminal Activities 43
8. Modeling and Testbeds for New Technologies 44
9. Metrics, Benchmarks, and Best Practices 45
10. Non-Technology Issues That Can Compromise Cyber Security . . .46

[source] http://www.nitrd.gov/pitac/reports/20050301_cybersecurity/cybersecurity.pdf

3.

Secure Software Engineering and Software Assurance

Commercial software engineering today lacks the scientific underpinnings and rigorous controls needed to produce high-quality, secure products at acceptable cost. Commonly used software engineering practices permit dangerous errors, such as improper handling of buffer overflows, which enable hundreds of attack programs to compromise millions of computers every year. In the future, the Nation may face even more challenging problems as adversaries – both foreign and domestic – become increasingly sophisticated in their ability to insert malicious code into critical software. From avoiding basic programming errors to developing massive systems that remain secure even if portions of the system software are compromised, significant new research on secure software engineering is needed. Research subtopics include:

[source] http://www.nitrd.gov/pitac/reports/20050301_cybersecurity/cybersecurity.pdf

Sécurité logicielle

Software development is not yet a science or a rigorous discipline, and the development process by and large is not controlled to minimize the vulnerabilities that attackers exploit.

Today, as with cancer, **vulnerable software can be invaded** and modified to cause damage to previously healthy software, and infected software can **replicate itself and be carried across networks** to cause damage in other systems. Like cancer, these damaging processes **may be invisible** to the lay person even though experts recognize that **their threat is growing**. And as in cancer, both **preventive actions and research are critical**, the former to minimize damage today and the latter to establish a foundation of knowledge and capabilities that will assist the cyber security professionals of tomorrow reduce risk and minimize damage for the long term.

[source] http://www.nitrd.gov/pitac/reports/20050301_cybersecurity/cybersecurity.pdf

Sécurité logicielle

Vulnerabilities in software that are introduced by mistake or poor practices are a serious problem today.

In the future, the Nation may face an even more challenging problem as **adversaries** – both foreign and domestic – **become increasingly sophisticated in their ability to insert malicious code into critical software**.

[source] http://www.nitrd.gov/pitac/reports/20050301_cybersecurity/cybersecurity.pdf

Evolution des logiciels

- A l'origine, nous avions des logiciels « simples »
- **1 - Connectivité**
  - Omniprésence des réseaux
  - Plus de logiciels connectés, plus d'attaques possibles
  - Problèmes de l'héritage des anciens logiciels
    - Non conçus pour être connectés
    - Transformés en services Web
  - Utilisation de mauvaises pratiques
- **2 - Extensibilité**
  - Conception de logiciels avec des fonctionnalités incrémentales
  - Systèmes d'exploitation avec modules et librairies, navigateurs Web avec modules, services Web construits sur des architectures extensibles (J2EE, .NET), etc.
- **3 - Complexité**
  - Plus de code, plus d'erreurs, plus de vulnérabilités

Complexité

http://www.informationisbeautiful.net/visualizations/million-lines-of-code/

Complexité

http://en.wikipedia.org/wiki/Source_lines_of_code

Complexité

http://www.informationisbeautiful.net/visualizations/million-lines-of-code/