



ORACLE®

Software Security Assurance

Get to know your vendors to preserve your security posture

Eric Maurice – Director, Oracle Software Security Assurance



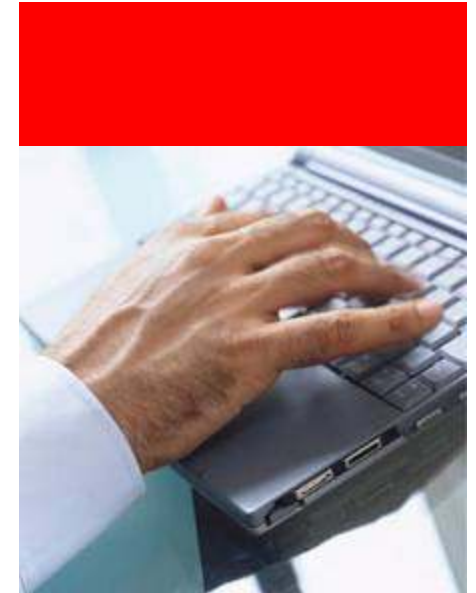
Abstract (hidden)

Vendors' Security Assurance practices have a tremendous impact on the security posture and cost of ownership of their customers. Yet, few customers are aware of the Security Assurance practices of their most strategic vendors, and even fewer customers actually assess their vendors' assurance practices when purchasing new systems or software. In this session, we will discuss:

- How the Security Assurance practices of the vendors/developers impact users of software
- What are the key elements of software security assurance, including secure development, vulnerability remediation (security patches) and disclosure practices, and independent security validations (FIPS, CC)
- What questions should be considered when purchasing new software and systems
- Briefly discuss the DHS Software Assurance initiatives (<https://buildsecurityin.us-cert.gov>) and other resources

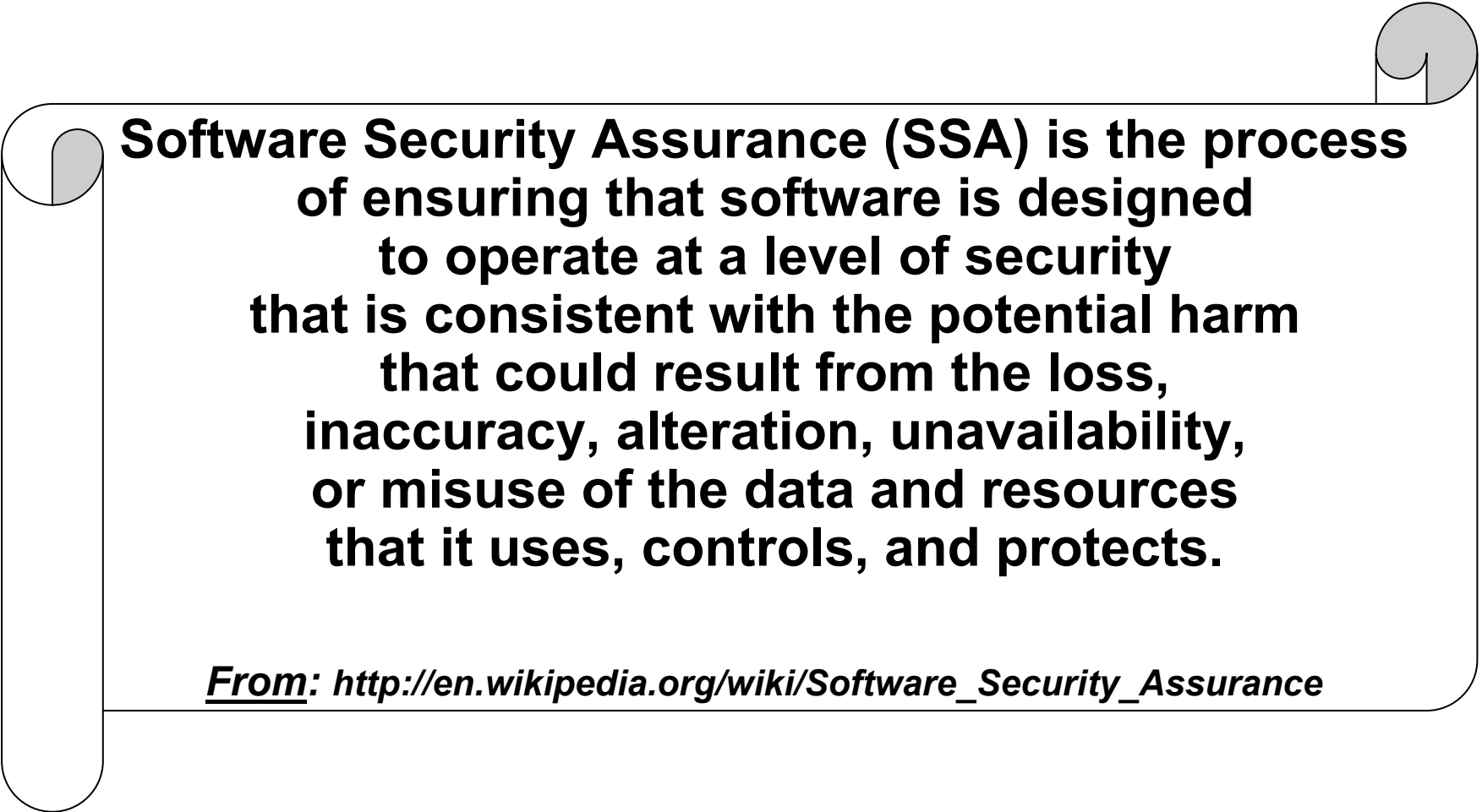
Today's Agenda

- What is Software Security Assurance?
- Why are we having a discussion about Software Security Assurance today?
- What should responsible software vendors (and developers) do?
 - Examples from Microsoft, Oracle, Cisco, and others
- What can you do to assess your suppliers' Assurance practices to protect your security posture?
- Software Assurance: is Open Source a better proposition?
- Conclusion & Questions





What is Software Security Assurance?



Software Security Assurance (SSA) is the process of ensuring that software is designed to operate at a level of security that is consistent with the potential harm that could result from the loss, inaccuracy, alteration, unavailability, or misuse of the data and resources that it uses, controls, and protects.

From: http://en.wikipedia.org/wiki/Software_Security_Assurance

Challenges

Why are we discussing Software Assurance today?

- In 2003, intangible assets in the U.S. economy accounted for \$5 trillion, or over one-third of the value of U.S. domestic corporations
 - For many organizations, intangibles assets exist electronically, in their corporate IT systems
- What if organizations could not trust the security controls provided by their electronic infrastructure?





Challenges

Why are we discussing Software Assurance today?

- Software is an important piece of the critical infrastructure
 - Software operate businesses and governments
 - COTS replace custom-designed systems
 - Systems are now interdependent
 - Vendors, suppliers, partners, customers, etc
- ↳ How can you make sure that your systems are free of major security defects? (unintentional security bugs)
 - ↳ How can you make sure that your systems are free of time bombs, Trojan, back doors, etc.? (intentional malware / secure supply chain issue)
 - ↳ How can you make sure that exposing your systems to a trusted third-party will not result in fully compromising your security posture?



Software Security Assurance

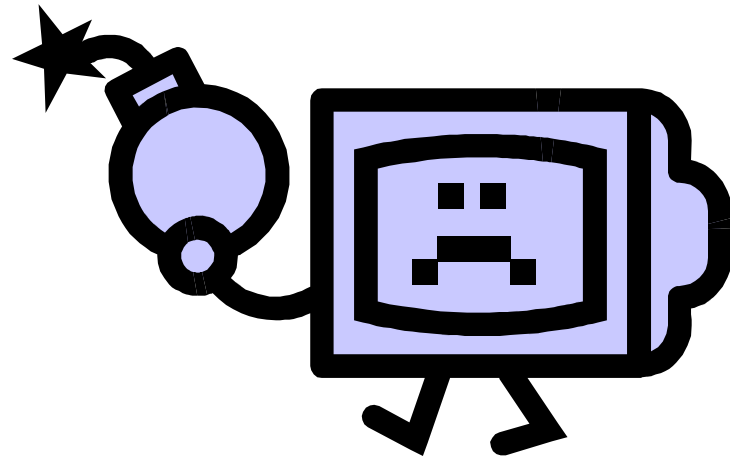
Preliminary thoughts

- Despite best effort, *complete, guaranteed* absence of vulnerability *forever* is a pipe dream
 - Enterprise, networked applications are complex
 - Threats evolve (and so does the skill set of the bad guys)
 - Deployments and uses are different from sites to sites, customers to customers, etc.
- For users, expectations of “vulnerability free” usually equates expectations of no security bug fixes (for an undetermined period)
 - Invite bad behavior by vendors (silent fixing, no fixing, etc.)
 - Little incentive for vendors to do ongoing assurance
- And then, there is the issue of the *software supply chain*
 - How can we make sure that software is free of intentionally-created vulnerabilities (malware)?
 - Is software more secure when developed domestically?

Software Security Assurance

Preliminary thoughts (cont'd)

- Unlike other products, software must be developed with the expectations that once in production, it will be placed in a malicious environment





Software Security Assurance

What should responsible vendors do? (key principles)

1. Vendors should adopt formal policies and procedures to foster secure development (reflecting a lifecycle view of security)

Prevent

2. Vendors should have transparent Security Assurance Policies (i.e. relevant policies should be known by customers)

Communicate

3. While vulnerabilities should be prevented as much as possible, vendors should have an effective and appropriate security patching program

Remediate



Software Security Assurance

What should responsible buyers do? (key principles)

1. Buyers should assess the security assurance practices of their suppliers *BEFORE* buying their products
2. *[In addition, they should show the same level of care when developing their own applications, or deploying new systems and applications]*



1/3. Lifecycle Approach to Security

Sample questions to ask your software vendors

- **Security must be a part of the “corporate DNA”**
 - Who is responsible for security assurance in the organization?
- **Security needs to be built-in, not bolted on**
 - Are security requirements expressed in the definition and design phases *before* actual coding starts?
- **Secure development practices should be followed throughout development**
 - Are developers trained on secure development practices?
 - Are security testing and code reviews taking place throughout the development of the products?
 - Is compliance with the security requirements documented in the various phases of development? Are products submitted for security assessment before being released (ethical hacking, etc.)?
- **Security Assurance shouldn’t stop when a software is released (i.e. Ongoing Assurance)**
 - Does the organization have formal procedures to work with customers, partners, security researchers when dealing with security bugs?
 - Does the organization have an effective security patching mechanism?
 - Does the organization submit its products for independent security validations (CC, FIPS, etc.)?



2/3. Transparent Security Assurance Policies

Sample questions to ask your software vendors

- **The vendor's Security Assurance practices, policies, and procedures must be known and understood by its customers**
 - Does the vendor have *formal* Secure Coding Standards?
 - How well trained are developers on these Standards?
 - What are the security fix policies?
- **The vendor must commit to meaningful external (i.e. independent) security validations**
 - Has the vendor obtained Common Criteria or FIPS validations for its products?
- **The vendor must have appropriate security vulnerability disclosure policies**
 - When are vulnerabilities disclosed?
 - Are all customers treated equally?
 - How are the severity of the bugs reported? (e.g. use of CVSS)



3/3. Effective Security Patching Program

Sample questions to ask your software vendors

- **A vendor must have a formal security bug remediation program**
 - What is the vendor's security patch program?
- **The frequency with which security patches are released must be predictable and appropriate**
 - How often are security bug fixes released?
 - Is the security patch release schedule known publicly?
 - If on a fixed schedule, does the vendor retain the ability to issue one-off security fixes for *critical* bugs?
- **The patch documentation, in particular the disclosures related to the severity of the vulnerabilities and the affected components must be appropriate**
 - What is disclosed in the patch documentation?
 - Does the vendor use a standard severity rating such as CVSS?
 - Are vulnerabilities clearly identified (use of CVE)?



For more information about what questions to ask

- DHS' Build Security In : “Software Assurance (SwA) in Acquisition: Mitigating Risks to the Enterprise”
 - <https://buildsecurityin.us-cert.gov/daisy/bsi/dhs/908-BSI.html>
- Echelon One: Security Criteria for Selecting a Database
 - <http://www.oracle.com/corporate/analyst/reports/infrastructure/sec/e1-oracle-microsoft-security-comparison.pdf>



Software Security Assurance

Is Open Source a better (i.e. more secure) alternative?

- Open Source advocates often claim “better security”
 - “Code is available for ALL to review”
- However, in reality, such a claim cannot be validated
 - Security strength (resilience) depends of level of scrutiny the code received and the skills of the reviewers
 - Access to source code may help malicious hackers when developing exploits (accelerated creation of exploits when new vulnerabilities are discovered)
 - Patching cannot be done on a fixed schedule



Software Security Assurance

What is the meaning (if any) to the number of published vulnerabilities?

- Certain vendors make claim related to the number of published vulnerabilities
 - “This product has X number of published vulnerabilities, it is NOT secure...”
- This kind of claims is *stupid and misleading* (“a red herring” for many analysts). This is because:
 - The number of vulnerabilities has very little (if anything) to do with the actual security features or controls provided by a piece of software (a vulnerability-free product may not have a single security feature)
 - There is no standard way for counting vulnerabilities (does a flaw in a section of code that results in a weakness that can be exploited on n interfaces a single vulnerability?)
 - Claim is based on *published* vulnerabilities
 - That is the ones that are publicly known and disclosed by the vendor (incentive for unscrupulous vendor to NOT announce vulnerabilities and possibly engage in systematic silent fixing)
 - This number is affected by the vendor’s disclosure practices
 - This number is affected by how popular the product may be with security researchers,
 - Etc.



Conclusion (1/2)

- Organizations can improve their security posture by assessing the Software Security Assurance of their vendors
 - Buyers need to be educated to ask the right questions:
 1. Has the vendor adopted formal policies and procedures to foster secure development (reflecting a lifecycle view of security)?
 2. Does the vendor have transparent Security Assurance Policies (i.e. relevant policies should be known by customer)?
 3. Does the vendor have an effective and appropriate security patching program?
- Implications of Software Security Assurance go beyond the number of published vulnerabilities:
 - Actually, claims related to the number of published vulnerabilities do little to help organizations improve their security posture



Conclusion (2/2)

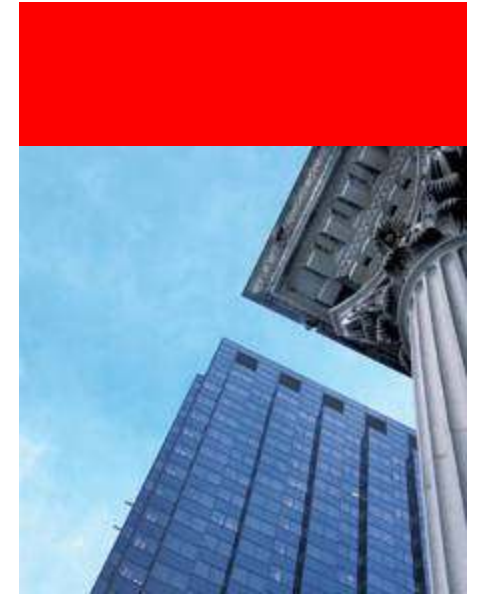
- Organizations need to have a “*defense in depth*” (i.e. layered) approach to security
 - Availability of external controls can mitigate vulnerabilities in affected product (by preventing exploitations)
 - HOWEVER, organizations need to include security patching activities in their normal maintenance windows

Final Thoughts

Security and Culture



- Corporate culture ultimately sets the course for process, people, plans, policies
- Changing corporate culture is like turning an oil tanker
- Process, plans, policies, people cannot protect against indifference
- “You can’t hire enough security police”
- Security must become part of corporate genetic material (nature) as implemented by plans, policies, process (nurture) and verified by compliance



ORACLE®