Funded by the European Commission
Seventh Framework Programme

Cyber ROAD

Development of the Cybercrime and Cyber-terrorism Research Roadmap

Grant Agreement N. **607642**

SDLC & Architectures

Author(s): Massimiliano Aschi (POSTEIT)

CC RG-41 and CT RG-41 - Development of coding standards for secure and fault-tolerant cyber-physical system development
CT RG-14 - Standardized interfaces to external modules and systems for providing in-depth security.
CT RG-40 - Component and System level penetration testing procedures during development and integration of complex systems.
CC RG-97 and CC RG-100 - Extending current security architectures to large distributed systems including players with different background knowledge in Security. Furthermore - CC RG-57 (for Cybercrime) - new authentication mechanisms need to be devised that provide usable security and continuous authentication.
CC RG-81 - Methodologies to increase preparedness and responsiveness in case of attack, including attacks - CT RG-75 (for Cyberterrorism) - targeting production process (e.g. to insert malware, backdoors or Trojan in final products).
CT RG-94 - Intelligent IDS, hardening mechanisms and awareness.

ABSTRACT

Insecure software, even when executed in trusted computing environments, still represents a potential enabling factor for cybercrime and cyberterrorism scenarios. At the same time, even the most secure and bug-free software when run in untrusted execution environments, represents an important and often underestimated risk. Hence, the adoption of threat modelling techniques is to be considered fundamental in order to implement a Secure Software Development Lifecycle (SDLC) enabling the implementation of risk management best practices and the usage of well-known metrics. Other promising and interesting techniques aiming at protecting programs even if they contain vulnerabilities are still considered not mature and further research efforts needs to be spent in order to let them being effective. The efficiency of the threat modelling process is also an important factor affecting the security of developed software: all threats should be modelled, understood and properly rated as relevant (or not) considering the large degree of subjectivity involved in this process.

Some areas of improvement and suggested strategies are listed below:

- Support the widely adoption of SDLC and threat modelling techniques - especially for mobile app development - by simplifying management frameworks (CMMs), reducing the time-to-market for application development and the cost for developers' education.
- Improve support to, and encourage the adoption of management frameworks (CMMs) and SDLC for novel environments and paradigms such as IoT and wearables
- Simplify the existing techniques for SDLC and reduce the dependency by specific vendors
- Integrate threat modelling and risk management focusing on how to transform threats in the final risk metrics
- Increase the awareness on the attack techniques of whoever is involved into software products' development
- Create a certification of quality for SDLC
- Improve mechanisms for automatic generation of exploits for software vulnerabilities in order to ideally automate discovery and exploitation tasks

Secure application architectures are another big topic to keep in mind while developing software. Secure design principles must be taken into account since the early development stages, facing the big challenges raised by the usage of distributed, complex systems and by the novel paradigms (e.g. cyber-physical systems, swarms, grids, clouds, etc.) each with its own uniqueness and challenges.

RESEARCH ACTION #1

Development of coding standards for secure and fault-tolerant cyber-physical system development

DISTANCE TO THE MARKET: **TRL 7**
COST OF THE TOPIC: **4 STREPs + 2 IP**
AVAILABILITY OF COMPETENCE IN EUROPE: **5**
TIME SPAN FOR ADDRESSING THE ACTION: **48 months**
ACTORS: **manufacturing industries, smart factories, machine-to-machine connectivity, smart sensors, smart tags, industrial control systems, autonomous cyber-physical systems, IoT devices, Internet of Services (IoS)**

Cybercriminals (and Cyberterrorists too) are showing to have cyber-physical systems in their sights, even if driven by different motivations.

Hacking Cyber-Physical/Industrial Systems controlling gas pipelines or energy grids, shutting down national transportation critical infrastructures or gaining complete control of the on-board systems of an airplane, enable modern terrorism scenarios that have the potential to inflict massive damages affecting hundred thousands of lives, having a considerable psychological impact while granting at the same time total anonymity.

The opportunity to steal intellectual property or acquire sensitive data from Industry 4.0 companies, is being more and more exploited by Cybercriminals in order to monetize information on the dark web, or to limit competitive advantage of a competitor maybe sabotaging its production chain.

New threats that will affect cyber-physical systems can be faced through:

- High-level categorization of control systems to enable cyber awareness and readiness
- Advanced Risk analysis should be performed and model-based (formal) approach to system development needs to be adopted.
- Industries have to increase their responsiveness and preparedness developing proper methodologies to face the possibility of a successful attack in place.
- At the same time innovative monitoring and detection techniques must be put in place in order to attempt the detection of suspicious activities in the factory (e.g. detect change in the manufacturing processes and activities).
- Use goal-based programming instead of programming for specifications

RESEARCH ACTION #2

Standardized interfaces to external modules and systems for providing in-depth security.

DISTANCE TO THE MARKET: **TRL 7**
COST OF THE TOPIC: **3 STREPs + 1 IP**
AVAILABILITY OF COMPETENCE IN EUROPE: **5**
TIME SPAN FOR ADDRESSING THE ACTION: **42 months**
ACTORS: **home automation systems, cyber-physical systems, interacting SW/HW modules**

Trusted and standard software interfaces as well as secure hardware interfaces are the building blocks upon which we have to build Trusted Computing in a highly automated environment. The main goals of trusted software/hardware components are to guarantee a standard level of security and to increase usability of components while accessing services offered by untrusted modules or systems (e.g., services offered by third parties). Furthermore, standard trusted components facilitate maintenance of the overall system, allowing focusing most of securing efforts on single systems/components and easing preventive-monitoring activities.

To enhance systems' security, we should promote a key principle of design that provides for denying full access to available resources by default. Access to resources should be always brokered by interfaces providing well-defined entry-points. Providing direct access to untrusted/uncontrolled modules/systems, could lead to disastrous results in terms of integrity and availability of highly automated components/services. Development and usage of property management gateways is strongly suggested so as that interacting subsystems and modules could be more reliable, secure and safe.

Component and System level penetration-testing procedures during development and integration of complex systems.

DISTANCE TO THE MARKET: **TRL 6**
COST OF THE TOPIC: **3 STREPs + 2 IP**
AVAILABILITY OF COMPETENCE IN EUROPE: **4**
TIME SPAN FOR ADDRESSING THE ACTION: **42 months**
ACTORS: **penetration testers, software developers, ICT Security industry, cyber-physical systems, swarms, grids, clouds, complex environments**

The development and integration activities related to components and systems parts of complex and heterogeneous environment (e.g. cyber-physical systems, swarms, grids, clouds, etc.), should always include systematic penetration-test activities. As well as unit and integration tests are executed during software development in order to fulfil functional and non-functional requirements and to detect unpredicted behaviours, component and system level penetration-tests should always be run as a mitigating action against possible future attacks brought either at system or at module/component level. Proper supporting methodologies, tools and techniques needs to be developed.

RESEARCH ACTION #4

Extending current security architectures to large distributed systems including players with different background knowledge in Security. Furthermore, new authentication mechanisms need to be devised that provide usable security and continuous authentication.

DISTANCE TO THE MARKET: **TRL 3**
COST OF THE TOPIC: **6 STREPs + 4 IP**
AVAILABILITY OF COMPETENCE IN EUROPE: **2**
TIME SPAN FOR ADDRESSING THE ACTION: **54 months**
ACTORS: **ICT Security industry, research centres specialized in security, software developers, standardization bodies**

Developing secure application architectures in large distributed systems is surely a big challenge. Secure design principles and well-known best-practices must be taken into account since the early design stages facing the big issues raised by the usage of distributed, complex systems and by the novel paradigms (e.g. cyber-physical systems, swarms, grids, clouds, etc.) each with its own uniqueness and challenges. The adoption of security-by-design principles, of design/coding standards and standardized interfaces to access resources/services, should also aim to reduce any issue related to potential security skill-gaps of the different stakeholders involved in the development of systems or components. Authentication mechanisms (also between modules/components when applicable) must be improved and adapted to novel, complex environments: they have to be also simpler, more effective and usable. Continuous authentication techniques should also be further developed and widely adopted.

Methodologies to increase preparedness and responsiveness in case of attack, including attacks targeting production process (e.g. to insert malware, backdoors or Trojan in final products).

DISTANCE TO THE MARKET: **TRL 6**
COST OF THE TOPIC: **6 STREPs + 3 IP**
AVAILABILITY OF COMPETENCE IN EUROPE: **3**
TIME SPAN FOR ADDRESSING THE ACTION: **48 months**
ACTORS: **cyber-physical systems, software, hardware, IoT, IoS, system developers, integrators, importers, stockists**

In recent years, software and hardware manufacturers - even some of the smallest one in the value chain – have been affected by attacks to their production line and processes. Mainly motivated by espionage purposes or by criminal lucrative intents, these attacks aimed at compromising the integrity of legitimate software/firmware injecting malware, backdoors or Trojans in final products. The potential impact of such kind of attacks is huge: imagine what could happen if the tampered firmware of a PC's motherboard would be replicated in hundred thousands of copies and sold in tens of different countries of the world.

New methodologies and tools to mitigate the risk of such kind of attacks happening need to be developed.

Software production lines must be considered an attractive target and an asset to defend from potential insiders and cyberattacks. The integrity of software and production processes should be monitored and granted at all times. Incident management procedures (including proper countermeasures) should be in place and personnel should be properly trained.

RESEARCH ACTION #6

Intelligent IDS, hardening mechanisms and awareness.

DISTANCE TO THE MARKET: **TRL 5**
COST OF THE TOPIC: **8 STREPs + 4 IP**
AVAILABILITY OF COMPETENCE IN EUROPE: **3**
TIME SPAN FOR ADDRESSING THE ACTION: **52 months**
ACTORS: **ICT professionals, ICT Security vendors, ICT vendors, standardization bodies, educators**

When facing novel threats, current signature/rule-based IDS systems show their limits: it is necessary to develop a new generation of autonomous, intelligent, adaptive IDS systems that for example should be able to learn new ways to mitigate or counter previously unknown attacks by experience and by inference and logic.

While keeping them effective, system hardening techniques must be simplified and proper tools and techniques developed (e.g. driven procedures to reduce the attack surface without the need to own deep security skills). Hardening should be taken into account in system design, development and delivery phases.

While struggling to counter increasingly sophisticated attacks, according to statistics the most effective techniques are still relatively simple and well-known. That's why awareness campaigns, tools and methodologies must be further developed for professionals implied in software development and system design.