Security Evaluation of a Linux-based Operating System: An Industry Experience.

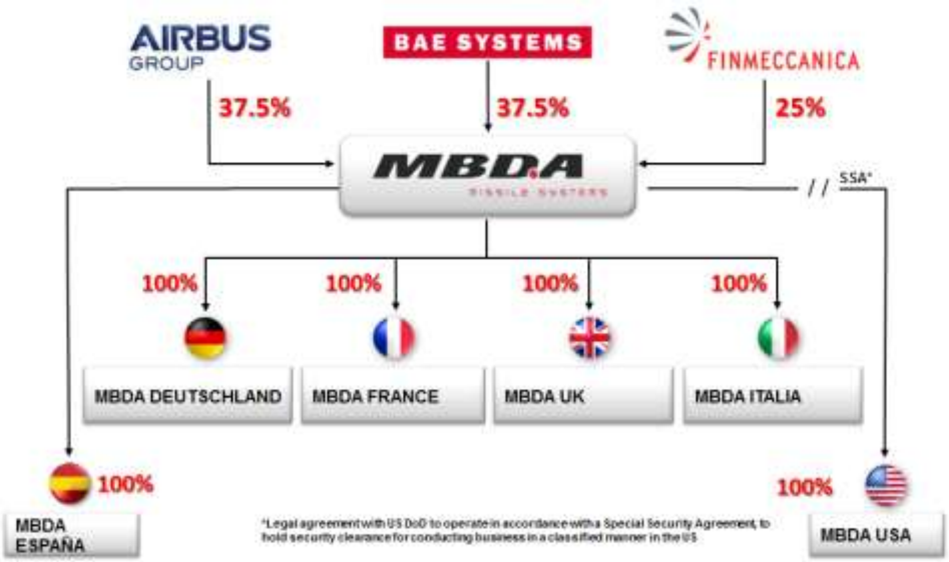
Giuseppe Procopio

MBDA Italia S.p.a – IRAD & Innovation
Software Engineering & Tecnhology

Security evaluation of the FIN.X SE V4.0:

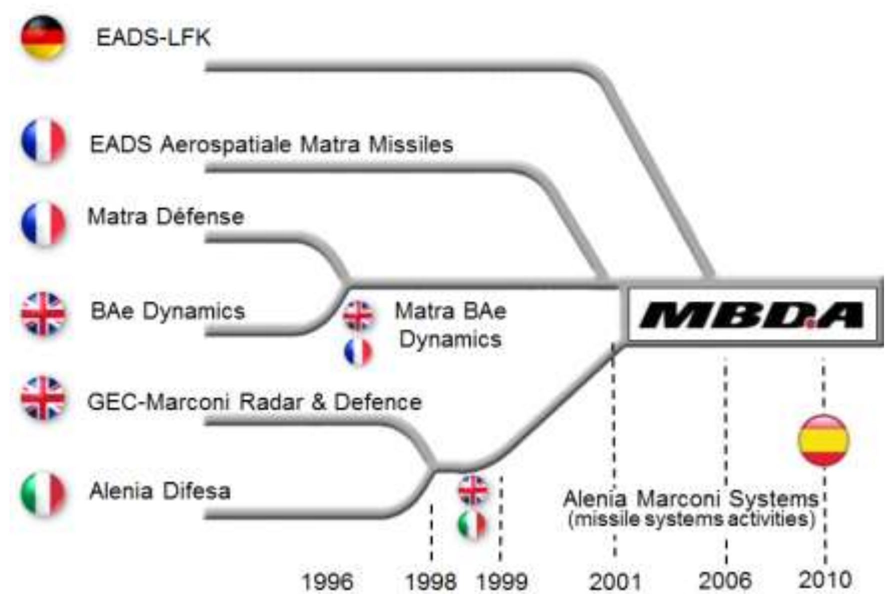
- **Introduction to FIN.X SE V4.0**
- The Common Criteria scheme
- Risk analysis
- Conclusions

MBDA



- Created in 2001 , MBDA is an industry leader in the defense sector
- Extensive international experience in the market of missiles and missile systems

- Three major shareholders: Airbus Group, BAE SYSTEMS, and Finmeccanica



FIN.X

- The FIN.X is a Linux-based operating system derived from the Gentoo distribution, whose strengths are its high flexibility, scalability, configurability and customization

FIN.X RTOS

RTCA/DO-178B Level D

RTCA DO-178B

- DO-178B Level D compliant
- Support for safety-critical applications

FIN.X RTOS

Security Enhanced EAL4+

Common Criteria EAL4+ Certified

- Common Criteria EAL4+ compliant
- Support for security-critical applications

FIN.X RTOS

- Desktop, workstation, and server (like Red Hat/Ubuntu).

FIN.X SE V4

- It follows the FIN.X SE V3.1, the first CC EAL4+ certified operating system in Italy :
  - https://www.commoncriteriaportal.org/files/epfiles/rc_finx_rtos_se_v1.0.pdf
- Designed for use in embedded systems, with real-time constraints, and operating in security-critical environments, where "the mission’s success" is the primary need
- Support to cyber-resilience of systems

Security evaluation of the FIN.X SE V4.0:

- ✓ Introduction to FIN.X SE V4.0
- ➢ The Common Criteria scheme
- ➢ Risk analysis
- ➢ Conclusions

The Common Criteria (ISO 15408)

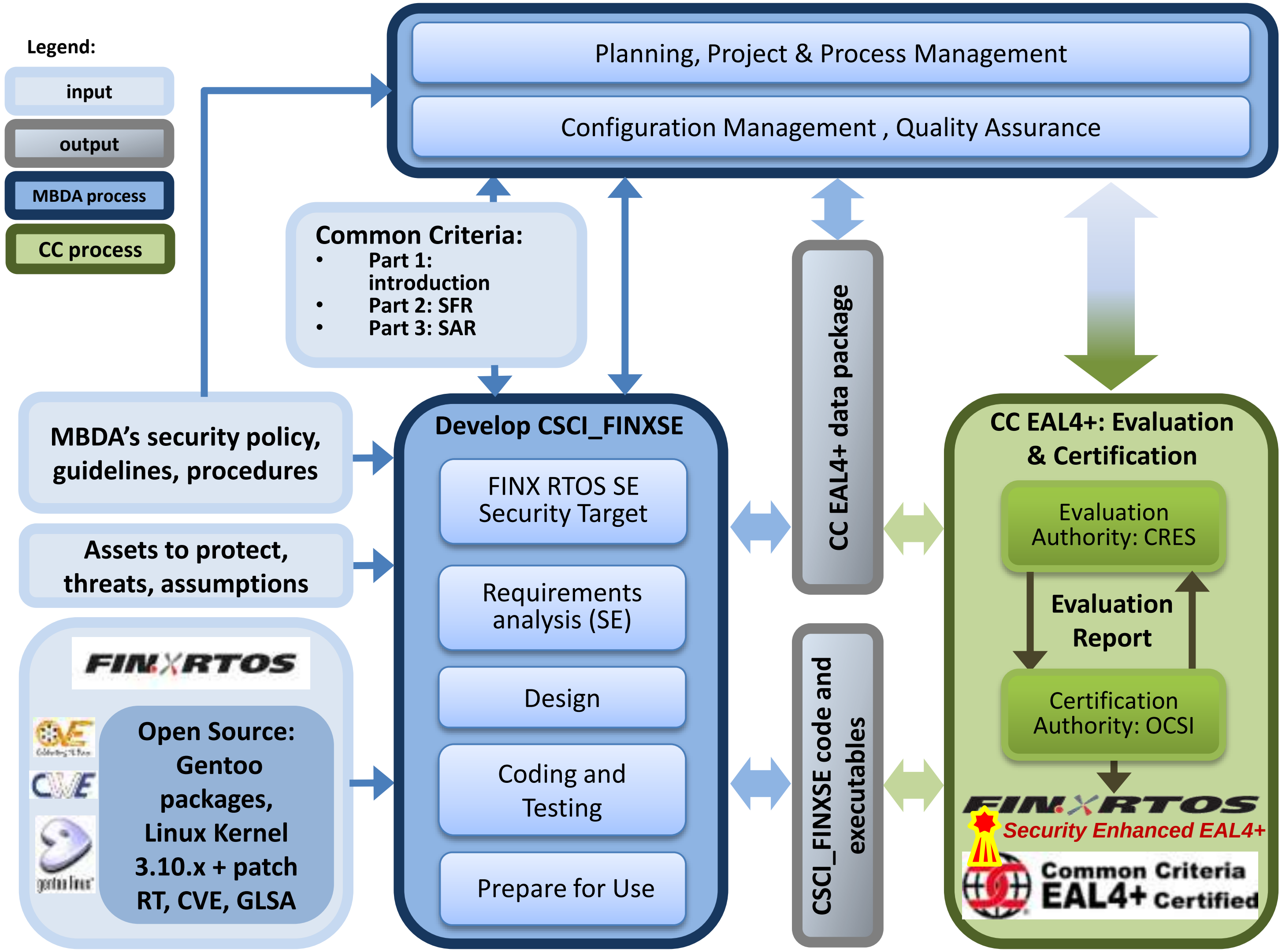
- An internationally recognized standard for evaluating the security capabilities of information technology hardware and software
- It provides a scheme where product or systems are evaluated by professional third parties with the aim to verify that they meet their security objectives
- 7 levels of quality assurance: EAL1 (low) -> EAL7 (high)
- Why getting FIN.X SE V4.0 certified ?
  - Compliance to CC is often a prerequisite for system's acceptance and it is recognized by all members of the CCRA
  - Safety's certification and security's certification became during the last years the dominant source of competitive differentiation for the OS's market, which is shared by few competitors mostly subjected to export restrictions and maintaining higher prices
  - The market analysis suggested placing the FIN.X SE V4 to the level of the leading competitors (RedHat , Suse , WindRiver , etc.) which is the level EAL4 increased with flaw remediation

The FIN.X SE Development and Evaluation Process

FINX RTOS SE V4 project's owner

Certification Authority (member of the CCRA)

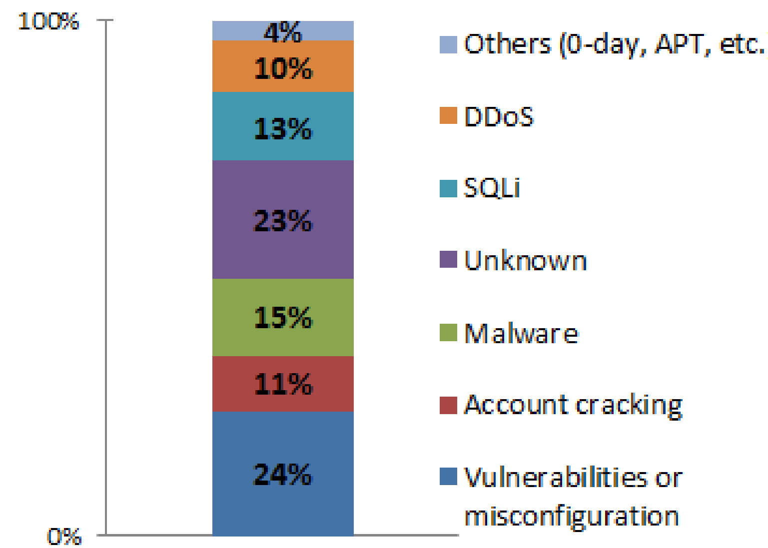
Evaluation Authority (accredited by OCSI)



Security evaluation of the FIN.X SE V4.0:

- ✓ Introduction to FIN.X SE V4.0
- ✓ The Common Criteria scheme
- ➢ Risk analysis
- ➢ Conclusions

Risk Analysis: threats evaluation (1/2)



- Common attack mechanisms (http://clusit.it/download/Rapporto_Clusit%202014.pdf):

Risk Analysis: threats evaluation (2/2)

- CC certification's process: main threats countered by the FIN.X SE V4.0
  - Unauthorized access to resources and/or information (internal to the system or sent over the network)
  - System integrity corruption
  - Inability to associate an action to the requesting user
  - Inability to perform traceability analysis

Risk Analysis: countermeasures

Security problem
- Assets
- Treats
- Security policy

→

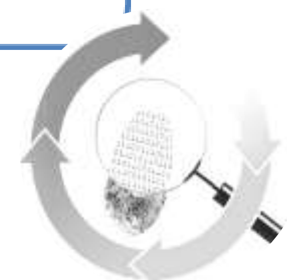
Security objectives
- Countermeasure are sufficient

→

Security Functional Requirements
- Security Audit
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TOE Security Functions
- TOE Access
- Trusted path/channels

✓ **Discretionary Access Control**
✓ **Security Management**
✓ **Resource's access management**

✓ **Advanced audit**
✓ **Intrusion detection**
✓ **Forensic analysis**



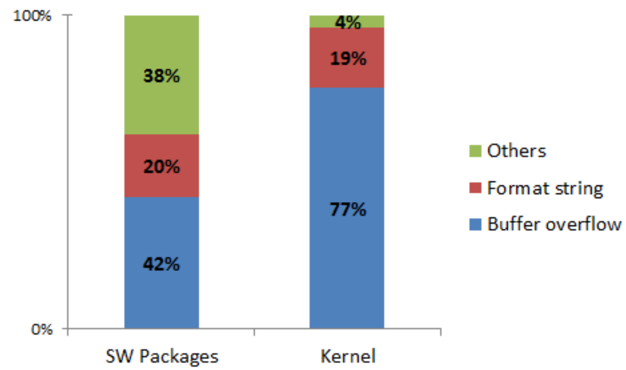
✓ **Advanced user management**
✓ **Advanced identification and authentication**

✓ **Strong cryptographic supports**

Software weaknesses

- The Open Source software:
    - Inherently vulnerable (not tied to a *secure* life cycle)
    - Very difficult to sanitize (high rate of **weaknesses**)

Common weaknesses reported by static analysers



- Current response to newly discovered vulnerability is to apply security patches, **BUT**:
    - Patches may be not so easy to apply
    - «Flaw Remediation» process may imply huge costs for system integration and re-validation
    - What can we do ?

Proactive defence

- Protection against memory corruption:
  - Use of Stack Canary (Stack Smashing Protector)
  - Detecting buffer overflows in functions that perform operations on memory and strings
  - Mark specific sections as «read-only»
  - Other executable' segments cannot be both writable and executable
  - Prevent stack and heap memory areas from being executable
- Configuration (partitioning layout, resource allocation, filtered access, authorized user account, etc.)
- Provide a suite of strong cryptographic algorithm
- Where needed, change the code to rule out insecure options
- Only signed code, from know host
- Only software required for the intended use

FIN.X SE V4.0: proactive defence in practice (1/2)

- Behaviour of executables under memory corruption attack

coverage

- **Attack case 1:** overwriting read-only sections

```
admin@finx-se ~ $ cc test.c -Wl,-z,relro -o test
admin@finx-se ~ $ ./test
Segmentation fault
admin@finx-se ~ $
```

100%

- **Attack case 2:** «classic» buffer overflow

```
admin@finx-se ~ $ cc -Wall -fstack-protector-all test.c -o test
admin@finx-se ~ $ ./test
*** stack smashing detected ***: ./test terminated
Segmentation fault
admin@finx-se ~ $
```

51%

- **Attack case 3:** buffer overflow by memory string operation

```
admin@finx-se ~ $ gcc test.c -O2 -D_FORTIFY_SOURCE=2 -o test
admin@finx-se ~ $ ./test
*** buffer overflow detected ***: ./test terminated
Aborted
admin@finx-se ~ $
```

99%

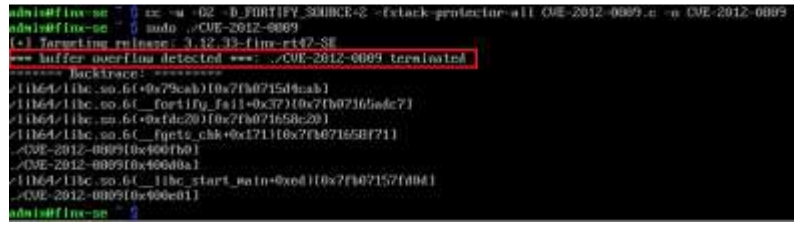
- **Attack case 4:** shell code

```
admin@finx-se ~ $ gcc test.c -o test
admin@finx-se ~ $ ./test
Segmentation fault
admin@finx-se ~ $ gcc test.c -zexecstack -o test
admin@finx-se ~ $ ./test
sh-4.2$
```

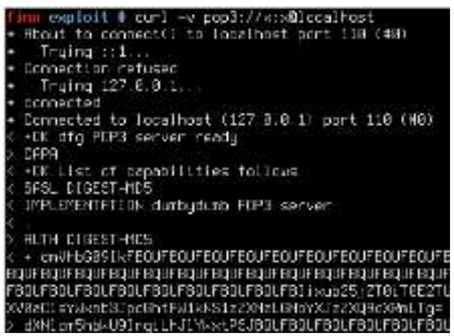
100%

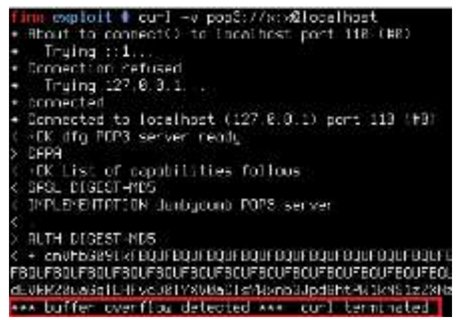
FIN.X SE V4.0: proactive defence in practice (2/2)

- Real cases:
  - CVE-2012-0809 (arbitrary code via format string sequences)



  - CVE-2013-0249 (Stack-based buffer overflow)





- But, results below expectations for kernel

Metrics

- Estimation of exposure to emerging vulnerabilities:
  - 90% of false positive for the kernel thanks to configuration tuning
  - Still in progress for software packages
- Packages (-fstack-protector-all, -O2 –D_FORTIFY_SOURCE=2, -fPIE -Wl,z,relro)
  - 70 % of software packages
  - Size overhead < 10%
- Kernel (-fstack-protector, CONFIG_DEBUG_RODATA, CONFIG_PROC_KCORE)
  - Size overhead < 1%
- CPU overhead < 5%
- Security tests:
  - > 800 tests
- Non regression tests:
  - > 4500 tests (basic system executables and kernel)

Security evaluation of the FIN.X SE V4.0:

- ✓ Introduction to FIN.X SE V4.0
- ✓ The Common Criteria scheme
- ✓ Risk analysis
- ➢ Conclusions

Conclusions

- FIN.X SE V4.0 currently under the Common Criteria scheme
- Open Source software is not always developed with security in mind
- Common practice is to patch newly discovered vulnerabilities
- But, flaw remediation may be unpractical or very costly
- The proposed approach enforces proactive defences together with reactive ones