ECSC

CURRICULA

JULY 2021

ABOUT ECSC

The growing need for IT security professionals is widely acknowledged worldwide. To help mitigate this shortage of skills, many countries launched national cybersecurity competitions targeting towards students, university graduates or even non-ICT professionals with a clear aim to find new and young cyber talents and encourage young people to pursue a career in cyber security. The European Cyber Security Challenge (ECSC) leverages on these competitions by adding a pan-European layer.

The European Cyber Security Challenge is an initiative by the European Union Agency for Cybersecurity (ENISA) and aims at enhancing cybersecurity talent across Europe and connecting high potentials with industry leading organizations.

CONTACT

For contacting the authors please use ecsc@enisa.europa.eu
For media enquiries about this paper, please use press@enisa.europa.eu.

AUTHORS/ACKNOWLEDGEMENTS

ENISA, European Union Agency for Cybersecurity

LEGAL NOTICE

COPYRIGHT NOTICE

TABLE OF CONTENTS

1. ACKNOWLEDGEMENTS

ENISA would like to acknowledge the important and substantial contribution of key experts in the field that made this proposal possible. Their constructive input and experience will make the European Cyber Security Challenge a more coherent, structured and engaging event. Among these experts, ENISA would like to mention, in particular:

Academia

- Vincent RIJMEN (lead contributor), KU Leuven Professor and co-author of Rijndael, the Advanced Encryption Standard (AES).
- Thorsten STRUFE (lead contributor), Professor for Privacy and IT Security at Technische Universität Dresden.
- Vasilis KATOS (contributor), Professor and Head of Computing at Bournemouth University.

Security researchers

- Miroslav STAMPAR (contributor), security researcher, co-author of the popular sqlmap.

Industry

- Cristina VATAMANU (reviewer), malware analyst, Bitdefender.

ENISA experts

- Razvan GAVRILA (editor and contributor), ENISA security expert in the field of cyber security training and skill development.
- Yonas LEGUESSE (contributor), ENISA security expert in the field of mobile security.
- Apostolos MALATRAS (contributor), ENISA security expert in the field of IoT.
- Adrien OGEE (contributor), ENISA security expert in the field crisis management.

2. THE PROBLEM

The growing need for information security professionals is widely acknowledged at European level [CED]. To help address this shortage of skill, European countries have launched national cyber security competitions for university graduates or even non-ICT professionals with a clear aim: find new and young cyber talents and encourage young people to pursue a career in cyber security [ENIa]. The European Cyber Security Challenge (ECSC), with the support of the European Union Agency for Network and Information Security (ENISA), leverages on these competitions by adding a pan-European layer.

During the third planning meeting of ECSC2017 (June 2017), the Steering Committee of the ECSC has asked ENISA to propose a draft ECSC curriculum for future editions. This document is an answer to this request and aims to tackle and satisfy three key issues that have been identified during the development and planning of previous ECSC editions:

- Align the challenge with the current cyber security needs of various stakeholders, such as public organizations, cyber security service providers, data controllers, academia, etc. This will be achieved by creating an open framework and annual review process of the ECSC curriculum that will allow for the early engagement of the aforementioned stakeholders.
- Participating countries should have a clear understanding prior to each event on what are the team and individual skillsets expected to be assessed during the challenge. By answering this need, the national level coaching process will by streamlined and the whole process will be made more organized and focused in terms of expected costs and efforts.
- The curriculum will be used to guide the design of future challenges, by placing a clear focus on what the implementation requirements are: what should be tested and how it should be tested. In addition, the curriculum will be used as a starting point for defining the assessment scheme and scoring system for future challenges.

Finally, a secondary, yet equally important, objective of this project is to engage a wider audience and increase the level of awareness on the importance of having events such as the ECSC.

3. DEVELOPMENT OF THIS CURRICULUM

The curriculum that we hereby describe aspires to be overly broad, spanning from core competencies to highly specialized knowledge. Evidently, only few experts have all the mentioned know-how and skills and many universities are offering courses that cover only a subset of them. Accordingly, we distinguish between core competencies and extended topics. Knowledge of the core competencies (as covered in Sections 4 and 5.1) is presumed by all participants. Furthermore, we suggest that each year, a selection is made among the additional, extended topics, as described in this curriculum, and the challenges are designed around this selection. The selection shall be communicated to the teams well in advance of the challenge. The team as a whole should be able to cover those topics, i.e., it should not be assumed nor expected that each team member should hold mastery over the entire spectrum.

Accordingly, in the following we first describe the key skills in general domains, their refinement to specific domains with their inherent particularities, and finally the approaches and methodologies covering different core aspects of the IT security fields.

4. KEY SKILLS AND GENERAL DOMAINS

The teams need to understand the general aspects of the security of ICT systems, and different classes of attack vectors and vulnerabilities. These comprise conceptual and systematic vulnerabilities, but also organizational vulnerabilities and human factors. Hence, the curriculum shall not only focus on known exploits and well-known security primitives. The teams also need to understand that insecurity is the direct effect of false assumptions about the users, the system environment, the implementation and deployment, as well as parameters, limitations of idealized models and mathematical properties. Modelling the systems and their environments, as well as systematic threat analyses and pen-testing methodologies are hence required as core competencies. In this context, it will be important to understand different adversarial properties. Identifying different attackers, attacks and strategies by the intention, behaviour, capabilities, and control belongs to the basic preconditions of the teams' expertise.

A main component of the ECSC has to be to foster adversarial thinking: the teams need to be able to and learn to find vulnerabilities, rather than only knowing about common security services.

The competences of the teams shall be based on understanding the conceptual problems that can lead to vulnerabilities in systems: instead of knowing the details of attacks we have seen in the recent past without understanding the conceptual problem that has led to exploitability, the teams should understand the causes, such as input mismatches, risks of misconfiguration, the increased attack surface that arises from exposing devices by Internet connectivity and globally accessible interfaces, human misunderstanding of technical concepts, and common issues in collaborative scenarios.

4.1 INFORMATION / CRYPTO

Cryptography stands at the heart of information security and without it Europe's digital single market cannot exist. For this reason, the attendees of the European Cyber Security Challenge should be familiar with cryptographic primitives and protocols, their usage and weaknesses. The following section presents the key concepts participating teams should be familiar with prior to attending the ECSC final. In this respect, a good reference book is for example [Knu].

Confidentiality/encryption

- Symmetric-key encryption and public-key encryption
- Strong encryption and weak encryption, key length and exhaustive key search, Moore's law
- Stream ciphers, one-time pad, LFSR-based stream ciphers, E0, SNOW-3G, RC4
- Blockciphers, DES, 3-DES, AES
- Blockcipher modes of operation for encryption: CBC, CTR, OFB, security bound for CBC
- One-way functions, Diffie-Hellman, ElGamal encryption, trapdoor oneway functions, RSA
- Hybrid encryption, KEM-DEM, PKCS standards

Integrity/data authentication

- Data authentication versus encryption
- Symmetric-key data authentication (MACs) and public-key data authentication, non-repudiation
- Hash functions: SHA-1, SHA-256, SHA-512, SHA-3

- Collisions, preimages, birthday paradox, exploiting a collision to forge signed code
- Blockcipher mode of operation for authentication: CBC-MAC, CMAC
- Blockcipher mode of operation for authenticated encryption: GCM, CCM, OCB
- Data authentication algorithm based on hash function: HMAC
- One-time MACs: PMAC, Poly1305
- Digital-signature algorithms: RSA, ElGamal, DSA, ECDSA
- Signatures with message recovery, signatures with appendix, encoding, PKCS standards

Identification/entity authentication

- Passwords, password quality, password storage, PAKE
- Challenge-response by means of MAC, by means of digital signature, reflection attack
- Authenticated key agreement

Key agreement

- Nonces, replay, timeliness
- Session keys, forward secrecy
- Trusted Third Parties
- Kerberos
- Diffie-Hellman, person-in-the-middle attack, key authentication
- STS/IKE

4.2 NETWORK

The connectivity of the devices has led to a large attack surface. This is the case both for the devices, which are suddenly globally reachable, but also for the network infrastructure itself. The teams, hence, have to understand the Internet architecture and basic protocols, as well as the basic security assumptions, threats, and protocols that are commonly used today.

Internet architecture and protocols

- Layered model: ISO/OSI, TCP/IP
- General connection technologies: Ethernet, Wi-Fi, Cellular networks
- Basic infrastructure protocols: BGP, DNS, SS7
- Basic application layer protocols: HTTP, FTP, SSH

Network security fundamentals

- Common threats: Eavesdropping, Masquerade, Modification/Loss of information, Forgery, Authorization violation, Sabotage (Denial of Service), Repudiation
- Dolev-Yao adversary, Sniffing, MitM, Spoofing, Distributed attacks, Reflection and Amplification
- Link layer security: 802.1X, PPP, CHAP, PPTP, WEP/WPA, MAC-sec
- IPsec

- Transport layer security: TLS, SSH
- General solution classes: Proof-of-Work, Stateless protocols
- Infrastructure security: BGPsec, EDNS, DNS Cookies, DNSSEC
- Reactive techniques: Firewalls, IDS, Honeypots

Practical network security

- Bluetooth (incl. BLE) security
- RFID and NFC security
- Wireless network security: ZigBee, Z-Wave, Wi-Fi, LPWAN, NB-IoT
- Network administration and security: promiscuous mode, ad-hoc networking, mesh networking, identity management, encryption, authentication, authorization

Suggested reading: [T+14], [Pad+17], [RS16]

4.3 OPERATING SYSTEMS (OS)

The operating system, being the fundamental resource management and abstraction of the machine, is at the core of device security. The teams therefore have to understand the basic principles of the operating systems, as well as the basic security services and security measures implemented and controlled at the OS level.

Operating Systems fundamentals

- Process state, context switching
- Threads, threading and multi-threading
- Memory management and kernel memory management
- File system, access permissions on files and folders (world-writeable folders, setuid executables)
- Network and stock protocols
- Remote file system
- Shells, GUIs and other interfaces
- Power and device management
- Hyper-visors and containers
- Return-oriented programming, ALSR

Access Control fundamentals

- Mandatory Access Control (MAC)
- Discretionary Access Control (DAC)
- Role Based Access Control (RBAC)
- Rule Based Access Control (RBAC)
- Access Control Lists (ACLs)
- Best Practices for Access Control

4.4 ORGANIZATIONAL AND HUMAN FACTORS

Several vulnerabilities arise from organizational issues and human factors and therefore related security implications should be an essential element of the ECSC. Accordingly, the teams shall understand common sources for vulnerabilities and the consequences they produce. These include essential security shortcomings stemming from the human factor, as well as fundamentals of physical security.

Contrary to common belief, security primitives are not the weakest link of systems in most cases, instead the users are. The teams have to therefore understand the basic shortcomings related to human factors and their behaviour (not only for offense but also defence in the ECSC).

Information gathering and reconnaissance

- Common non-technical information gathering techniques (dumpster diving, shoulder surfing, etc.)
- Technology aided information gathering techniques (corporate websites, search engine, public databases, etc.)

Attack vectors

- In person attacks
- Phone social engineering attacks
- Phishing/email attacks
- USB/media drops
- Pretexting via social media (fake social media profiles, etc.)
- Weak passwords (default passwords, common password policies, weak passwords: dictionaries, digital dossier knowledge, etc.)

Suggested reading: [Had10]

5. SPECIFIC DOMAINS

Some application domains are characterized by very specific requirements and common vulnerabilities. While they generally fall into one of the classes described above, we describe the corresponding key skills for the Web, Mobile environments, the IoT, OS specifics, and Privacy at a slightly higher level of detail in the following.

Due to the prevalence of Web vulnerabilities, we consider this section to be part of the core competencies that are assumed. The subsequent subsections cover rather enhanced skills.

5.1 WEB

The Web has evolved from static HTML pages over to dynamic pages and to complex applications that are used through browsers or exposed as APIs to mobile applications.

The necessarily open interface, exposed to the entire Internet offers access to anybody and hence must be treated with particular diligence. Commonalities of LAMP and three-tier architectures cause similar conceptual vulnerabilities that, if discovered, can be exploited at huge numbers of sites.

We distinguish between basic principles in the Web context, conceptual problems that lead to common vulnerabilities, and specific attack strategies and in this respect the required skills expected from ECSC teams are described in what follows.

DevOps principles

- Web application design (e.g., server configuration, directory structure, virtual hosting, robots.txt, sitemap.xml, etc.)
- Web programming principles (e.g., JavaScript, JSON, Ajax, WebSockets, etc.)
- Web application firewall principles (e.g., ModSecurity)
- Encodings (e.g., URL encoding, double encoding, Unicode encoding, etc.)
- Manual inspection and assessment of web traffic (e.g., Burp, HackBar, etc.)
- Sending custom HTTP requests (e.g., curl)
- Low-level sockets programming (e.g., client for custom protocol)
- High-level web automation programming (e.g., Python)

Common conceptual problems

- Broken authentication (passwords, password recovery), missing access control (full/partial)
- Misconfiguration: default settings, default passwords, daemons at escalated privileges, hardening
- Sensitive data exposure, direct object references, path traversal, password transmission and storage
- Session management
- Input validation, injection attacks (SQL, Command injection), cross site scripting, output encoding/escaping, input sanitization

- Reflection, XSRF
- Unpatched systems/known vulnerabilities
- Under-protected APIs, REST security, app sniffing/reverse engineering, REST/API guessing/discovery

Web application assessment and exploitation

- Detection of server entry points (e.g., admin panel)
- Detection of application entry points (e.g., user provided input)
- Basic testing of found entry points (e.g., deliberate invalid input)
- Detection of basic security problems (e.g., database error reports, web server error reports, usage of obsolete plugins, etc.)
- OWASP Testing Guide
- Testing for default credentials (e.g., admin:admin)
- Detection and exploitation of most common web vulnerabilities:
    - SQL injection (SQLi): boolean-based blind, time-based blind, error-based, UNION-query, stacking, bypassing authentication schema
    - Directory traversal
    - File inclusion (FI) - local and remote
    - Cross-site scripting (XSS) - reflective, stored and DOM-based
    - Cross-site request forgery (CSRF)
    - XPath injection
    - Server-side template injection (SSTI)
    - Session management manipulation (e.g., Cookies)
- Usage of automatic vulnerability detection and exploitation tools (e.g., Nmap, Nikto, sqlmap, etc.)
- Detection and identification of security solutions (e.g., WAF)
- Bypassing logical constraints, filters and/or security solutions (e.g., WAF)
- Tools for network traffic sniffing: Wireshark, Nmap, etc.
- SOAP, REST, JavaScript, and Node.js security.
- Communication protocols: MQTT, CORE, CoAP, DTLS, XMPP, IPv6, 6LoWPAN, WebSockets, AMQP
- Security protocols: ACE, oneM2M Security E2E Authentication and Dynamic Authorization, OAuth, OpenID, SAML
- Deep knowledge of OWASP Top 10 vulnerabilities

Suggested reading: [GT16], [BP17], [Lyo09]

5.2 MOBILE

Mobile devices have become full-fledged high-performance computers with permanent connectivity. Their market penetration, depending on the global region, reaches up to almost 100% of the population. They are commonly provisioned with an extensive set of sensors to collect environmental information around their owners and about their owners. The combination of these properties, as well as their fast development in very short development cycles (which inevitably entails vulnerabilities), has made them a prime target for attack. Teams of ESCS have to be familiar with the following primitives in terms of the mobile ecosystem.

Mobile essentials

- Android Architecture
- iOS Architecture

5.3 IOT

Internet of Things (IoT) refers to a cyber-physical ecosystem of devices with sensors and actuators that make use of decision-making mechanisms to digitise and thus facilitate various facets of daily life. At the core of IoT systems and services is data, feeding into a continuous cycle of sensing, decision making, and actions. The pervasive nature of IoT systems nowadays, the fact that they are utilised to digitise legacy infrastructures, potentially critical ones, and the large interconnection of said systems, all contribute to highlighting the significance of IoT security. IoT is a vast field of research, encompassing many domains and continuously evolving, and consequently it is a daunting task to define a curriculum of topics pertinent to IoT security. In the following, we list the fundamental elements of IoT cyber security that the ECSC teams should have mastery of.

IoT fundamentals

- Familiarization with IoT development platforms and boards
- Administration/programming of Raspberry Pi, Arduino, etc.
- Secure configuration of IoT platforms, microcontrollers and SoC
- Handling of General-Purpose Input/Output (GPIO)

Suggested reading: [Mar14], [Mon16], [KKV14]

5.4 SPECIFIC OPERATING SYSTEMS AND HARDWARE SUPPORT

The reach of the Internet has led to increased attacks and cyber incidents, which, among other countermeasures, caused a collection of attempts to design secure operating systems, or at least make some operating systems more secure. Accordingly, ECSC teams should be familiar with relevant knowledge pools and developments and in particular the following topics.

OS principles

- Security Kernel
- Unix security architecture
- Windows security architecture
- MacOS security architecture
- Linux Security Modules architecture
- SELinux design, AppArmor
- Security architecture of hyper-visors and containers

Suggested reading: [And06], [Ale17], [Bac86]

Physical/Hardware security

- Trusted computing platform
- Hardware security modules
- Trusted execution environments (ARM TrustZone, Intel SGX, AMD SEV)
- Memory Protection Extensions, Control Flow Enforcement
- Physically unclonable functions
- Physical Layer Security

5.5 PRIVACY, SURVEILLANCE AND CENSORSHIP

Undoubtedly, privacy aspects have to be considered in the context of the ECSC and they should reflect both offensive and defensive notions. Abusing knowledge of victims can simplify attacks (e.g., through social engineering, spear-phishing, password-guessing). Conversely, the defensive notion of privacy is useful to hide activities, parties, and to overcome denial of service of any sort (e.g., censorship, eclipsing). ECSC teams must be proficient in both.

Offensive

- Eavesdropping, crawling
- Digital dossier aggregation
- Adversarial learning, adversarial machine learning: training data integrity, algorithmic integrity and verification
- Inference attacks from public profiles to “private” attributes

Defensive

- Basic concepts: proxies, onion routing, mixes, DC-Nets, covert traffic
- Network anonymization: TOR, mix cascades
- Anonymous services: anonymous return addresses, hidden services, darknets
- Data anonymization and de-identification: k-anonymity, l-diversity

Suggested reading: [Tro+17], [Bil+09], [KSG13], [G´er17]

5.6 PKI IN PRACTICE

The distribution of credentials represents a special problem in itself, as identification and trust on the Web are hard to establish and initialize. It is still necessary to know who owns certain public key pairs and certificates, and in this respect different Public Key Infrastructures have been developed and partially deployed. The teams of ECSC should have profound understanding of such technologies.

PKI fundamentals

- Communicating by using PGP, GPG or similar, keychain, trust in a key-pair
- Certificates, root certificates, trusted CAs, self-signed certificates, certificate chain, parsing of X.509, revoking, CRLs, OCSP, timestamping

6. APPROACHES AND METHODOLOGY

Beyond the aforementioned application domains it is necessary to understand the fields that aim at systematizing the approaches to assess and increase the security of ICT systems. We therefore suggest that the teams have a solid understanding of the methodologies developed in the following fields.

6.1 RECONNAISSANCE

Reconnaissance attacks can be active or passive. It is an attempt to gain information about targeted computers or networks that can be used as a preliminary step toward a further attack seeking to exploit the target system. Active reconnaissance involves port scans and OS scans, while passive reconnaissance relies on sniffing regular host traffic in order to gain information about its capabilities and vulnerabilities.

Passive reconnaissance

- Whois
- Google Dorks / Google Hacking
- Nmap
- DNS Tools
- DNS Recon
- dnscan
- theharvester

Active reconnaissance

- AMAP – Application Mapper. AMAP uses the results from Nmap to mine for more info.
- Nessus – Vulnerability Scanner
- Scanrand – Fast network scanner
- Paratrace – TCP Traceroute that utilizes selected TTL messages

6.2 CRYPTANALYSIS

A good survey on modern cryptanalysis techniques can be found in the overview of the external analysis on Simon and Speck [Bea+17]. An in-depth treatment of block ciphers, differential cryptanalysis and linear cryptanalysis can be found in [KR11].

6.3 OPERATIONS SECURITY

An important part of keeping a local setup secure is the constant monitoring and analysis of the current state. Correlation of monitoring data as well as its presentation and means for manual analysis have to be understood. It thus a prerequisite for ECS teams to have a solid grasp of techniques and tools for the logging and processing of log files, in order to support detection of suspicious behaviour.

Attack detection

- SIEM & Log Analysis (Alerts & Audit logs)
- HIDSW/HIPS
- Network Traffic Analysis (NETFLOWS, PCAP)
- Malware Repository Search
- Artefact Analysis

6.4 FORENSICS / MALWARE ANALYSIS

Forensics is the process of analysing systems to provide evidence for legal cases. Core skills refer to the understanding that at its heart: forensics aims to establish a claim under given assumptions by providing evidence with clear provenance. For this reason, primitives of forensics, as well as of malware analysis should be understood by all teams of ECSC.

Forensics fundamentals

- Locards Transfer Principle, physical transfer, transfer of traits, association, event reconstruction
- Digital evidence, integrity and authenticity, media of evidence, ephemeral vs persistent evidence, avoidable vs unavoidable (hardly avoidable, hard to hide) evidence, chain of custody (logging, crypto hash for digital evidence preservation)
- Types of data: primary, secondary, program, configuration, logs/protocols
- Levels of certainty
- Data discovery, recovery (RAM, file carving), extraction, integrity/authenticity, evidence of absence of data/manipulated evidence (log files), triage: live/incident response (TriageIR, TR3Secure, Kludge), disk triage (bulk extractor)
- Steps of IT-forensics: (a) what happened, (b) where, (c) when, (d) how; potentially (e) attribution (by whom), (f) how is prevented in the future
- File forensics, encoding, file headers, magic numbers, metadata
- Email forensics (header analysis, SPF, DMARC, DKIM)
- RAM forensics (volatility)
- Network forensics, pcap analysis (Wireshark), flow analysis
- Detecting steganography
- Tools
    - Live imaging, e.g., FTK Imager
    - Testing for encryption, e.g., EDD
- Supportive tools: IDS (host/network), immutable logs, honey pots

Malware analysis

- Static analysis [Insb] [Kon]

- Dynamic analysis [Pjl]
- Malware Sandbox / automated analysis [Neu+14] [Cuc]
- Anti-analysis Techniques [Lab]

Suggested reading: [SIK12], [LIG14], [Lig+11], [CAR14]

Mobile vulnerabilities and exploits

- Secure App Development
- Mobile forensics (Android, iOS)
- Mobile pentesting
- Tools and Resources

Suggested reading: [Val] [rapb], [rapa], [Now], [ENIb], [Wika], [Insa], [Yer], [Wikb], [Sec].

7. MEASURING ACHIEVEMENTS

When it comes to measuring the level of mastery of a particular skill, the document distinguishes between the following three levels:

1. Understanding the nature of the problem
2. Applying the correct principles to simple, isolated, textbook examples
3. Solve the problem in a real-world context, connect, and transfer knowledge between fields.

In educational terminology, the first level can be achieved by listening to someone explaining the problem or reading/studying a document. The second level can be achieved by making so-called pen-and-paper exercises (which may be computer assisted, but where the computing environment does not simulate a real environment). The third level requires that the students have access to a lab environment containing a simulation of a real system.

In order to facilitate accurate measurement of the students' competences at all three levels, it is important to include challenges that require competence at only one or two of the levels. Otherwise, we risk that students who get stuck early can never show their competences in the other fields. For challenges that do combine skills at 2 or 3 different levels, it is important to make sure that the students can realistically complete all steps of the challenge.

The third level is typically measured in the type of challenges that [Hac16] calls attack/defence style competitions. Students have to (defend) administer a computer system, database, web server, etc., and at the same time test the defences of the other teams.

The first and the second level are typically measured in quizzes and puzzles (called jeopardy). Because CTF competitions are stressful situations for students, it can be expected that the students will make many small mistakes, e.g., in mathematical computations, in programming, etc. Such mistakes might not always be evident in the answers to puzzles and quizzes. To the extent possible, the challenges should be made in a style that leads to answers that contain inherent redundancy or that can be verified by solving them again according to a different solving strategy. Including an attack-style challenge (or code-breaking puzzle) can be useful to make the students think deeper about defence options. Secondly, during a competition, including attacks against other teams as challenges can help to assess the other teams' strength in defence. However, it is important to keep in mind that we want to raise students who are good at defending against cyber-threats, rather than to teach them how to become cybercriminals. Therefore, the challenges should include a sufficient amount of constructive challenges where students are asked to build something secure, to use an existing technology in the proper way, etc.

Some key principles for the design of challenges:

1. The challenge should be made in a style that leads to answers that contain inherent redundancy.
2. The challenge should be solvable by using multiple solving strategies.
3. The challenge should include a sufficient amount of constructive elements.
4. The challenge should include clear references to the concepts which relate to it.
5. The challenge should underline the importance of understanding the cost of different defence strategies.
6. The challenge should test one or more of the key skills mentioned in Section 4.
7. There should be challenges of a wide range of difficulty levels.

8. CHALLENGES DESIGN CONSIDERATIONS

All challenges should follow a common 9-step design process.

1. Identification of requirements

   The ECSC planning committee should draft a series of guidelines on the challenge, in terms of nature (local, remote, cooperation-based, offline, online, etc.), scale (individual, team, etc.), difficulty (basic, intermediate, advanced) and skills (see section 4). For challenges to be solved by individuals, the combination of skills selected should remain within a given skill set, a meaningful combination of skill sets or a combination of one advanced and two or more basic skills. No restrictions on skills should apply to challenges to be solved by teams.

2. Identification of the challenge developer

   The ECSC planning committee should share the requirements for each challenge with potential challenge developer, analyse and select proposals.

3. Selection of topics

   Based on the skills selected for the challenge, a list of topics to be addressed must be chosen. A variety of topics should be covered and range from traditional security cases, inspired as much as possible from real cases, to forward-looking cases requiring out-of-the-box thinking from participants. Provided multiple parties are to develop challenges, the ECSC planning committee should be involved to ensure topics diversity. For all challenges, the most complex topics should be introduced last so as to avoid early drop-outs.

4. Identification of flags

   In order to allow for automatic scoring for each of the technical topics identified, flags must be inserted during the design of the challenge at every important step of the analysis. These flags will allow participants to know dynamically that they are making progress, up until the end of the challenge. Depending on whether incidents are to be analysed locally, remotely or within a specific environment, flags will vary. These can range from callbacks to pop-ups, IP addresses to access, files to open, phone numbers to call, etc.

5. Development of a narrative

   Given the overall narrative ark proposed by the ECSC planning committee, each challenge should provide a number of hooks to fit within the narrative. These can range from technical elements such

IP addresses, webpages, techniques, tactics and procedures, indicators of compromise, or personal data, to narrative elements such as news stories, pictures, videos, interrogation scripts, photocopies, etc. Multiple Easter eggs should also be placed within each challenge so as to incentivize participants to complete the challenges, to discover, and to share with their peers. Easter eggs can range from information about other challenges, completely unrelated elements such as jokes, etc. The ECSC planning committee could decide to further gamify the search for Easter eggs and hereby define a common theme.

6. Document review

   Upon finalization of the previous steps, descriptive elements should be submitted to the ECSC planning committee for review. The ECSC will verify compliance with the requirements and approve or request modifications.

7. Challenge development

   Once all documents have been reviewed and approved by the ECSC planning committee, the technical elements of the challenge should be developed.

8. Creation of a solution document

   In order to support the review of the challenge by the ECSC planning committee but also to inform challenge takers after the ECSC competition on how to solve each particular challenge, a solution document should be created. It should follow a predefined template and contain a description of all the steps, tools and techniques used to solve the challenge. Last but not least, each step should be weighted in terms of complexity, so as for the ECSC to implement in the scoring system.

9. Final acceptance

   The ECSC planning committee will appoint a testing committee which will attempt to solve the challenge without prior knowledge. Upon final acceptance from the testing and the review committee, the ECSC planning committee will integrate the challenge in the ECSC event and scoring system.

9. REFERENCES

[Bac86] Maurice J. Bach. The Design of the UNIX Operating System. Prentice Hall, 1986.

[And06] A. S. Woodhull Andrew. S. Tanenbaum. Operating Systems Design and Implementation, 3rd edition. Prentice-Hall, 2006.

[Bil+09] Leyla Bilge et al. “All Your Contacts Are Belong to Us: Automated Identity Theft Attacks on Social Networks”. In: World Wide Web Conference (WWW). 2009, pp. 551–560.

[Lyo09] Gordon Fyodor Lyon. Nmap network scanning: The official Nmap project guide to network discovery and security scanning. Insecure, 2009.

[Had10] Christopher Hadnagy. Social Engineering: The Art of Human Hacking. John Wiley & Sons, 2010.

[KR11] Lars R. Knudsen and Matthew J.B. Robshaw. The Block Cipher Companion. 2011.

[Lig+11] Michael Halr Ligh et al. Malware Analyst’s Cookbook: Tools and Techniques for Fighting Malicious Code. ISBN: 978-0-470-613030. John Wiley & Sons, Inc., 2011.

[SIK12] Andrewe SIKORSKI Michael; HONIG. Practical Malware Analysis. ISBN: 1-59327-290-1. No Starch Press, Inc, 2012.

[KSG13] Michal Kosinski, David Stillwell, and Thore Graepel. “Private traits and attributes are predictable from digital records of human behavior”. In: PNAS (2013).

[CAR14] Harlan A. CARVEY. Windows Forensic Analysis Toolkit Advanced Analysis Techniques for Windows 8, 4th Edition. ISBN: Syngress Publishing, Inc., 2014.

[KKV14] Tero Karvinen, Kimmo Karvinen, and Ville Valtokari. Make: Sensors: A hands-on primer for monitoring the real world with arduino and raspberry pi. Maker Media, Inc., 2014.

[LIG14] Michael Hale LIGH. the Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory. ISBN: John Wiley & Sons, Inc., 2014.

[Mar14] Michael Margolis. “Arduino Cookbook”. In: (2014). [Neu+14] Sebastian Neuner et al. “Enter sandbox: Android sandbox comparison”. In: arXiv preprint arXiv:1410.7749 (2014).

[T+14] Kevin Townsend, Carles Cuf´ı, Robert Davidson, et al. Getting started with Bluetooth low energy: Tools and techniques for lowpower networking.” O’Reilly Media, Inc.”, 2014.

[GT16] Dominique Guinard and Vlad Trifa. Building the web of things: with examples in node. js and raspberry pi. Manning Publications Co., 2016.

[Hac16] Hacking-Lab. The CTF-Player Handbook. Version 1.2. 2016.

[Mon16] Simon Monk. Raspberry Pi cookbook: Software and hardware problems and solutions.” O’Reilly Media, Inc.”, 2016.

[RS16] Michael Rossberg and Guenther Schaefer. Security in Fixed and Wireless Networks. Wiley, 2016.

[Ale17] Pavel Yosifovich Alex Ionescu Mark E. Russinovich. Windows Internals, Part 1. Microsoft Press, 2017.

[Bea+17] Ray Beaulieu et al. Notes on the design and analysis of SIMON and SPECK. Cryptology ePrint Archive, Report 2017/560. http: //eprint.iacr.org/2017/560. 2017.

[BP17] Jessey Bullock and Jeff. T. Parker. Wireshark for Security Professionals: Using Wireshark and the Metasploit Framework, 1st Edition. Wiley, 2017.

[G´er17] Aurelien Geron. Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. 2017.

[Pad+17] John Padgette et al. “Guide to Bluetooth Security”. In: Special Publication (NIST SP)-800-121 Rev 2 (2017). [Tro+17] Carmela Troncoso et al. “Systematizing Decentralization and Privacy: Lessons from 15 years of research and deployments”. In: Proceedings of PETs (2017).

[CED] CEDEFOP. Skill shortages in Europe: Which occupations are in demand – and why. url: http://www.cedefop.europa.eu/ en/news-and-press/news/skill-shortages-europe-whichoccupations-are-demand-and-why.

[Cuc] Cuckoo. Cuckoo Sandbox. url: https://cuckoosandbox.org/.

[ENIa] ENISA. Cybersecurity competition—the status in Europe. url: https://www.enisa.europa.eu/publications/cybersecuritycompetitions-2014-the-status-in-europe. [ENIb] ENISA. Smartphone Secure development guidelines. url: https: //www.enisa.europa.eu/publications/smartphone-securedevelopment-guidelines-2016.

[Insa] Infosec Institute. Android Forensics Labs. url: http://resources. infosecinstitute.com/android-forensics-labs/#gref. [Insb] Infosec Institute. Static Analysis of iOS app. url: http:// resources.infosecinstitute.com/part-15-static-analysisof-ios-apps-using-inalyzer/#gref.

[Knu] Lars R. Knudsen. Cryptology, how to crack it.

[Kon] Konloch. Bytecode Viewer. url: https://github.com/Konloch/ bytecode-viewer.

[Lab] Sophos Labs. Android Anti-emulation techniques. url: https: //news.sophos.com/en-us/2017/04/13/android-malwareanti-emulation-techniques/.

[Now] NowSecure. Secure Mobile Development best practices. url: https: //www.nowsecure.com/ebooks/secure-mobile-developmentbest-practices/.

[Pjl] Pjlantz. Droidbox. url: https://github.com/pjlantz/droidbox.

[rapa] rapid7. Vulnerability Database (Android). url: https://www. rapid7.com/db/search?utf8=%E2%9C%93&q=Android&t=a. [rapb] rapid7. Vulnerability Database (iOS). url: https://www.rapid7. com/db/search?utf8=%E2%9C%93&q=Apple%2BiOS&t=a.

[Sec] Now Secure. Santoku Howto. url: https://santoku-linux. com/howtos/.

[Val] ENISA Vishnu Valentino. Hacking Android Smartphone tutorial using Metasploit. url: http://www.hacking-tutorial.com/ hacking-tutorial/hacking-android-smartphone-tutorialusing-metasploit/.

[Wika] Forensics Wiki. iPhone forensics. url: http://www.forensicswiki. org/wiki/Apple_iPhone.

[Wikb] SecMobi Wiki. Collection of mobile security resources. url: https: //github.com/secmobi/wiki.secmobi.com.

[Yer] Swaroop Yermalkar. Learning iOS Penetration Testing. url: https: //www.amazon.com/Learning-Penetration-Testing-SwaroopYermalkar/dp/1785883259.

ENISA
European Union Agency for Cybersecurity

Athens Office
1 Vasilissis Sofias Str.
151 24 Marousi, Attiki, Greece

Heraklion Office
95 Nikolaou Plastira
700 13 Vassilika Vouton, Heraklion, Greece

enisa.europa.eu