10 Kaspersky Industrial
Cybersecurity
Conference

IEC 62443-4-1 certificate: KICS for Networks.
Как мы доказывали, что безопасность разработки ЛК на высоте

Dmitry Shmoylov

Head of Software security

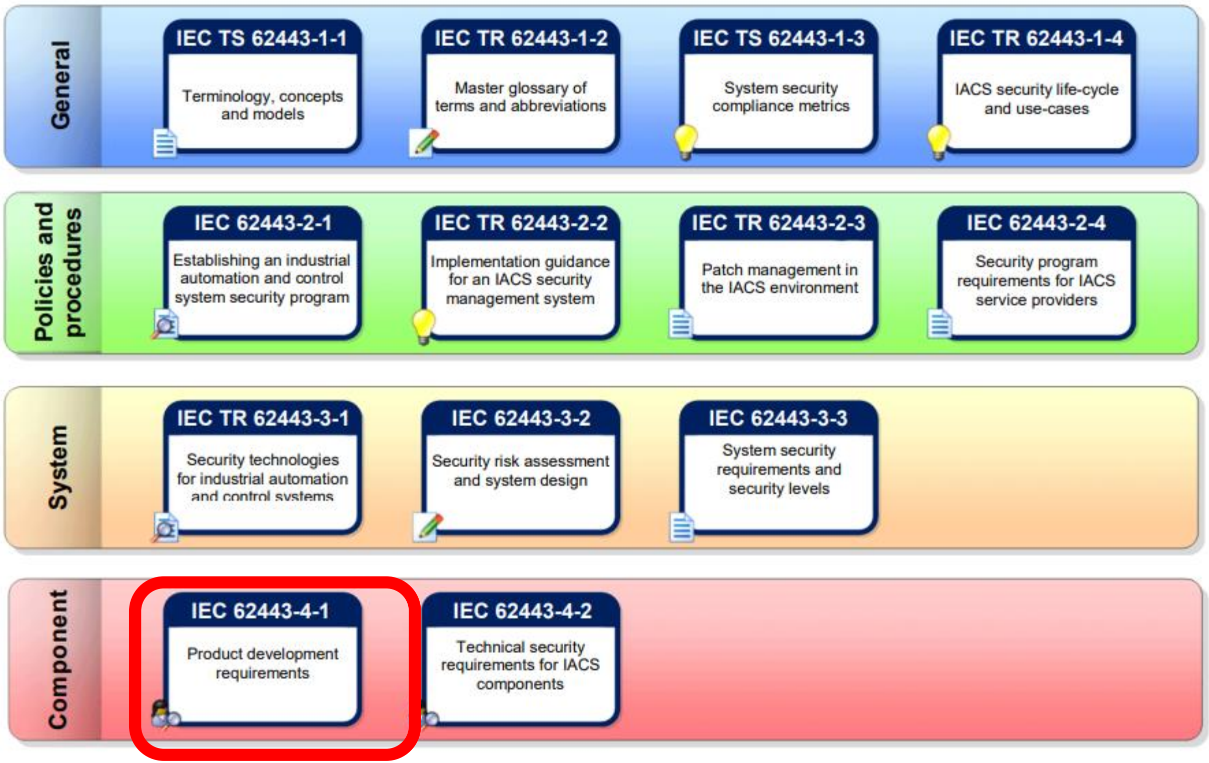
Ivan Lyukshin

Head of Technology Solutions and Endpoint Detection and Response Development

kaspersky

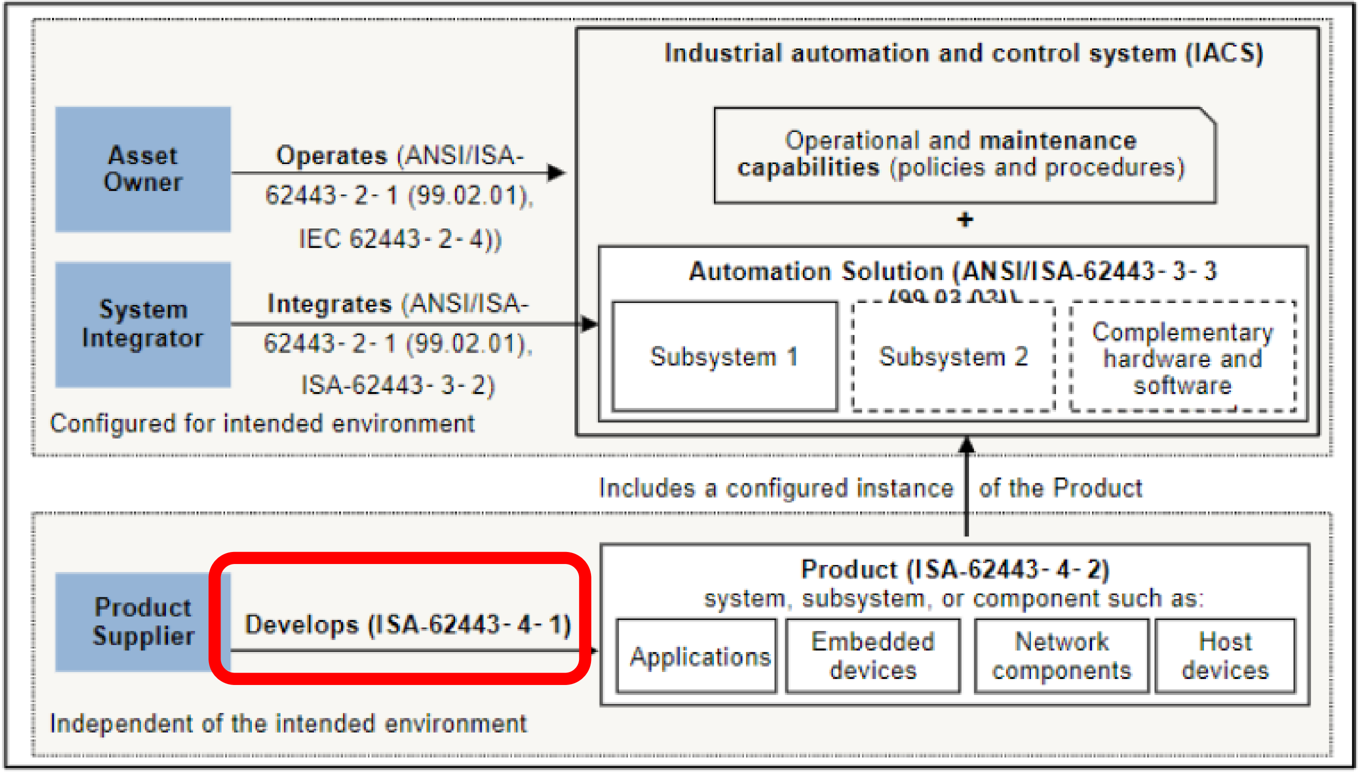
What is IEC 62443-4-1

IEC 62443:2018. Part 4-1: Secure product development lifecycle requirements



General	IEC TS 62443-1-1 Terminology, concepts and models	IEC TR 62443-1-2 Master glossary of terms and abbreviations	IEC TS 62443-1-3 System security compliance metrics	IEC TR 62443-1-4 IACS security life-cycle and use-cases
Policies and procedures	IEC 62443-2-1 Establishing an industrial automation and control system security program	IEC TR 62443-2-2 Implementation guidance for an IACS security management system	IEC TR 62443-2-3 Patch management in the IACS environment	IEC 62443-2-4 Security program requirements for IACS service providers
System	IEC TR 62443-3-1 Security technologies for industrial automation and control systems	IEC 62443-3-2 Security risk assessment and system design	IEC 62443-3-3 System security requirements and security levels	
Component	IEC 62443-4-1 Product development requirements	IEC 62443-4-2 Technical security requirements for IACS components		

What is IEC 62443-4-1

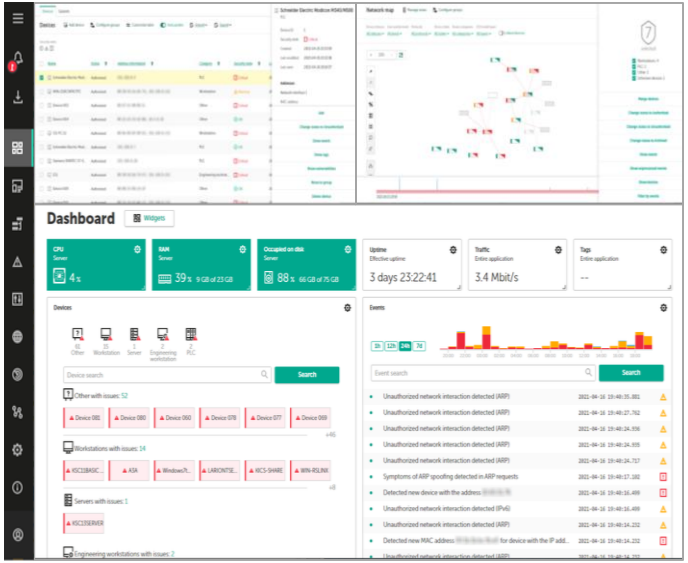

Industrial automation and control system (IACS)
Asset Owner
Operates (ANSI/ISA-62443-2-1 (99.02.01), IEC 62443-2-4))
Operational and maintenance capabilities (policies and procedures)
+
System Integrator
Integrates (ANSI/ISA-62443-2-1 (99.02.01), ISA-62443-3-2)
Automation Solution (ANSI/ISA-62443-3-3 (99.03.03))
Subsystem 1
Subsystem 2
Complementary hardware and software
Configured for intended environment
Includes a configured instance of the Product
Product Supplier
Develops (ISA-62443-4-1)
Product (ISA-62443-4-2)
system, subsystem, or component such as:
Applications
Embedded devices
Network components
Host devices
Independent of the intended environment

IEC 62443-4-1 scope.

- Development lifecycle & Secure development lifecycle

  for:

- KICS for networks



Dashboard

CPU Server 4%

RAM Server 39% 9 GB of 23 GB

Occupied on disk Server 88% 66 GB of 75 GB

Uptime Effective uptime 3 days 23:22:41

Traffic Entire application 3.4 Mbit/s

Tags Entire application --

Devices

Events

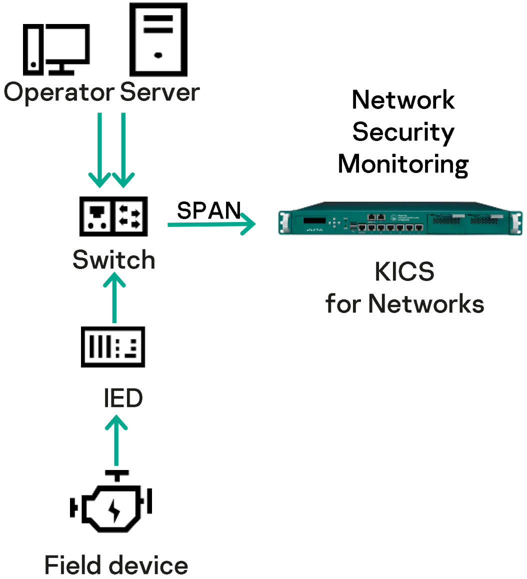
Unauthorized network interaction detected (ARP)

Symptoms of ARP spoofing detected in ARP requests

Detected new device with the address

Unauthorized network interaction detected (IPv6)

Detected new MAC address for device with the IP add...



OT Intrusion Detection

Ability to detect APTs on the lowest level (ICS Protocols DPI and specific signatures)

Asset Inventory

Passive detection of OT components and their communications

OT Risk Management

Vulnerability, Network Configuration and other risks visualization

IEC 62443-4-1. Structure

- Practices:
    - Security management
    - Specification of security requirements
    - Secure by design
    - Secure implementation
    - Security verification and validation testing
    - Management of security-related issues
    - Security update management
    - Security guidelines

IEC 62443-4-1 certification project team

kaspersky

Kaspersky certification team

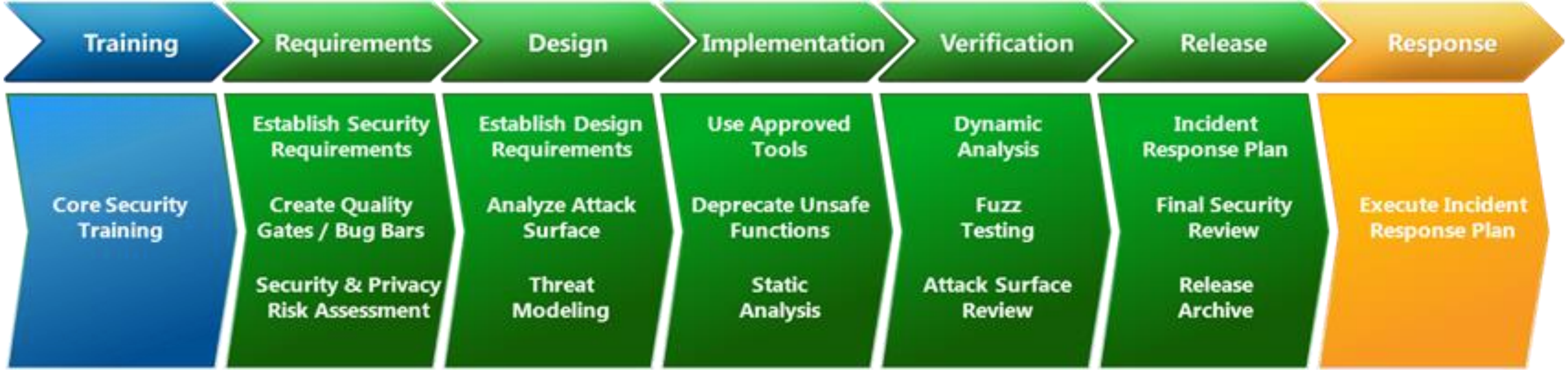
- Product managers
- Certification team
- Development team
    - Project manager
    - Architect / Security champion
    - Test Manager
- Product security team
- Business process managers
- Information security team
- Doc & loc team

TÜV AUSTRIA

TUV Austria auditors team

- Project manager
- Auditor Team Lead
- Auditor / Tech expert

SDLC: Secure development



Training
- Core Security Training

Requirements
- Establish Security Requirements
- Create Quality Gates / Bug Bars
- Security & Privacy Risk Assessment

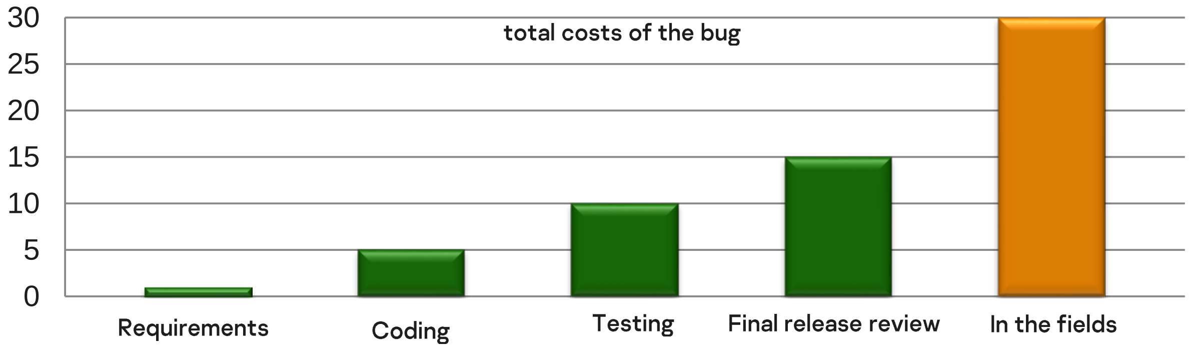
Design
- Establish Design Requirements
- Analyze Attack Surface
- Threat Modeling

Implementation
- Use Approved Tools
- Deprecate Unsafe Functions
- Static Analysis

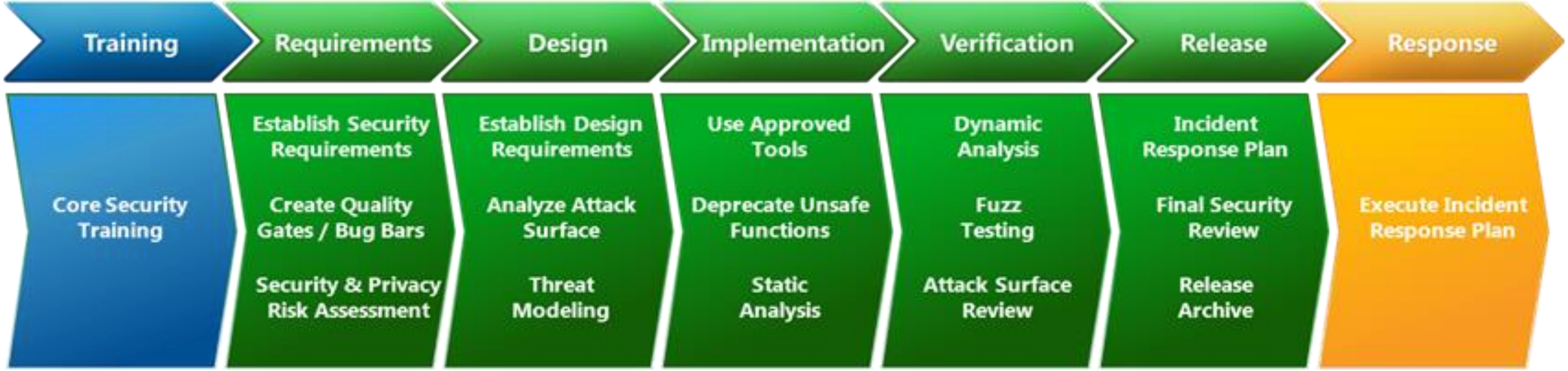
Verification
- Dynamic Analysis
- Fuzz Testing
- Attack Surface Review

Release
- Incident Response Plan
- Final Security Review
- Release Archive

Response
- Execute Incident Response Plan



SDLC & IEC 62443-4-1



Training	Requirements	Design	Implementation	Verification	Release	Response
Core Security Training	Establish Security Requirements Create Quality Gates / Bug Bars Security & Privacy Risk Assessment	Establish Design Requirements Analyze Attack Surface Threat Modeling	Use Approved Tools Deprecate Unsafe Functions Static Analysis	Dynamic Analysis Fuzz Testing Attack Surface Review	Incident Response Plan Final Security Review Release Archive	Execute Incident Response Plan

Specification of security requirements

Secure by design

Secure implementation

Security verification and validation testing

Management of security-related issues

Security management

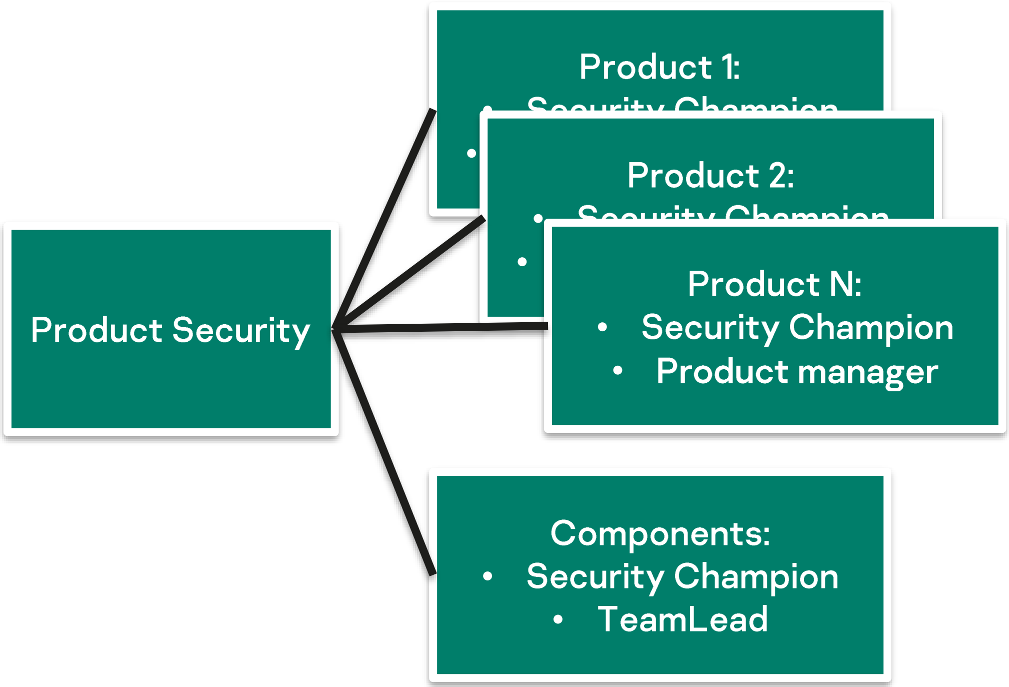
Kaspersky SDLC. Documents

- SM-1: Development process

Topic	Status
Pentest instruction	APPROVED
Secure code review procedure	APPROVED (SHAREPOINT)
Secure coding guideline (checklist)	APPROVED
Threat Modelling procedure	APPROVED
Security Champion instruction for threat modelling	APPROVED
Fuzzing instruction	APPROVED
Vulnerability management procedure	APPROVED
SDL process overview	APPROVED (SHAREPOINT)
Static analysis procedure	APPROVED
Dynamic analysis procedure	APPROVED
3rd party libs using procedure	APPROVED

Kaspersky SDLC: Roles

- SM-2: Identification of responsibilities



Product Security

Product 1:
- Security Champion

Product 2:
- Security Champion

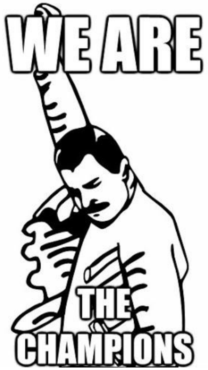
Product N:
- Security Champion
- Product manager

Components:
- Security Champion
- TeamLead

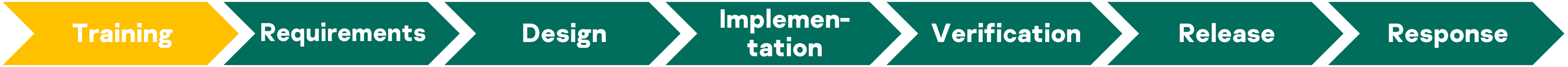
- Security champions inside dev teams
- Product managers
- Development teams
- Q&A
- Product Security

Kaspersky SDLC: Roles

- Development team
  - Product manager
  - Project manager
  - Architect
  - Developers
  - Testers
  - Security champion
- Product security team



SDLC. Education



Training

Requirements

Design

Implementation

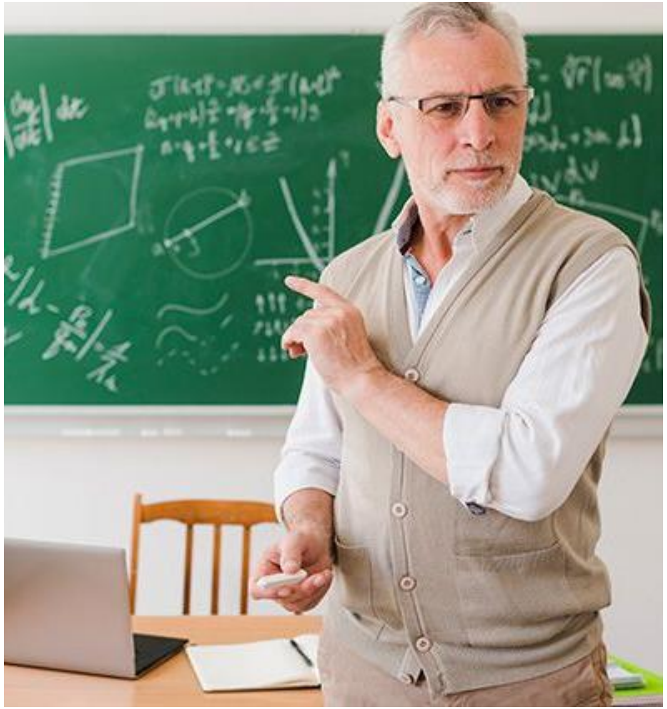
Verification

Release

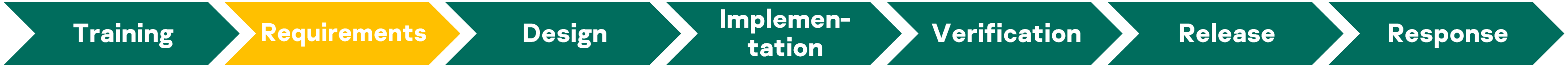
Response

- SM-1: Development process

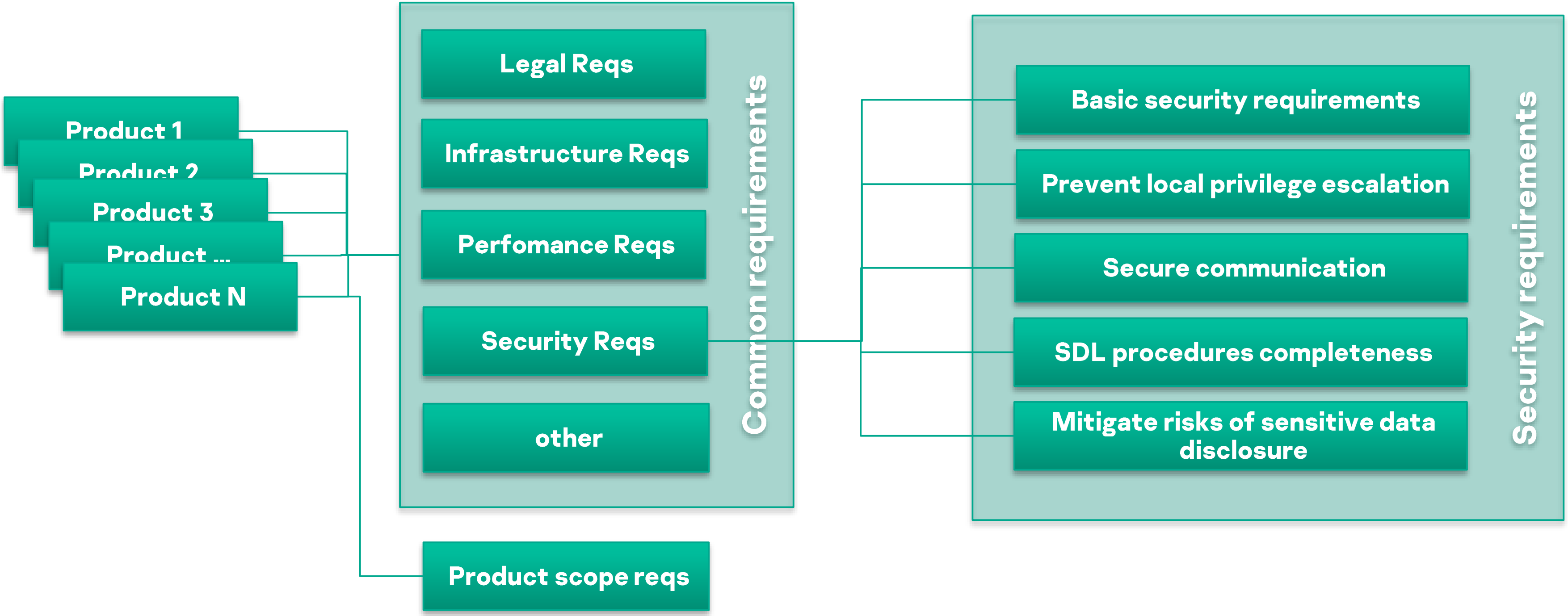
	Cource	Role
1	SDL intro	NewComers, Developers, architects, tech leads
2	[SDL] Secure coding C#	Developers, architects, tech leads
3	[SDL] Threat Modelling	Security champions
4	[SDL] Threat Modelling Automation	Security champions
5	[SDL] Secure coding, part 1	Developers, architects, tech leads
6	[SDL] Secure coding, part 2	Developers, architects, tech leads
7	[SDL] Secure coding for Linux	Developers, architects, tech leads
8	[SDL] Fuzzing	Security champions
9	[SDL] Encryption	Developers, architects, tech leads



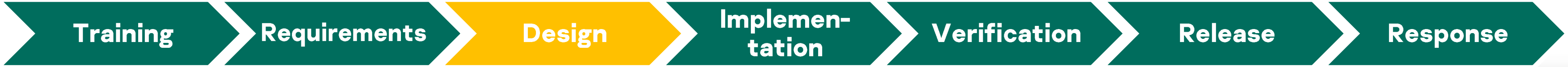
SDLC. Common requirements



- SM-1: Development process (security requirements definition)



SDLC. Threat modelling



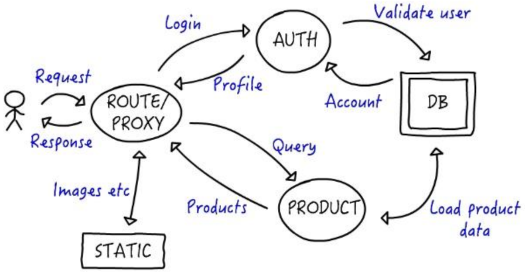
Training → Requirements → Design → Implementation → Verification → Release → Response

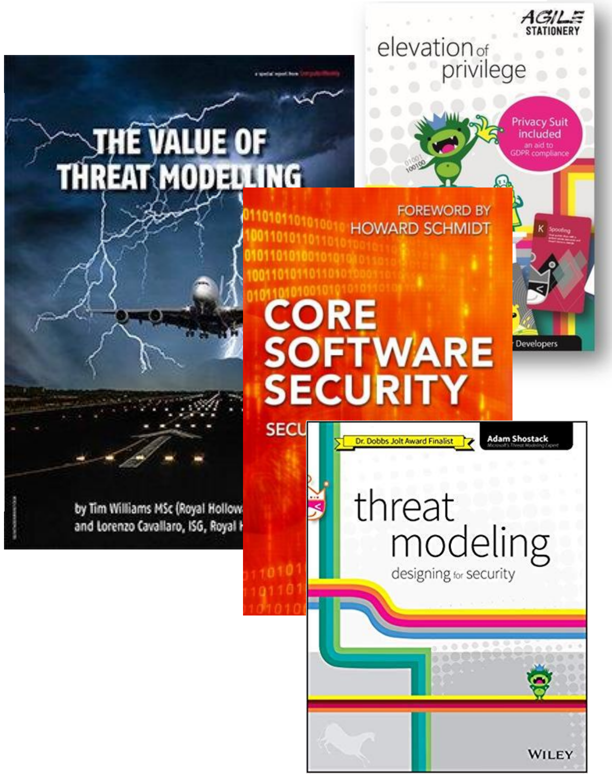
- SM-1: Development process (secure design)

Methodologies:

1. STRIDE
2. DREAD
3. PASTA
4. Kill Chain
5. OWASP
6. ..... mixed?

Data Flow Diagram

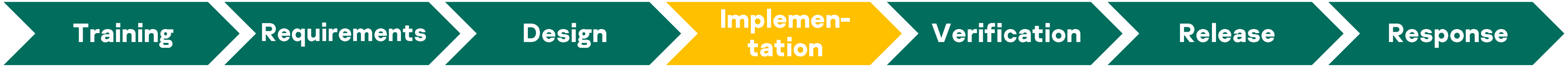




«Threat modelling guru» recomm

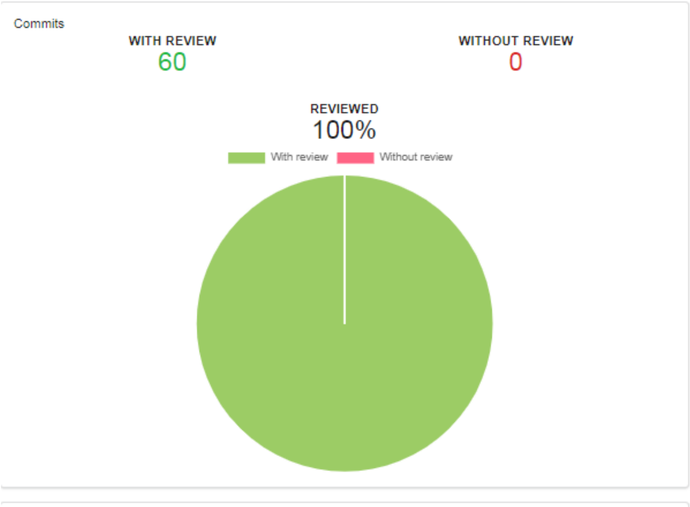
1. First of all threat modelling for main modules
2. Result documentation, store all artefacts
3. To draw diagrams, pictures, schemas. Ideally – architecture.
   Tools: paper, whiteboard, software tools
4. «What if…» questions
5. Risk identification and prioritization

SDLC. Code review


Training
Requirements
Design
Implementation
Verification
Release
Response

- Secure implementation (including coding guidelines)

Code coverage report

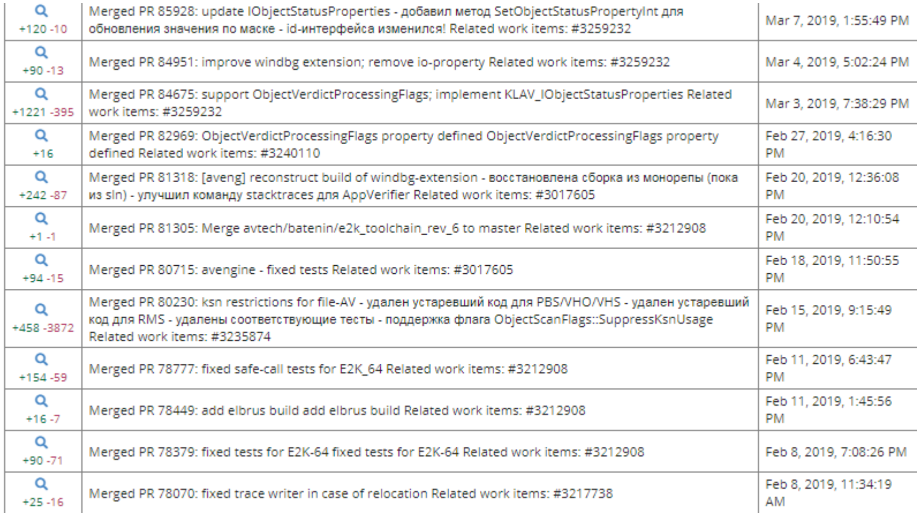

Commits

WITH REVIEW
60

WITHOUT REVIEW
0

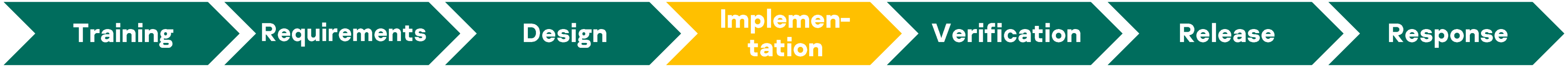
REVIEWED
100%

Secure coding guideline (checklist)



🔍 +120 -10	Merged PR 85928: update IObjectStatusProperties - добавил метод SetObjectStatusPropertyInt для обновления значения по маске - id-интерфейса изменился! Related work items: #3259232	Mar 7, 2019, 1:55:49 PM
🔍 +90 -13	Merged PR 84951: improve windbg extension; remove io-property Related work items: #3259232	Mar 4, 2019, 5:02:24 PM
🔍 +1221 -395	Merged PR 84675: support ObjectVerdictProcessingFlags; implement KLAV_IObjectStatusProperties Related work items: #3259232	Mar 3, 2019, 7:38:29 PM
🔍 +16	Merged PR 82969: ObjectVerdictProcessingFlags property defined ObjectVerdictProcessingFlags property defined Related work items: #3240110	Feb 27, 2019, 4:16:30 PM
🔍 +242 -87	Merged PR 81318: [aveng] reconstruct build of windbg-extension - восстановлена сборка из монорепы (пока из sln) - улучшил команду stacktraces для AppVerifier Related work items: #3017605	Feb 20, 2019, 12:36:08 PM
🔍 +1 -1	Merged PR 81305: Merge avtech/batenin/e2k_toolchain_rev_6 to master Related work items: #3212908	Feb 20, 2019, 12:10:54 PM
🔍 +94 -15	Merged PR 80715: avengine - fixed tests Related work items: #3017605	Feb 18, 2019, 11:50:55 PM
🔍 +458 -3872	Merged PR 80230: ksn restrictions for file-AV - удален устаревший код для PBS/VHO/VHS - удален устаревший код для RMS - удалены соответствующие тесты - поддержка флага ObjectScanFlags::SuppressKsnUsage Related work items: #3235874	Feb 15, 2019, 9:15:49 PM
🔍 +154 -59	Merged PR 78777: fixed safe-call tests for E2K_64 Related work items: #3212908	Feb 11, 2019, 6:43:47 PM
🔍 +16 -7	Merged PR 78449: add elbrus build add elbrus build Related work items: #3212908	Feb 11, 2019, 1:45:56 PM
🔍 +90 -71	Merged PR 78379: fixed tests for E2K-64 fixed tests for E2K-64 Related work items: #3212908	Feb 8, 2019, 7:08:26 PM
🔍 +25 -16	Merged PR 78070: fixed trace writer in case of relocation Related work items: #3217738	Feb 8, 2019, 11:34:19 AM

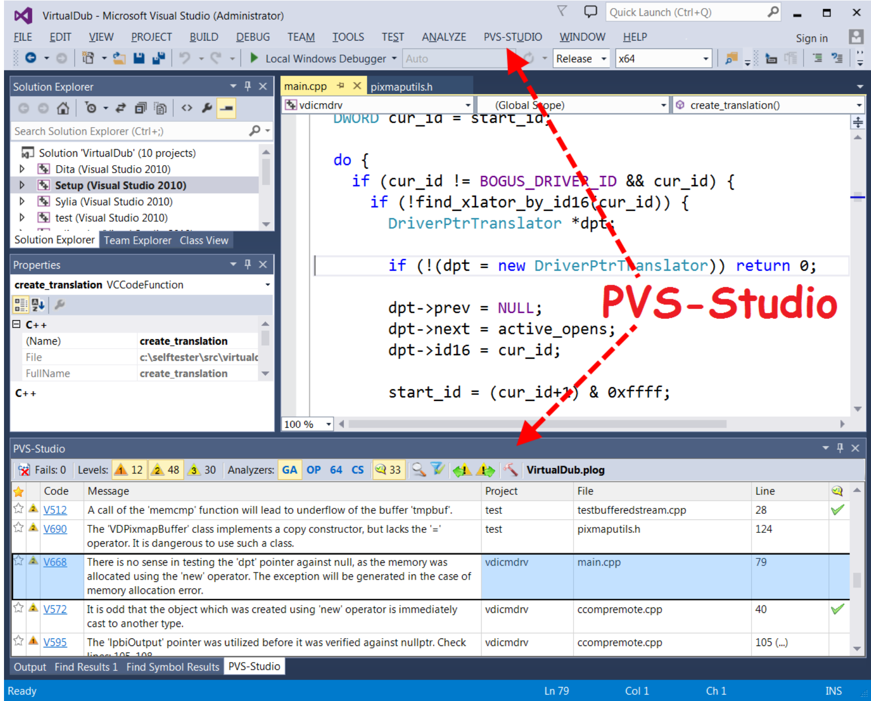
SDLC. Tools. SAST


Training
Requirements
Design
Implementation
Verification
Release
Response

- Security verification and validation testing

1. **Use SAST tools**
   ▶ **For any language**
2. **Approved SAST configs**
3. **All code commits should be tested by SAST**

Tools:
Clang, PVS, Clang-tidy, SVACE, tslint, pylint .....


VirtualDub - Microsoft Visual Studio (Administrator)
Quick Launch (Ctrl+Q)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE PVS-STUDIO WINDOW HELP
Sign in
Local Windows Debugger
Auto
Release
x64
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'VirtualDub' (10 projects)
Dita (Visual Studio 2010)
Setup (Visual Studio 2010)
Sylia (Visual Studio 2010)
test (Visual Studio 2010)
Solution Explorer
Team Explorer
Class View
Properties
main.cpp
pixmaputils.h
vdicmdrv
(Global Scope)
create_translation()
100 %
PVS-Studio
Output
Find Results 1
Find Symbol Results
PVS-Studio
Ready
Ln 79
Col 1
Ch 1
INS

create_translation VCCodeFunction

C++

(Name)	create_translation
File	c:\selftester\src\virtualc
FullName	create_translation

C++

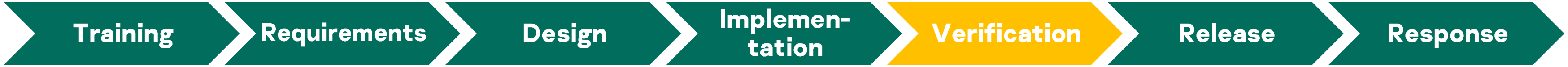
```
DWORD cur_id = start_id;

do {
  if (cur_id != BOGUS_DRIVER_ID && cur_id) {
    if (!find_xlator_by_id16(cur_id)) {
      DriverPtrTranslator *dpt;

      if (!(dpt = new DriverPtrTranslator)) return 0;

      dpt->prev = NULL;
      dpt->next = active_opens;
      dpt->id16 = cur_id;

      start_id = (cur_id+1) & 0xffff;
```

PVS-Studio

Fails: 0 Levels: 12 48 30 Analyzers: GA OP 64 CS 33 VirtualDub.plog

Code	Message	Project	File	Line
V512	A call of the 'memcmp' function will lead to underflow of the buffer 'tmpbuf'.	test	testbufferedstream.cpp	28
V690	The 'VDPixmapBuffer' class implements a copy constructor, but lacks the '=' operator. It is dangerous to use such a class.	test	pixmaputils.h	124
V668	There is no sense in testing the 'dpt' pointer against null, as the memory was allocated using the 'new' operator. The exception will be generated in the case of memory allocation error.	vdicmdrv	main.cpp	79
V572	It is odd that the object which was created using 'new' operator is immediately cast to another type.	vdicmdrv	ccompremote.cpp	40
V595	The 'lpbiOutput' pointer was utilized before it was verified against nullptr. Check lines: 105, 108	vdicmdrv	ccompremote.cpp	105 (...)

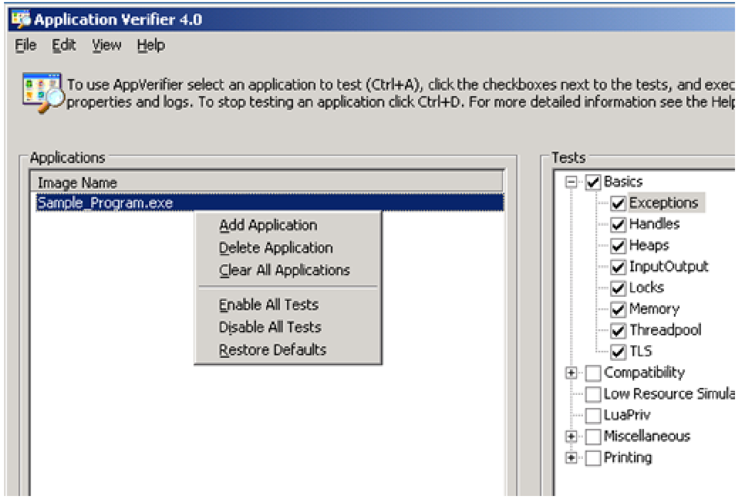
SDLC. Tools. Dynamic analysis



- Security verification and validation testing

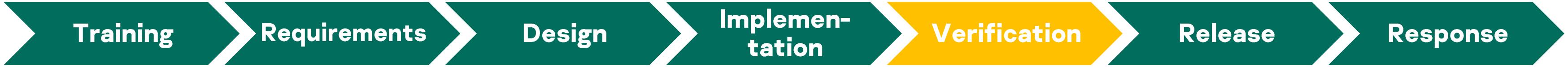
Approved sanitizer tools configurations

- Application Verifier (Win)
- Driver Verifier (Win)
- Dr.Memory/compiler specific
- Clang
- Sanitizers (asan,tsan,ubsan)
- Valgrind
  - Out-of-bounds accesses to heap, stack and global
  - Use-after-free
  - Use-after-return
  - Double-free, invalid free
  - Memory leaks
- **Kaspersky Product Security Requirement Verifier**



To use AppVerifier select an application to test (Ctrl+A), click the checkboxes next to the tests, and exec properties and logs. To stop testing an application click Ctrl+D. For more detailed information see the Help

SDLC. Tools. Dynamic analysis



Training → Requirements → Design → Implementation → Verification → Release → Response

- Security verification and validation testing

Product Security Requirement Verifier

DLL Hijacking (win):

- DLL is loaded by short or invalid path
- DLL is loaded by unsafe path
- Resource is accessed by path with weak ACL

Data corruption (Write to external folder under privileged user**)**

PE header checking:

- module does not support DEP
- module does not support ASLR
- module does not support isolation
- module does not support GS

Other:

- Write & Execute memory detected
- Using insecure protocol detected
- In process loaded unsigned module

Ooops. Covid19 restrictions & home office

Training

Requirements

Design

Implementation

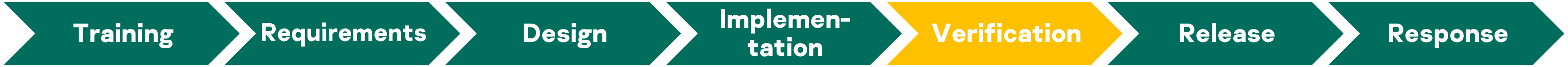
Verification

Release

Response

Auditor’s dog made our day :)

SDLC. Tools. Functional testing



Training

Requirements

Design

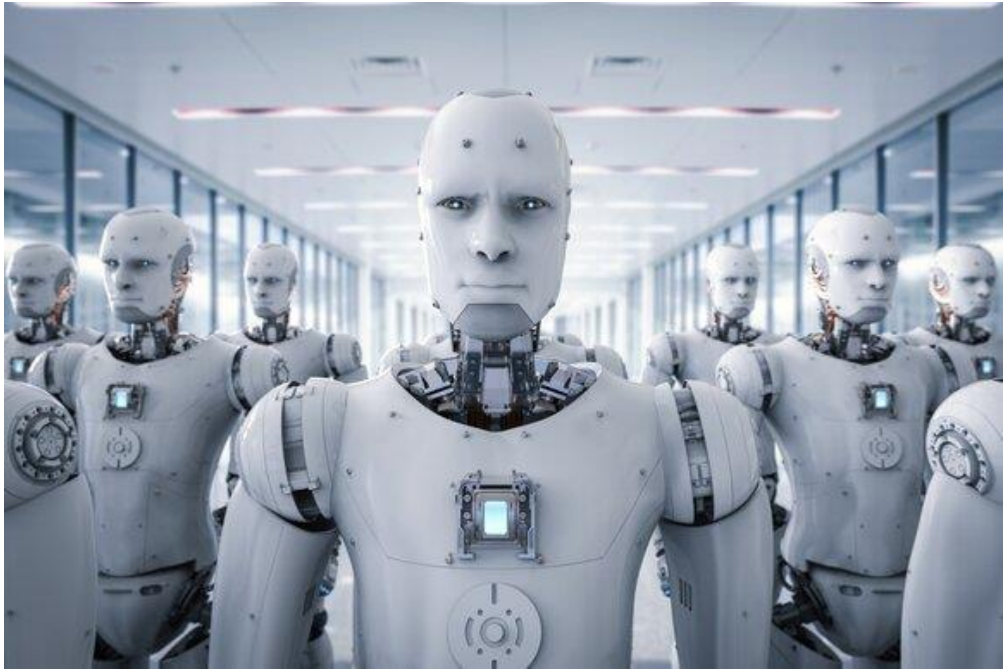
Implementation

Verification

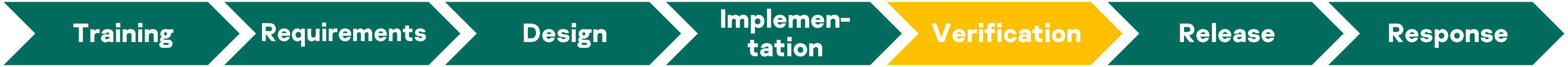
Release

Response

- Security verification and validation testing



SDLC. Tools. Fuzzing testing



Training → Requirements → Design → Implementation → Verification → Release → Response

- Security verification and validation testing

  - Fuzz tests criteria:
    - Coverage
    - Fuzz time
  - Fuzz farm: own virtual infra

  - Fuzz tests - form development teams
  - Fuzz farm – product security
  - Fuzz bugs – product security

Fuzzing with AFL

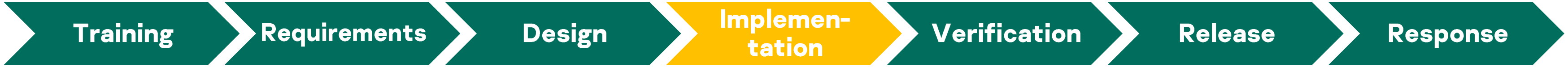
```
==16345== ERROR: AddressSanitizer: unknown-crash on address 0x60820001220b at pc 0x7f4a98de23f7 bp 0x7fffa0b72380 sp 0x7fffa0b71b40
READ of size 25600 at 0x60820001220b thread T0
    #0 0x7f4a98de23f6 (/usr/lib/x86_64-linux-gnu/libasan.so.0+0xe3f6)
    #1 0x410dad in memcpy /usr/include/x86_64-linux-gnu/bits/string3.h:51
    #2 0x410dad [illegible] process_heartbeat /h[illegible]shadowsword/HEARTBLEED/openssl[illegible]_lib.c:258[illegible]
    #3 0x49d16c [illegible]
    #4 0x4a0ca3 [illegible]
    #5 0x465a2d [illegible]
    #6 0x4764b3 [illegible]
    #7 0x402487 in main /home/shadowsword/HEARTBLEED/ope[illegible].1f/selftls.c:5[illegible]
    #8 0x7f4a9882cec4 (/lib/x86_64-linux-gnu/libc.so.6+0x21ec4)
    #9 0x40299c in _start (/home/shadowsword/HEARTBLEED/openssl-1.0.1f/selftls+0x40299c)
0x608200016748 is located 0 bytes to the right of 17736-byte region [0x608200012200,0x608200016748)
allocated by thread T0 here:
    #0 0x7f4a98de941a (/usr/lib/x86_64-linux-gnu/libasan.so.0+0x1541a)
    #1 0x4d7946 in CRYPTO_malloc /home/shadowsword/HEARTBLEED/openssl-1.0.1f/crypto/mem.c:308
```

Metalnem/
sharpfuzz

AFL-based fuzz testing for .NET

Fuzzing with libFuzzer

SDLC. 3-rd party libs. Composition analysis



- SM-9: Security requirements for externally provided components
- SM-10: Custom developed components from third-party

Product Security Team:

1. Daily check for public CVE
2. Prioritization
3. Change requests for 3rd party libs update

Development teams:

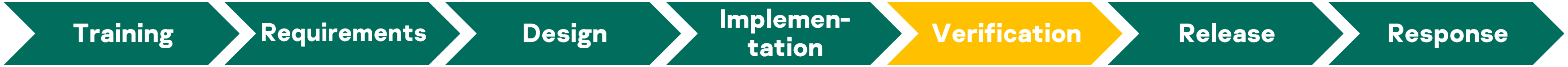
System	Surce
GitHub	https://github.com/
npm	https://www.npmjs.com/
yarn	https://yarnpkg.com/
Maven	https://maven.apache.org/
PyPI	https://pypi.org/
NuGet	https://www.nuget.org/
Rubygems	https://rubygems.org/

open source

Common Vulnerabilities and Exposures

NVD

SDLC. Penetration testing



Training → Requirements → Design → Implementation → Verification → Release → Response

- SM-11: Assessing and addressing security-related issues

•Pentest top rated Threat modelling risks

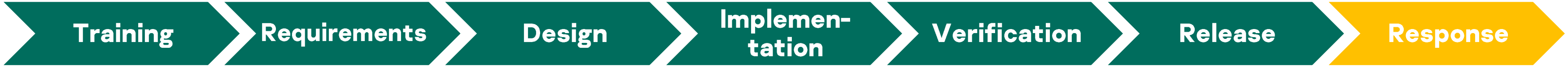
•Pentest network API

•Pentest «something strange»

•KOS pentest

•Pentest from several independent teams

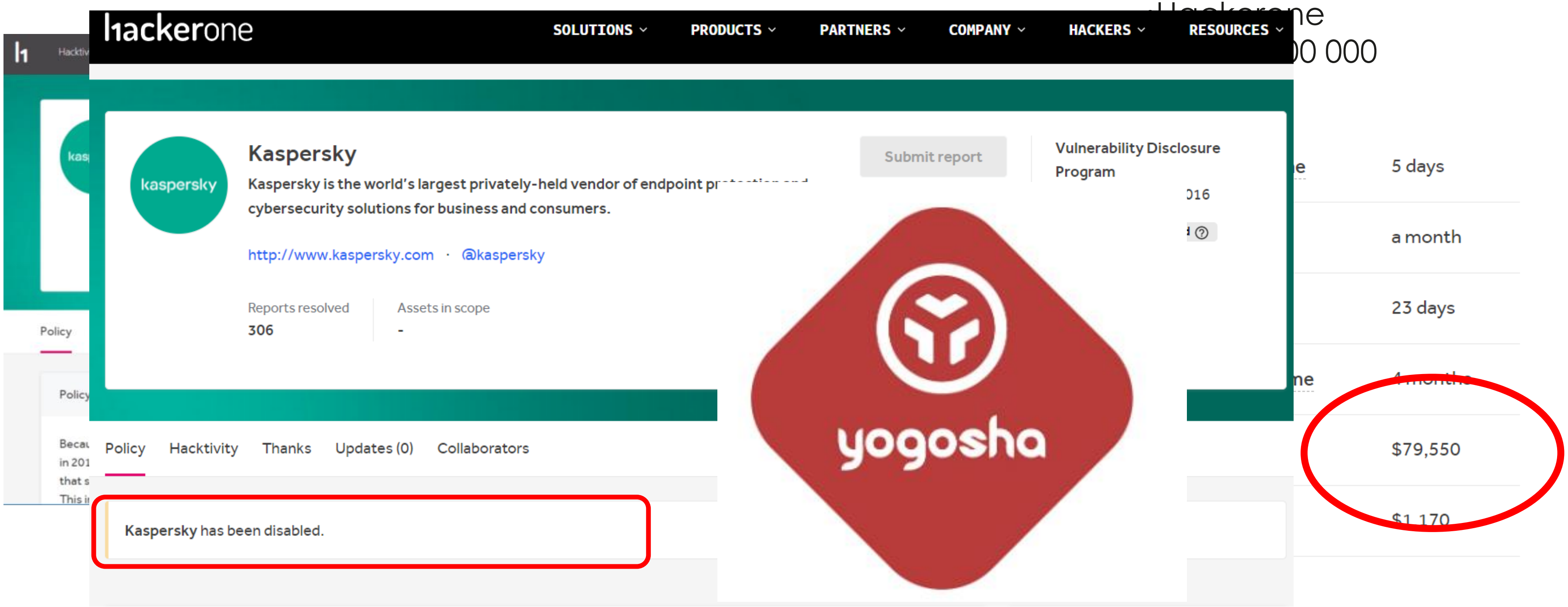
SDLC. Public vulnerabilities



Training → Requirements → Design → Implementation → Verification → Release → Response

- SM-11: Assessing and addressing security-related issues

•BugBounty
•Hackerone
00 000



Kaspersky

Kaspersky is the world's largest privately-held vendor of endpoint pr... cybersecurity solutions for business and consumers.

http://www.kaspersky.com · @kaspersky

Vulnerability Disclosure Program

016

Reports resolved 306

Assets in scope -

Policy | Hacktivity | Thanks | Updates (0) | Collaborators

Kaspersky has been disabled.

Policy

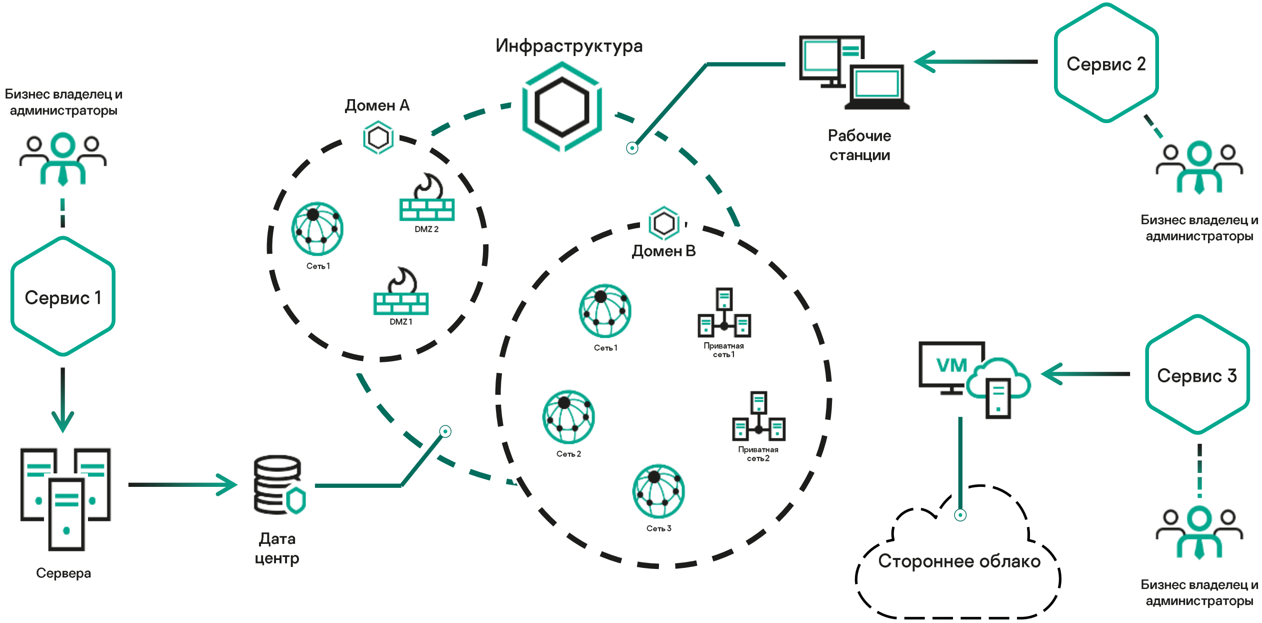
Becau in 201 that s This i

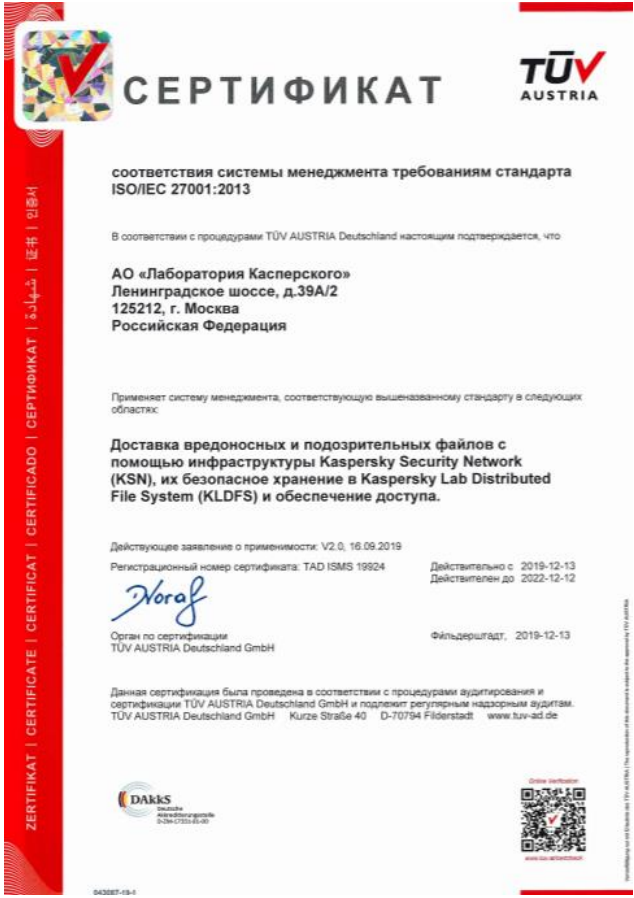
e	5 days
	a month
	23 days
ne	4 months
	$79,550
	$1,170

Infrastructure security

- SM-7: Development environment security

«Classical» information security controls





ZERTIFIKAT | CERTIFICATE | CERTIFICADO | СЕРТИФИКАТ | شهادة | 证书 | 인증서

СЕРТИФИКАТ

TÜV AUSTRIA

соответствия системы менеджмента требованиям стандарта ISO/IEC 27001:2013

В соответствии с процедурами TÜV AUSTRIA Deutschland настоящим подтверждается, что

**АО «Лаборатория Касперского»
Ленинградское шоссе, д.39А/2
125212, г. Москва
Российская Федерация**

Применяет систему менеджмента, соответствующую вышеназванному стандарту в следующих областях:

Доставка вредоносных и подозрительных файлов с помощью инфраструктуры Kaspersky Security Network (KSN), их безопасное хранение в Kaspersky Lab Distributed File System (KLDFS) и обеспечение доступа.

Действующее заявление о применимости: V2.0, 16.09.2019

Регистрационный номер сертификата: TAD ISMS 19924

Действительно с 2019-12-13
Действителен до 2022-12-12

Noraf

Орган по сертификации
TÜV AUSTRIA Deutschland GmbH

Фильдерштадт, 2019-12-13

Данная сертификация была проведена в соответствии с процедурами аудитирования и сертификации TÜV AUSTRIA Deutschland GmbH и подлежит регулярным надзорным аудитам.
TÜV AUSTRIA Deutschland GmbH Kurze Straße 40 D-70794 Filderstadt www.tuv-ad.de

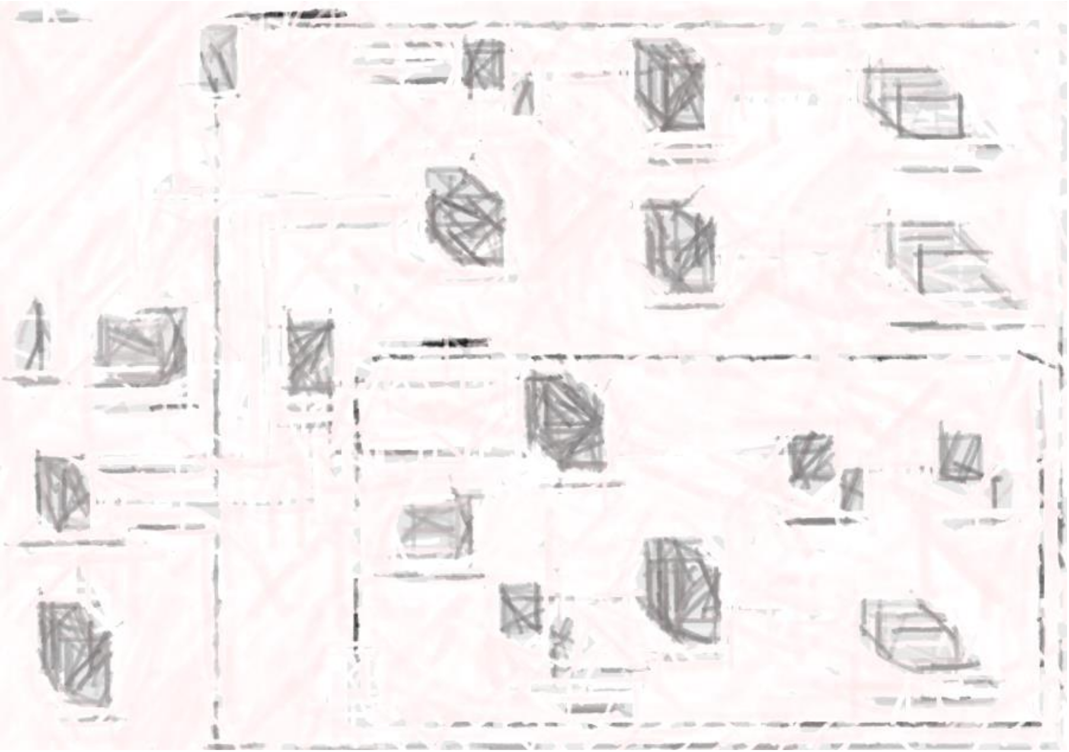
DAkkS
Deutsche Akkreditierungsstelle
D-ZM-17331-01-00

Online Verification

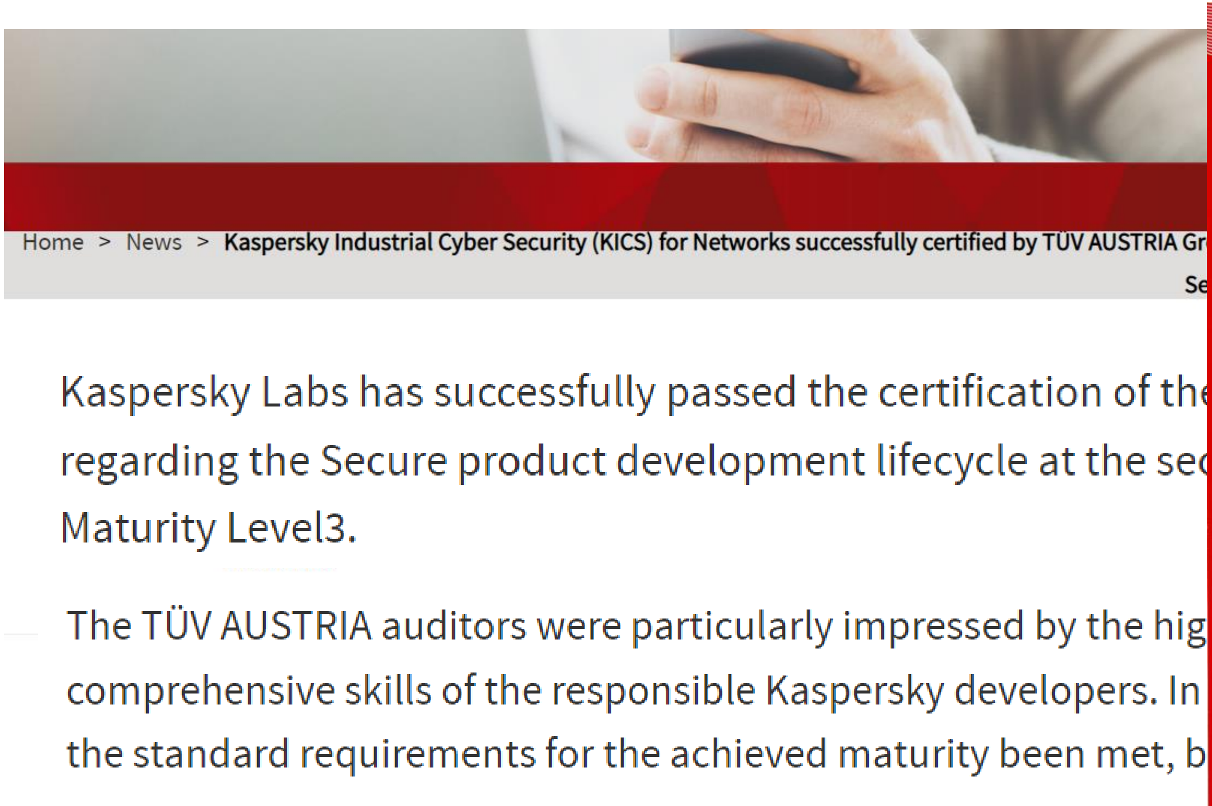
043087-19-I

Infrastructure security

- SM-8: Controls for private keys



SDLC. Security certification

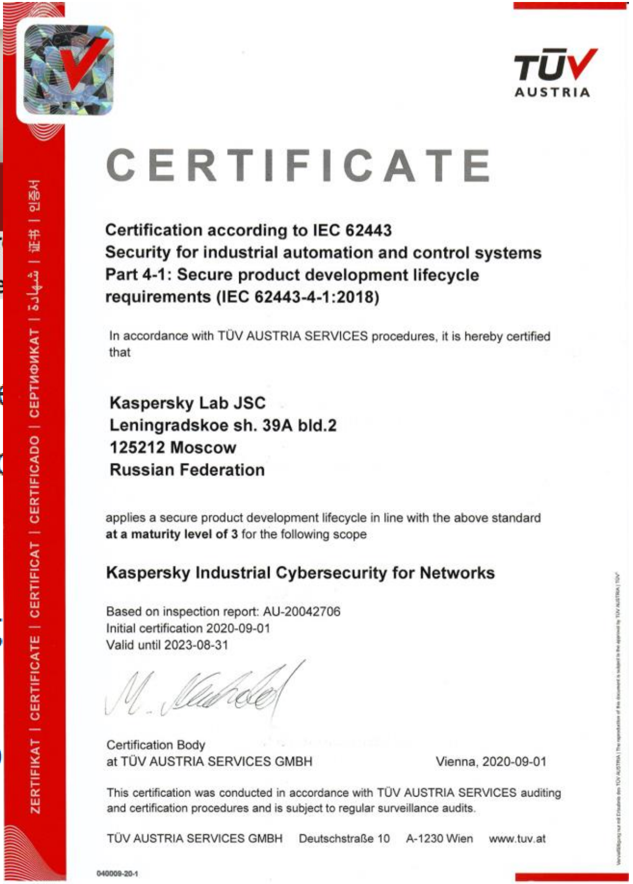


Home > News > Kaspersky Industrial Cyber Security (KICS) for Networks successfully certified by TÜV AUSTRIA Gr

Se

Kaspersky Labs has successfully passed the certification of the regarding the Secure product development lifecycle at the se Maturity Level3.

The TÜV AUSTRIA auditors were particularly impressed by the hig comprehensive skills of the responsible Kaspersky developers. In the standard requirements for the achieved maturity been met, b



ZERTIFIKAT | CERTIFICATE | CERTIFICADO | СЕРТИФИКАТ | شهادة | 证书 | 인증서

TÜV AUSTRIA

CERTIFICATE

Certification according to IEC 62443
Security for industrial automation and control systems
Part 4-1: Secure product development lifecycle
requirements (IEC 62443-4-1:2018)

In accordance with TÜV AUSTRIA SERVICES procedures, it is hereby certified that

Kaspersky Lab JSC
Leningradskoe sh. 39A bld.2
125212 Moscow
Russian Federation

applies a secure product development lifecycle in line with the above standard **at a maturity level of 3** for the following scope

Kaspersky Industrial Cybersecurity for Networks

Based on inspection report: AU-20042706
Initial certification 2020-09-01
Valid until 2023-08-31

Certification Body
at TÜV AUSTRIA SERVICES GMBH

Vienna, 2020-09-01

This certification was conducted in accordance with TÜV AUSTRIA SERVICES auditing and certification procedures and is subject to regular surveillance audits.

TÜV AUSTRIA SERVICES GMBH Deutschstraße 10 A-1230 Wien www.tuv.at

040009-20-1

Thank you!

SECURITY IS EVERYONE'S RESPONSIBILITY

Dmitry Shmoylov
Head of Software security

Ivan Lyukshin
Head of Technology Solutions and Endpoint
Detection and Response Development

kaspersky