Integrated Site Security for Grids

www.isseg.eu

The Integrated Site Security for Grids (ISSeG) project complements Grid security by providing recommendations and training for each Grid site to improve their Site Security.

Information Society and Media

CERN

Forschungszentrum Karlsruhe in der Helmholtz-Gemeinschaft

Science & Technology Facilities Council

eGee Related Project

Integrated Site Security for Grids is co-funded by the European Commission – Project Number 026745

ISSeG

Integrated Site Security for Grids

Throughout the lifetime of the project, www.isseg.eu will be populated with training resources and recommendations. Below are some extracts from the website.

System administrator security checklist

1. **Harden** the operating system and the applications
2. Keep the operating system and the applications **up-to-date**
3. Use a local **firewall**
4. Take advantage of the **logs**
5. Ensure that all **passwords** are secure
6. Take extra precautions for **privileged accesses**
7. Use **security products** (e.g. anti-virus, intrusion detection, integrity checkers)
8. Take into account **physical security**
9. Keep your **security knowledge** up-to-date

For more details go to www.isseg.eu

Software developer security checklist

1. General: Design your software with security in mind from the beginning. Adding security later will never work.

2. Architecture: Divide the program into semi-independent parts, each part should work correctly even if others fail. Build multiple layers of defense. Simple solutions are usually the most secure.

3. Design: Make security-sensitive parts of your code small. Don't require more privileges than you need. Avoid standard default passwords. Deny by default. Limit resource consumption, to limit the likelihood or impact of a Denial of Service attack. Fail securely: e.g., if there is a runtime error when checking a user's access rights, assume s/he has none. In distributed or web applications don't trust the client.

4. Cryptography: Use trusted, public algorithms, protocols and products.

5. Implementation: Read and follow guidelines for your programming language and software type. Think of the security implications of what your code does. Reuse trusted code (libraries, modules etc.). Write good-quality, readable and maintainable code (bad code won't ever be secure)

6. Coding: Don't trust input data. Validate all input data. Don't make any assumptions about the environment. Beware of race condition. Deal with errors and exceptions. Fail gracefully. Protect passwords and secret information. Be careful when handling files, especially temporary files: don't fall for the symbolic link attack. Be careful with shell calls, eval functions etc.

7. After implementation: Review your code and let others review it. When a (security) bug is found, search for similar ones. Use tools specific to your programming language: memory testers, bug finders etc.

For more details go to www.isseg.eu

General user security advice

1. Do not tell anyone your **password:** even if they claim to be official or support staff.

2. Be aware of the different types of **spam: Curiosity** ('look at this', empty mail, …); **Trust** (seemingly from a friend, colleague, …); **Authority** (seemingly from security, management, …)

3. Do not click on **web links** in unexpected emails, instant messages and chat. "Fake" web links can link to a different web site than expected and can be used to infect your computer or steal your password.

4. Do not open **attachments** that you are not expecting.

5. Avoid installing unnecessary software: it may contain Trojan horses, spyware or other malicious software that could infect your computer. Quick online research should help identify malicious software.

6. Configure to run without **administrator privileges** to minimize the consequences of an "attack".

For more details go to www.isseg.eu

Information Society and Media

CERN

Forschungszentrum Karlsruhe in der Helmholtz-Gemeinschaft

Science & Technology Facilities Council

EGEE Related Project