

RESISTO:

D3.4_ Methods for cyber-physical security management for telecom CI



RESISTO

D3.4 – METHODS FOR CYBERPHYSICAL SECURITY MANAGEMENT FOR TELECOM CI

Document Manager:	Mirjam Fehling-Kaschek	Fraunhofer	Editor
--------------------------	------------------------	------------	--------

Project Title:	RESilience enhancement and risk control platform for communication infraSTructure Operators
Project Acronym:	RESISTO
Contract Number:	786409
Project Coordinator:	LEONARDO
WP Leader:	Fraunhofer

Document ID N°:	RESISTO_D3.4_191218_01	Version:	1.1
Deliverable:	D3.4	Date:	18/12/2019
		Status:	APPROVED

Document classification	Public
--------------------------------	---------------

Approval Status	
Prepared by:	Mirjam Fehling-Kaschek (Fraunhofer)
Approved by: (WP Leader)	Mirjam Fehling-Kaschek (Fraunhofer)
Approved by: (Coordinator)	Bruno Saccomanno (LDO)
Advisory Board Validation (Advisory Board Coordinator)	Carmen Patrascu (ORO)
Security Approval (Security Advisory Board Leader)	Paolo Di Michele (LDO)

CONTRIBUTING PARTNERS

Name	Company / Organization	Role / Title
Mirjam Fehling-Kaschek, Gael Haab, Jörg Finger, Natalie Miller	Fraunhofer	Scientific Researcher
Giuseppe Amato, Giuseppe Celozzi	TEI	Contributors
Sylvia Bach	BUW	Contributor
Evangelos Sfakianakis	OTE	Contributor
Lucian Enescu, Octavian Echim, Ioan Constantin	ORO	Information Security Experts
Marco Mella	TIM	Information Security Expert

DISTRIBUTION LIST

Name	Company / Organization	Role / Title
PMT	RESISTO CONSORTIUM	NA
Markus Muller	EC DG REA	EC Programme Officer
General Public	NA	NA

REVISION TABLE

Version	Date	Modified Pages	Modified Sections	Comments
0.1	09/08/2019	All	All	First Version
0.2	15/08/2019	All	All	WP3 review comments
0.8	12/09/2019		4.5	Considering EU review comments
0.9	15/10/2019		All	Final Release
1.0	18/12/2019		All	Implementation of AB comments

COPYRIGHT STATEMENT



© 2018-2021 This document and its content are the property of the RESISTO Consortium. All rights relevant to this document are determined by the applicable laws. Access to this document does not grant any right or license on the document or its contents. This document or its contents are not to be used or treated in any manner inconsistent with the rights or interests of the RESISSO Consortium or the Partners detriment and are not to be disclosed externally without prior written consent from the RESISTO Partners. Each RESISTO Partner may use this document in conformity with the RESISTO Consortium Grant Agreement provisions.

The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Program, under the Grant Agreement No 786409.

The views and opinions in this document are solely those of the authors and contributors, not those of the European Commission.

PROJECT CONTACT



LEONARDO
Via Puccini 2 – Genova (GE) – 16154 – Italy
Tel.: +39 348 6505565
E-Mail: bruno.saccomanno@leonardocompany.com

RESISTO PROJECT – PUBLISHABLE EXTENDED ABSTRACT

Communications play a fundamental role in the economic and social well-being of the citizens and on operations of most of the CIs. Thus they are a primary target for criminals having a multiplier effect on the power of attacks and providing enormous resonance and gains. Also extreme weather events and natural disasters represents a challenge due to their increase in frequency and intensity requiring smarter resilience of the Communication CIs, which are extremely vulnerable due to the ever-increasing complexity of the architecture also in light of the evolution towards 5G, the extensive use of programmable platforms and exponential growth of connected devices. The fact that most enterprises still manage physical and cyber security independently represents a further challenge. RESISTO platform is an innovative solution for Communication CIs holistic situation awareness and enhanced resilience (aligned with ECSO objectives). Based on an Integrated Risk and Resilience analysis management and improvement process availing all resilience cycle phases (prepare, prevent, detect, absorb, etc.) and technical resilience capabilities (sense, model, infer, act, adopt), RESISTO implements an innovative Decision Support System to protect communication infrastructures from combined cyber-physical threats exploiting the Software Defined Security model on a suite of state of the art cyber/physical security components (Blockchain, Machine Learning, IoT security, Airborne threat detection, holistic audio-video analytics) and services (Responsible Disclosure Framework) for detection and reaction in presence of attacks or natural disasters. Through RESISTO Communications Operators, will be able to implement a set of mitigation actions and countermeasures that significantly reduce the impact of negative events in terms of performance losses, social consequences, and cascading effects in particular by bouncing efficiently back to original and forward to operational states of operation.

EXECUTIVE SUMMARY

This deliverable (D3.4) summarizes the final status of task 3.2 (T3.2). Goal of this task is to collect fast and flexible methods supporting the risk and resilience assessment. The methods should comply with existing standards and handbooks as far as possible and cover cyber-physical threats in telecommunication infrastructures.

The general aim of WP3 is to define the long term control loop of the RESISTO platform, providing a joint risk and resilience analysis and management process for cyber, physical and cyber-physical threats. The methods and tools collected within this task serve as input and support to the analysis and management process.

The report is based on the first version of the report (D3.3). Updates and modifications made for D3.4 are summarized in the Introduction.

CONTENTS

ABBREVIATIONS	11
1. INTRODUCTION	12
2. RISK AND RESILIENCE MANAGEMENT	14
2.1. Input collection via spread sheet template	14
2.2. Inferring critical combinations	15
3. EXISTING STANDARDS AND HANDBOOKS	16
3.1. Standards	16
3.1.1. ISO/IEC 27000 family	16
3.2. Frameworks, Handbooks and Guidelines	19
3.2.1. Plan-Do-Check-Act (PDCA) cycle	19
3.2.2. NIST Framework for Improving Critical Infrastructure Cybersecurity	20
3.2.3. NIST SP 800-100, Information Security Handbook: A Guide for Managers	20
3.2.4. NIST Computer Security Incident Handling Guide	20
3.2.5. Centre for Internet Security (CIS)	20
3.2.6. Open Web Application Security Project (OWASP)	21
3.2.7. Open Source Security Testing Methodology Manual (OSSTMM)	21
3.2.8. ENISA Recommendations to IT Industry	21
3.2.9. Microsoft Security Development Lifecycle	24
3.2.10. MITRE ATT&CK framework	24
4. COLLECTION OF ASSESSMENT METHODS AND TOOLS	26
4.1. Deductive approaches	27
4.1.1. Attack trees	27
4.1.2. Stuxnet attack tree example	28
4.1.3. GhostNet attack tree example	28
4.1.4. Dynamic attack tree	29
4.1.5. Combined attack tree	31
4.1.6. Attack-Defence tree	33
4.2. Honeypots	33
4.2.1. Physical Honeypots	35
4.2.2. Network Honeypots	36
4.2.3. Application Honeypots	36
4.2.4. Data Honeypots	37
4.2.5. Research Honeypots	37
4.2.6. Production Honeypots	38
4.2.7. Quarantine Honeypots	38
4.2.8. Canary Honeypots	39
4.2.9. Social Engineering Honeypots	39
4.3. Penetration test assessment	40

4.3.1. Goals	40
4.3.2. Types of Penetration Tests	41
4.3.3. Penetration Testing Stages	42
4.3.4. Penetration Testing Activities	42
4.3.5. Penetration Testing Programme	44
4.3.6. Penetration Testing Methodologies	44
4.4. MITRE ATT&CK.....	45
4.4.1. Sample of Tactic - Credential Access.....	46
4.5. Usage of the tools within RESISTO	47
4.6. Requirements traceability	48
5. WEB-APP FOR RISK AND RESILIENCE ASSESSMENT.....	50
6. SUMMARY	55
REFERENCES	56

Table of figures

<i>Figure 1: RESISTO logical architecture (see Deliverable D2.6 “RESISTO platform and tools reference architecture” for more information). Aim of the task described in this report is to identify and evaluate methods for the fast and flexible risk and resilience assessment in the Long Term Control Loop.</i>	12
<i>Figure 2: Risk and resilience management processes (Häring, 2017). The risk management process (right) follows the definition of ISO 31000 (2009) Risk management – Principles and guidelines.</i>	14
<i>Figure 3: Plan-do-check-act cycle</i>	19
<i>Figure 4: Physical Safe - Attack tree (Schneier, 1999)</i>	27
<i>Figure 5: Stuxnet attack tree example (Ola Flaten, 2014)</i>	28
<i>Figure 6: GhostNet - Attack tree example (Ola Flaten, 2014)</i>	29
<i>Figure 7: Dynamic tree example: Stuxnet (Florian Arnold, 2015)</i>	29
<i>Figure 8: Dynamic Attack Tree: RAS malicious access example (Ludovic Piètre-Cambacédès, 2010)</i>	30
<i>Figure 9: Simple example of combined attack tree (Marco Gribaudo, 2015).</i>	31
<i>Figure 10: Multiple Domain Combined Cyber-Physical Attack tree (Marco Gribaudo, 2015)</i>	32
<i>Figure 11: Attack-Defence trees (Barbara Kordy, 2012).</i>	33
<i>Figure 12: Honeypot characteristics</i>	34
<i>Figure 13: Logical diagram for penetration testing programme</i>	44
<i>Figure 14: Extract of ATT&CK Matrix™ - https://attack.mitre.org/matrices/enterprise/</i>	45
<i>Figure 15: ATT&CK tactics</i>	46
<i>Figure 16: Techniques of Credential Access tactics</i>	47
<i>Figure 17: Risk and resilience management process based on (Häring, 2017). The usage of the tabular Excel inputs and graphical network representations is indicated by orange boxes and the support by the Shiny app and simulation tools by blue or green boxes, respectively.</i>	50
<i>Figure 18: Start screen of the Shiny app. The main panel in black on the left allows to switch between the main features of the app. The option of viewing the tables is selected by default. It allows to browse all tables of the Excel inputs and searching them.</i>	52
<i>Figure 19: Connections option of the Shiny app, visualizing the connections between the items. In the plotted example SF5 was clicked and information about this system function and the connected system components and threats is printed below the plot.</i>	53
<i>Figure 20: Correlation option for the Shiny app, printing connection strength matrices for two chosen tables, e.g. critical combinations of threats and system functions. See main text for details about the strength computation</i>	53
<i>Figure 21: Threat Ranking option of the Shiny app. The user can define a score model based on the inputs in the threats table to rank the threats according to this model. Further details are given in the main text.</i>	54

ABBREVIATIONS

2G, 3G, 4G	Second, third and fourth generation of mobile phone systems
AI	Artificial Intelligence
CI	Critical Infrastructure
EU	European Union
HW	Hardware
ICT	Information and Communication Technology
IoT	Internet of Things
IT	Information Technology
KPI	Key Performance Indicator
LTCL	Long-Term-Control-Loop (of the RESISTO platform)
LTE	Long Term Evolution (= 4G)
OT	Operational Technology
PDCA	Plan-Do-Check-Act
Pentest	Penetration Test
PLCs	Programmable Logic Controllers
SDL	Security Development Lifecycle
SSH	Secure Shell
STCL	Short-Term-Control-Loop (of the RESISTO platform)
SW	Software
T	Task (this deliverable refers to T3.2)
WP	Work Package (this deliverable refers to WP3)

1. INTRODUCTION

The main objective of the RESISTO project is to improve the resilience in communication infrastructures by developing a platform for an optimized decision support. The RESISTO platform interfaces to existing communication infrastructures and modularly integrates tools and methods in the integration platform, which consists of two control loops, the short term and the long term control loop. A global scheme of the architecture of the integration platform is shown in Figure 1.

Aim of work package (WP) 3 “Cyber-physical risk/resilience assessment and improvement process for preparation, prevention and protection” is the definition of the long term control loop (LTCL) of the RESISTO platform.

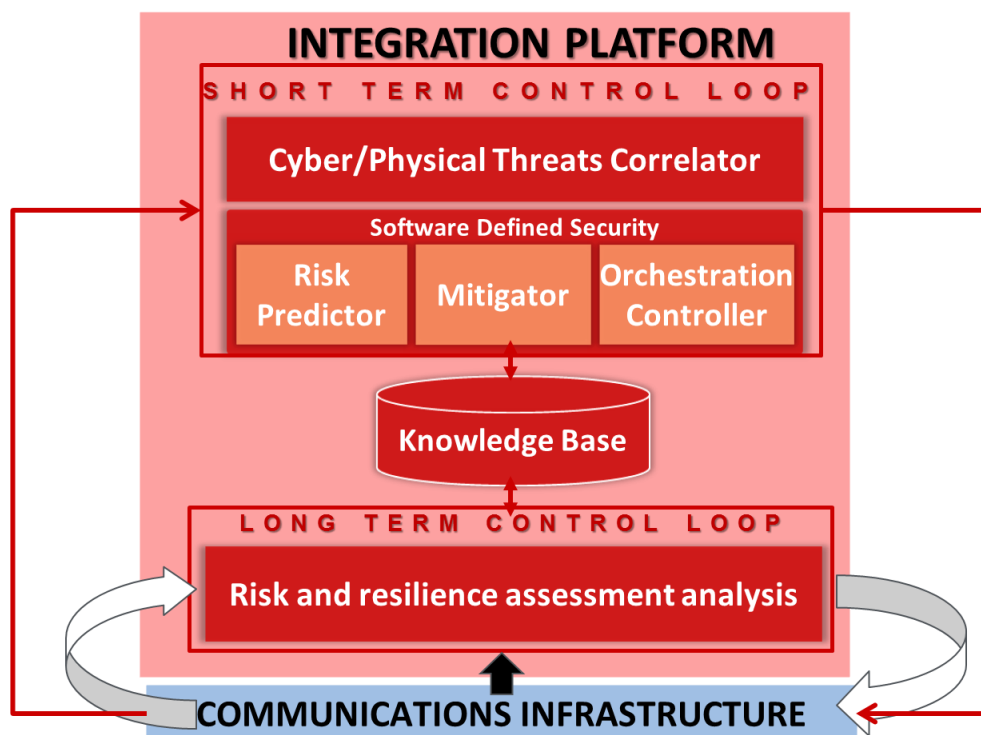


Figure 1: RESISTO logical architecture (see Deliverable D2.6 “RESISTO platform and tools reference architecture” for more information). Aim of the task described in this report is to identify and evaluate methods for the fast and flexible risk and resilience assessment in the Long Term Control Loop.

The main feature of the long term control loop is the risk and resilience analysis and management process for telecommunication CIs. It covers telecommunication specific cyber-physical risk control and resilience analytics. Besides the risk and resilience analysis tool for cyber, physical and cyber-physical threats, a main outcome of the WP is the definition and provision of key performance indicators (KPIs) for the risk and resilience assessment.

WP3 is split into five tasks:

T3.1 Long term learning cyber-physical risk and resilience management

T3.2 Methods/Plans for joint cyber-physical security management process

T3.3 Physical protection and prevention methods: assessment and cyber-physical interaction

T3.4 Risk and resilience quantities and related KPIs for telecommunications infrastructure

T3.5 Desk-top application to use case scenarios for second use cases refinement

Goal of T3.2 is to provide a list of methods for the fast and flexible risk and resilience assessment. The methods should be based on best practice of the operators and technical partners and comply with existing standards and handbooks if possible.

This report summarizes the final status of T3.2 and builds on the intermediate version of the report, deliverable D3.3. For this final report, all technical partners and in particular the operators were asked to review and update the list of standards and tools in chapter 3 and 4. A major focus was set on the question if and how the tools could be applied or used in RESISTO. In the following, the structure of the report is summarized and updates with respect to the first version of the report (D3.3) are highlighted for each chapter.

Chapter 2 summarizes the risk and resilience management approach followed in the RESISTO project. A focus is set on a tabular input collection method and the determination of criticalities.

- ➔ Updates: Only minor updates were added in the text since the resilience management process is the main objective of T3.1 and will be updated in D3.2 “Risk and resilience management process for cyber-physical threats of telecom CI”.

Chapter 3 provides an overview of existing standards and handbooks known and referred to by the telecommunication operators.

- ➔ Updates: Large modifications were made in section 3.2.8 where new results from an ENISA report were added. A new contribution was added about the MITRE ATT&CK framework in section 3.2.10.

Chapter 4 lists methods for the risk and resilience assessment known to or used by the telecommunication operators.

- ➔ Updates: Information about various types of pentests were added in section 4.3.2. A new contribution was added in section 4.4 about MITRE ATT&CK. In addition, the relevance for RESISTO was studied considering two aspects: the usage of the tools in RESISTO in section 4.5 and the traceability of requirements from D2.1 “End user requirements for integrated cyberphysical risk and resilience management” in section 4.6.

Chapter 5 provides a short description of a fast and flexible assessment tools in form of a web-application, developed to support the risk and resilience analysis management process.

- ➔ Updates: The text and references to other WPs and deliverables were updated. In particular, tools from chapter 4 were graphically added to the risk and resilience management process (see *Figure 17*).

Chapter 6 summarizes this report.

- ➔ Updates: The summary was updated to take into account that this is the final version of the report from T3.2.

It should be noted that the aspect of security versus resilience management was discussed thoroughly in the consortium. It was agreed that security functions should also be considered by the risk and resilience management process. A more detailed discussion is given in section 3.3.5 of D3.6 “Damage/Vulnerability models for physical and cyber threats of telecom CI” from T3.3.

2. RISK AND RESILIENCE MANAGEMENT

The risk and resilience assessment is performed by following an integrated risk and resilience management process, which is described in (Häring, 2017). This assessment and improvement process extends the ISO 31000 standard by further dividing the subsequent steps of the closed management process loop, as shown in *Figure 2*. The ISO 31000 was updated in 2018, but the extension to the integrated risk and resilience management process still holds (see also deliverable D3.1 “Risk and resilience management process for cyber-physical threats of telecom CI”).

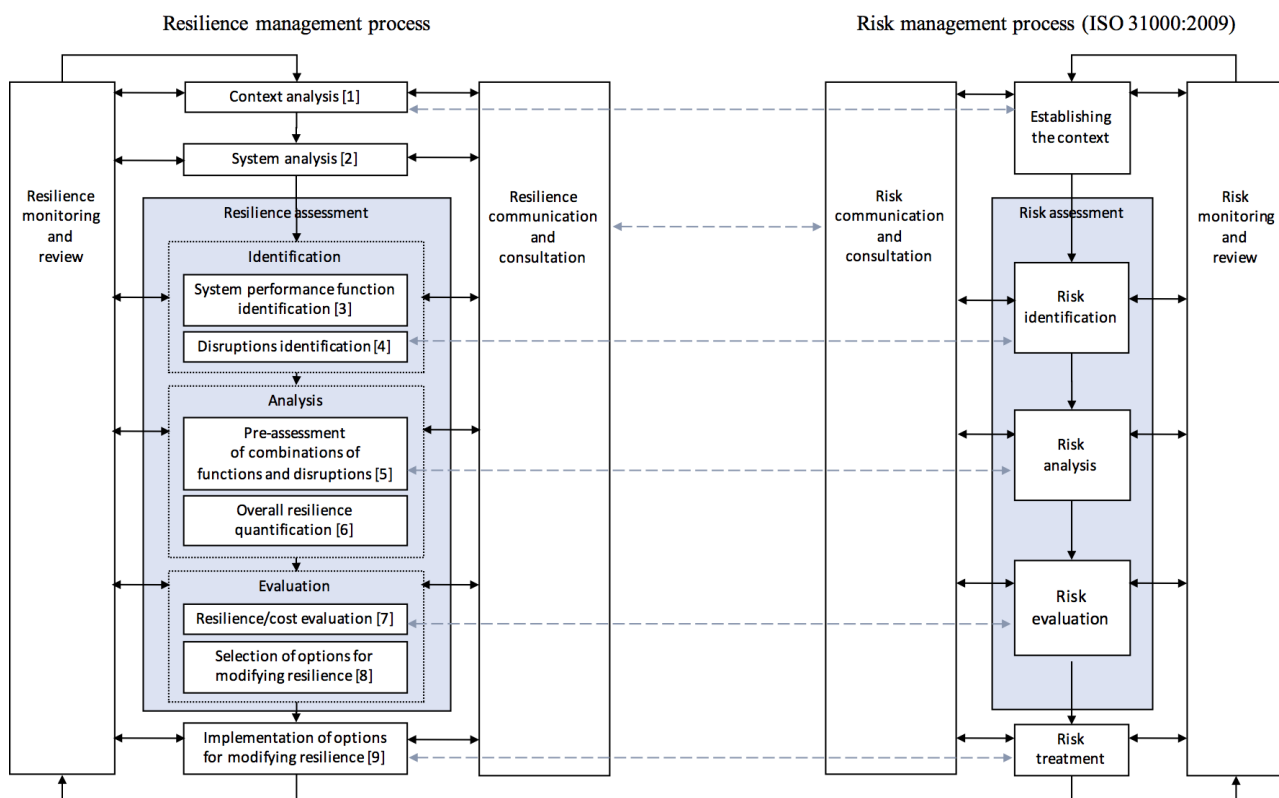


Figure 2: Risk and resilience management processes (Häring, 2017). The risk management process (right) follows the definition of ISO 31000 (2009) Risk management – Principles and guidelines.

Several specific inputs (e.g. information about the system) and tools (e.g. for resilience quantification) are needed in order to process all nine subsequent steps. A more detailed description of the management process is given in the deliverable D3.1 of T3.1, which will be revised and updated in the coming deliverable D3.2 “Risk and resilience management process for cyber-physical threats of telecom CI”.

2.1. Input collection via spread sheet template

One main input source to the risk and resilience management process was developed within the RESISTO project: an Excel template composed of four main tables covering tabular information about

1. System components, 2. System functions, 3. Threats and 4. Mitigation options. The contents of the tables are linked, allowing to deduce the necessary connections between the inputs.

In order to access the information of the tables, a Shiny (RStudio, 2019) web-app was developed, supporting a fast and flexible risk and resilience assessment. The app is described in chapter 5.

2.2. Inferring critical combinations

Critical combinations refer to a set of system (performance) functions and threats for which critical resilience issues are prognosticated. Their determination and evaluation are major steps in the risk and resilience management process (see step 5 in *Figure 2*).

The interlinkages of the Excel tables described in the previous section allow to directly deduce the correlation matrix for critical combinations. The output of correlation matrices is an implemented feature of the Shiny web-app described in chapter 5.

3. EXISTING STANDARDS AND HANDBOOKS

This chapter provides an overview of standards, frameworks, handbooks and guidelines relevant for risk and resilience assessment methods.

3.1. Standards

3.1.1. ISO/IEC 27000 family

The ISO/IEC 27000 family is a collection of information security standards that provide a globally recognized framework for best-practice information security management. The relevant standards from this collection are listed in the following.

ISO/IEC 27001:2013 "Information technology — Security techniques — Information security management systems — Requirements (ISO, 2013a)

Is an information security management system (ISMS) standard. It specifies a management system that is intended to bring information security under management control and gives specific requirements.

Structure of the standard:

- (4.) Organizational context and stakeholders
- (5.) Information security leadership and high-level support for policy
- (6.) Planning an information security management system; risk assessment; risk treatment
- (7.) Supporting an information security management system
- (8.) Making an information security management system operational
- (9.) Reviewing the system's performance
- (10.) Corrective action
- Annex A (List of controls and their objectives):
 - A.5: Information security policies (2 controls)
 - A.6: Organization of information security (7 controls)
 - A.7: Human resource security - 6 controls that are applied before, during, or after employment
 - A.8: Asset management (10 controls)
 - A.9: Access control (14 controls)
 - A.10: Cryptography (2 controls)
 - A.11: Physical and environmental security (15 controls)
 - A.12: Operations security (14 controls)
 - A.13: Communications security (7 controls)
 - A.14: System acquisition, development and maintenance (13 controls)
 - A.15: Supplier relationships (5 controls)
 - A.16: Information security incident management (7 controls)
 - A.17: Information security aspects of business continuity management (4 controls)
 - A.18: Compliance; with internal requirements, such as policies, and with external requirements, such as laws (8 controls)

ISO/IEC 27002:2013 — Information technology — Security techniques — Code of practice for information security controls (second edition) (ISO, 2013b)

It provides best practice recommendations on information security controls for use by those responsible for initiating, implementing or maintaining information security management systems (ISMS).

Structure of the standard:

- Information Security Policies
- Organization of Information Security
- Human Resource Security
- Asset Management
- Access Control
- Cryptography
- Physical and environmental security
- Operation Security- procedures and responsibilities, Protection from malware, Backup, Logging and monitoring, Control of operational software, Technical vulnerability management and Information systems audit coordination
- Communication security - Network security management and Information transfer
- System acquisition, development and maintenance - Security requirements of information systems, Security in development and support processes and Test data
- Supplier relationships - Information security in supplier relationships and Supplier service delivery management
- Information security incident management - Management of information security incidents and improvements
- Information security aspects of business continuity management - Information security continuity and Redundancies
- Compliance - Compliance with legal and contractual requirements and Information security reviews

Other ISO/IEC 27000 family standards, see also (Disterer, 2013):

1. ISO/IEC 27003 — Information security management system implementation guidance
2. ISO/IEC 27004 — Information security management — Monitoring, measurement, analysis and evaluation
3. ISO/IEC 27006 — Requirements for bodies providing audit and certification of information security management systems
4. ISO/IEC 27007 — Guidelines for information security management systems auditing (focused on auditing the management system)
5. ISO/IEC TR 27008 — Guidance for auditors on ISMS controls
6. ISO/IEC 27009 — Essentially an internal document for the committee developing sector/industry-specific variants or implementation guidelines for the ISO27K standards
7. ISO/IEC 27010 — Information security management for inter-sector and inter-organizational communications
8. ISO/IEC 27011 — Information security management guidelines for telecommunications organizations based on ISO/IEC 27002
9. ISO/IEC 27013 — Guideline on the integrated implementation of ISO/IEC 27001 and ISO/IEC 20000-1 (derived from ITIL)

10. ISO/IEC 27014 — Information security governance.
11. ISO/IEC TR 27016 — information security economics
12. ISO/IEC 27017 — Code of practice for information security controls based on ISO/IEC 27002 for cloud services
13. ISO/IEC 27018 — Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors
14. ISO/IEC TR 27019 — Information security for process control in the energy industry
15. ISO/IEC 27031 — Guidelines for information and communication technology readiness for business continuity
16. ISO/IEC 27032 — Guideline for cybersecurity
17. ISO/IEC 27033-1 — Network security - Part 1: Overview and concepts
18. ISO/IEC 27033-2 — Network security - Part 2: Guidelines for the design and implementation of network security
19. ISO/IEC 27033-3 — Network security - Part 3: Reference networking scenarios - Threats, design techniques and control issues
20. ISO/IEC 27033-4 — Network security - Part 4: Securing communications between networks using security gateways
21. ISO/IEC 27033-5 — Network security - Part 5: Securing communications across networks using Virtual Private Networks (VPNs)
22. ISO/IEC 27033-6 — Network security - Part 6: Securing wireless IP network access
23. ISO/IEC 27034-1 — Application security - Part 1: Guideline for application security
24. ISO/IEC 27034-2 — Application security - Part 2: Organization normative framework
25. ISO/IEC 27034-6 — Application security - Part 6: Case studies
26. ISO/IEC 27035-1 — Information security incident management - Part 1: Principles of incident management
27. ISO/IEC 27035-2 — Information security incident management - Part 2: Guidelines to plan and prepare for incident response
28. ISO/IEC 27036-1 — Information security for supplier relationships - Part 1: Overview and concepts
29. ISO/IEC 27036-2 — Information security for supplier relationships - Part 2: Requirements
30. ISO/IEC 27036-3 — Information security for supplier relationships - Part 3: Guidelines for information and communication technology supply chain security
31. ISO/IEC 27036-4 — Information security for supplier relationships - Part 4: Guidelines for security of cloud services
32. ISO/IEC 27037 — Guidelines for identification, collection, acquisition and preservation of digital evidence
33. ISO/IEC 27038 — Specification for Digital redaction on Digital Documents
34. ISO/IEC 27039 — Intrusion prevention
35. ISO/IEC 27040 — Storage security
36. ISO/IEC 27041 — Investigation assurance
37. ISO/IEC 27042 — Analysing digital evidence
38. ISO/IEC 27043 — Incident investigation
39. ISO/IEC 27050-1 — Electronic discovery - Part 1: Overview and concepts
40. ISO 27799 — Information security management in health using ISO/IEC 27002 - guides health industry organizations on how to protect personal health information using ISO/IEC 27002.

3.2. Frameworks, Handbooks and Guidelines

3.2.1. Plan-Do-Check-Act (PDCA) cycle

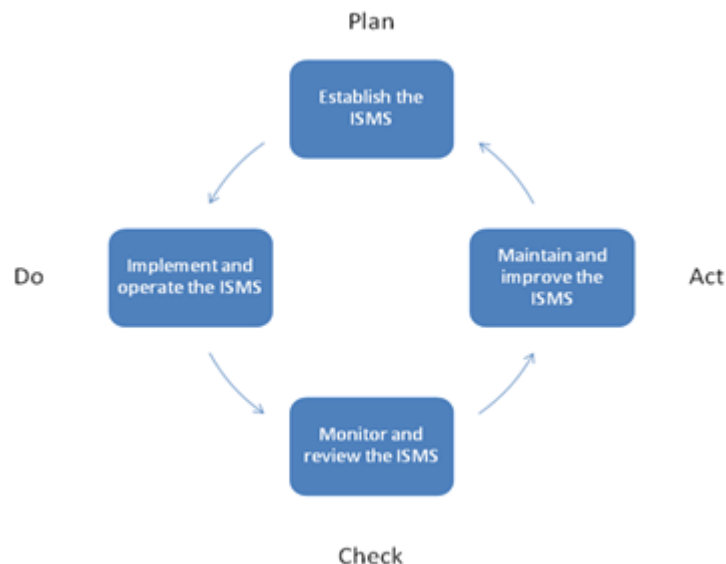


Figure 3: Plan-do-check-act cycle

When to Use Plan–Do–Check–Act

- As a model for continuous improvement.
- When starting a new improvement project.
- When developing a new or improved design of a process, product or service.
- When defining a repetitive work process.
- When planning data collection and analysis in order to verify and prioritize problems or root causes.
- When implementing any change.

Plan–Do–Check–Act Procedure

- Plan. Recognize an opportunity and plan a change.
- Do. Test the change. Carry out a small-scale study.
- Check. Review the test, analyse the results and identify what you have learned.
- Act. Take action based on what you learned in the study step: If the change did not work, go through the cycle again with a different plan. If you were successful, incorporate what you learned from the test into wider changes. Use what you learned to plan new improvements, beginning the cycle again.

3.2.2. NIST Framework for Improving Critical Infrastructure Cybersecurity

The NIST Cybersecurity Framework¹ provides a policy framework of computer security guidance for how private sector organizations in the United States can assess and improve their ability to prevent, detect, and respond to cyber-attacks.

3.2.3. NIST SP 800-100, Information Security Handbook: A Guide for Managers

This Information Security Handbook² provides a broad overview of information security program elements to assist managers in understanding how to establish and implement an information security program.

3.2.4. NIST Computer Security Incident Handling Guide

The NIST Computer Security Incident Handling Guide³ seeks to assist organizations in mitigating the risks from computer security incidents by providing practical guidelines on responding to incidents effectively and efficiently. It includes guidelines on establishing an effective incident response program, but the primary focus of the document is detecting, analysing, prioritizing, and handling incidents:

3.2.5. Centre for Internet Security (CIS)

The Center for Internet Security provides a series of Guidelines⁴ and Controls⁵ for the safeguarding of Operating Systems, Software and Networks and Foundational and Advanced Cyber Security Actions developed to aid mitigation of most common types of Cyber Attacks

¹ <https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf>

² <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-100.pdf>

³ <https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-61r2.pdf>

⁴ <https://www.cisecurity.org/cis-benchmarks/>

⁵ <https://www.cisecurity.org/controls/>

3.2.6. Open Web Application Security Project (OWASP)

The Open Web Application Security Project ⁶(OWASP) aims to maintain and update an dynamic list of the 10 most common Web Application Security Risks and Controls and Techniques designed to improve the Cyber Security in every software development cycle.

OWASP provides a comprehensive framework for Penetration Testing Methodologies for Web Applications.

3.2.7. Open Source Security Testing Methodology Manual (OSSTMM)

OSSTMM⁷ is a methodology to test the operational security of physical locations, human interactions, and all forms of communications such as wireless, wired, analogue, and digital.

3.2.8. ENISA Recommendations to IT Industry

In this paper titled “ENISA Recommendations to IT Industry”, ⁸ENISA puts forward 10 messages to industry. The ultimate goals behind these messages are

- (a) to encourage the establishment of a high level of cybersecurity across all industry segments and
- (b) to ensure that cybersecurity is an enabler and not an inhibitor of a more efficient market.

Additionally, in a recent threat landscape report (European Union Agency for Cybersecurity, 2018), ENISA listed the 15 most important cyber threat trends. After analysis, it is considered that honeypot data collection can give important insights for most of the cyber threats and can have a mitigating role in all of them, as presented in

Table 1. Particularly, honeypots can play a major role in the analysis of web-based attacks, web application attacks, and botnet threats. They also have an important role when addressing insider threats, cyber-espionage, exploit kits, data breaches, identity theft, denial of service, malware, ransomware, and spam. Nevertheless, honeypots can be less effective against information leakage, phishing and physical threats, and generally against attacks administered by user interactions.

⁶ <https://www.owasp.org>

⁷ <http://isecom.org/research/osstmm.html>

⁸ <https://www.enisa.europa.eu/publications/enisa-position-papers-and-opinions/enisa-recommendations-to-it-industry>

Table 1: Cyber Threat Trends and Potential Role of Honeypot Data Collection

Cyber threat	Description	Honeypot potential
Malware	Malicious software remains the most frequently encountered cyber threat. Evolved techniques (including click-less and file-less infections, worm-based spreading, hybrid attacks, wiping of traces, different infection vectors, and obfuscation-based resistance against heuristic blocking) make malware difficult to resist.	Honeypots have an important role in detecting new malware. By playing the role of a vulnerable host to be infected, honeypots can collect and observe malware in action.
Web-based attacks	Attacks against web servers or web application servers are often used in combination with attacks. For example, compromised servers enable malware infections and provide control points for other compromised nodes.	Honeypots can study attack vectors against and from honeypot web servers. For instance, honeypots can monitor adversary's reconnaissance techniques and adversary's control channels. Honeypots may also discover previously unknown vulnerabilities - zero-day attacks - by real-time monitoring and quick fingerprinting of successful attacks.
Web application attacks		
Phishing	Phishing is a social engineering attack that often relates to different technical means. Adversaries may e.g. utilize malware to mislead victims or capture web servers for to send mass phishing e-mails or to provide fake sites.	As phishing typically involves sophisticated end-user actions, honeypots cannot represent the adversaries' primary targets the users. Instead, honeypots' role lies in secondary phases of the attack e.g. in luring adversaries to compromise honeypot server to deploy phishing sites.
Spam	Unsolicited emails have recently reduced in numbers but still more than half of the emails are spam. Spam has also improved in quality as better obfuscation techniques have made it more difficult to detect. Adversaries often utilize captured devices (also honeypots) for spamming.	Honeypots provide a mean to track adversaries (by following where the control messages come) as well as to learn how the spam is generated in order to create effective filtering solutions.

Cyber threat	Description	Honeypot potential
Denial of service	Denial and Distributed Denial of Service (DoS, DDoS) attacks are a major threat against different online businesses. They have also been taken more seriously e.g. due to recent large botnet attacks and emergence of DDoS-as-a-service providers.	As availability related attacks are typically executed from captured devices, honeypots are a good tool for learning and mitigating them. Honeypots can e.g. find control servers and channels as well as identify targeted victims to enable early warnings and mitigation actions.
Ransomware	Malware that encrypts victim's data for blackmailing has become a prominent threat in the recent years.	A honeypot may have a role e.g. in exploring ransomware's distribution servers.
Botnets	Botnets - a network of captured nodes running automated attack software (robots) - is a threat that is utilized e.g. in DoS or fake advertisement hits. Recent IoT botnets like Mirai and Reaper have demonstrated how massive amounts of vulnerable low-cost things can be captured and harnessed into a malicious botnet. A recent trend is that also virtualized nodes are being captured.	Honeypots, which pretend to be vulnerable things or nodes, are captured into botnets and can provide valuable information on how devices are captured and what the adversary's purposes are.
Insider threat	Persons with privileges and inside organisation are high-severe and difficult to protect threat as the focus is typically on the perimeter defence.	Honeypots can provide defence against misbehaving or inadvertent users as they may catch insiders snooping and accessing on targets where they should not be.
Physical manipulation/ damage/ theft/ loss	Unauthorized manipulation of hardware and software, or theft/loss of hardware and software.	Honeypots are typically software products whose purpose is to detect and discover remote attacks. However, physical deception techniques may be applied to protect against local attacks, physical manipulation. Adversaries may e.g. perform some reconnaissance operations remotely against a honeypot that will guide the adversary to wrong physical location. Deceptive software honeypots may e.g. provide misleading information on assets or defences of particular physical machine.
Data breaches	Data can be stolen via various attacks and must hence be protected in different layers throughout the whole life cycle. EU General Data Protection Regulation (GDPR) emphasizes the risk of breaches for companies.	Honeypots provide a clear indicator on data breaches: as no one should have authorized reasons to access a honeypot, all honeypot accesses are real alerts.

Cyber threat	Description	Honeypot potential
Identity theft	Obtaining and using confidential information in order to impersonate a person or system is a special case of data breaches that are increasing every year.	Honeypots can pretend to be a source of confidential data in order to lure adversaries
Information leakage	Data collected by big internet companies and business data stored by companies may leak due to hostile or inadvertent actions of insider.	Honeypots cannot prevent leaking of data that is already in leaker's possession but monitoring of outbound traffic from honeypots reveals attempts to collect restricted material.
Exploit kits	Exploit kits are a form of web-based attacks where malicious or infected web server attacks vulnerabilities in browsers.	A honeypot server can detect if its capturer is distributing exploit kits. Thus honeypot provides a first hand place to collect new exploit kits and learn their mechanisms. On the other hand, one could also depict use of deceptive technologies in the browser side: vulnerable looking web browsers could search web to find malicious servers.
Cyber-espionage	Spying performed by nations or competing companies is difficult to prevent and detect when the adversaries are very sophisticated - when attacks are classified as Advanced Persistent Threats (APT).	Honeypots provide some changes to catch and monitor these stealth high-risk adversaries who have already circumvented other defences.

3.2.9. Microsoft Security Development Lifecycle

The Security Development Lifecycle ⁹(SDL) is a software development process that helps developers build more secure software and address security compliance requirements while reducing development cost.

3.2.10. MITRE ATT&CK framework

The MITRE Corporation is a not-profit-organizations that work in several areas as artificial intelligence, cyber resilience and cyber threat sharing. One tool released from MITRE is ATT&CK framework. This framework is useful to implement a behavioural based threat model to identify detection capabilities of our infrastructure using adversary emulation.

⁹ <https://www.microsoft.com/en-us/sdl>

This framework is useful for categorizing and characterizing adversary activities in several stages of the cyber attack lifecycle.

The knowledge base of the framework is presented in a matrix where has described the adversary tactics and techniques used by attacker, based on real-world observations of cyber attack. This tool can be used to develop a specific threat model and related test, that represent which attacker tries to attack our infrastructure and its goal.

ATT&CK framework ¹⁰is useful, as well as to evaluate of detection capabilities of sensors, also for prioritize defences.

¹⁰ <https://attack.mitre.org/>

4. COLLECTION OF ASSESSMENT METHODS AND TOOLS

In order to respond to security weaknesses RESISTO proposes a number of long term activities that should be part of the operator constant security assessment of the telecom infrastructure to identify actions that can be put in place before the actual attack occurs. If we have to sum up in just one sentence what a security methodology we can propose is “Know yourself but also your enemy”.

Several methods can be applied to analyse a threat typically with two main goals:

- re-conduct the threat to its root causes e.g. SW, HW, procedural single vulnerabilities and combinations
- identify attack typical events sequences/relations .

The proposed methodologies shall be included in the long term control loop but should be performed regularly in particular need to be aligned with the changes the network undergoes when new sw or hw is included in the network operator infrastructure. Every time a new release of a product or significant change is made to the telecom infrastructure there is a need for an assessment to verify that the change is not jeopardizing the network stability and security or resilience. The techniques suited to be considered in the long term control loop to make sure the operator knows in advance the possible weaknesses that are introduced in the network because a new sw or hw product is made part of its network can be assessed using Penetration testing. Penetration tests should be part of the cyclic security activity of a Telecom operator in particular they should be performed when new hw and sw is introduced or sw is upgraded. The scope of a Penetration Test (PenTest) plan already includes a number of test that aim to find the security weaknesses of a computer system and intentionally attack the system. Pen Test plans and tools on the other side should be selected based on the knowledge of the potential attacks that the system can undergo to.

Therefore in order to make a good plan there is the need for techniques that can be used to describe how an attacker could possibly attack a system. RESISTO proposal is to use the following: attack trees, MITRE ATT&ACK and Honeypots.

Of course RESISTO will report suggestion on how to use those techniques, but the knowledge gained using them shall be used to improve the coverage of the penetration tests.

Regarding the attack trees in the rest of the document a number of techniques will be presented that are stemming from this idea but RESISTO believes that attack defense trees can incorporate attack and defense roles creating a more dynamic description of the step progression of an attack and on the other side what kind of defense tactic can be put in place to mitigate each action performed by the attacker.

Furthermore in order to keep an up to date description of potential attacks and identify beforehand countermeasures RESISTO suggests to include in the long term control loop the analysis of attack coming from data collected by honeypots and MITRE ATT&CK methodology.

MITRE-ATT&CK (where the latest means Adversarial Tactics, Techniques, and Common Knowledge) have been used to document common tactics, techniques based on real-world observations of advanced persistent threats (APT) used against the most common platforms (i.e. Microsoft Windows mainly, but also Linux and Mac OSX platform) therefore they can be used to create a knowledge base and a model for attacker behaviour in several phases of attack lifecycle.

Honeypots represent HW and/or SW modules simulating exposed parts of the telecom network whose main purpose is to collect data about potential intrusions which shall not damage any legitimate function of the system. The observation made during the attack will be saved to acquire knowledge about new threats but also to identify known ones. The collected info can be fruitfully exploited in the long-term loop to plan and must be integrated in the way RESISTO detects threats and decides the most appropriate mitigation actions.

Moreover the output of this long term control activity shall be considered in the definition of short term control loop to assess the event correlation and the mitigation actions performed by the corresponding modules of RESISTO namely the Event correlator and the workflows implemented in the Workflow Manager.

4.1. Deductive approaches

These methods use a graphical modelling of a threat by means of each sub-components relationship i.e. vulnerabilities on which the attacker can leverage, attacker actions/events/consequences and asset involved. The models can either focus on the dependencies between elements e.g. « what happens if » or on possible chains of observable events. The methods can also be represented tabularly, referring to as tabular deductive assessment.

4.1.1. Attack trees

An attack tree is a tree-like representation of an attack scenario. This modelling was made popular by Schneier inspirational work (Schneier, 1999) as a tool to evaluate the security of complex systems. The root of an attack tree corresponds to an attacker's goal. The children of a node in the tree are refinements of the node's goal into subgoals. The leaves of the tree are the actions to be executed by the attacker. Below the classical example from Schneier's paper (Schneier, 1999) for an attack to a physical safe:

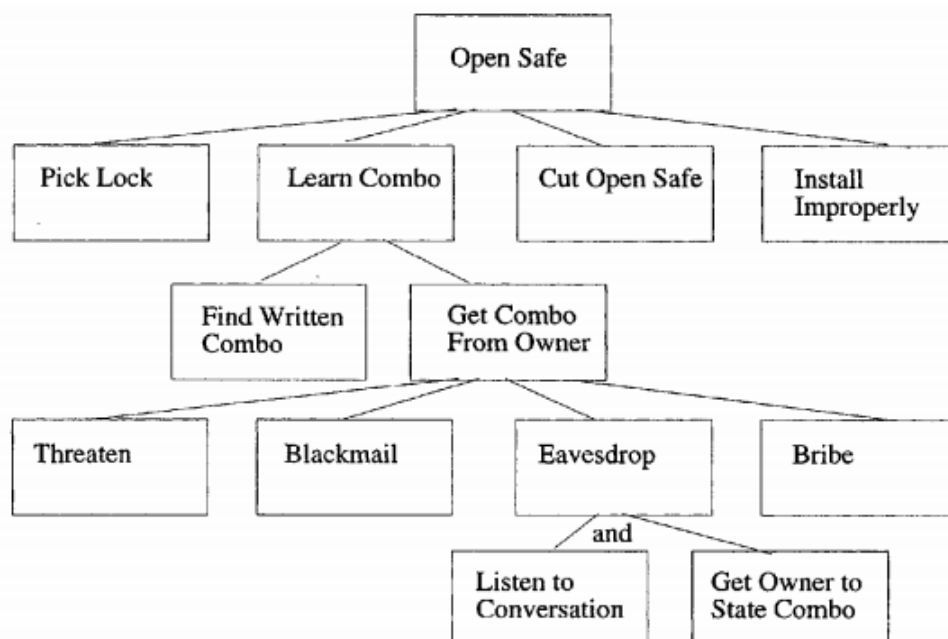


Figure 4: Physical Safe - Attack tree (Schneier, 1999)

4.1.2. Stuxnet attack tree example

Stuxnet was a computer worm targeting Iranian nuclear enrichment facilities, its ultimate goal was to damage the centrifuges used in the enrichment process (Chen, 2011). The worm attacks industrial control systems by modifying the code on programmable logic controllers (PLCs). PLCs are computers made specifically for automation of industrial systems.

Below a possible attack tree (Ola Flaten, 2014):

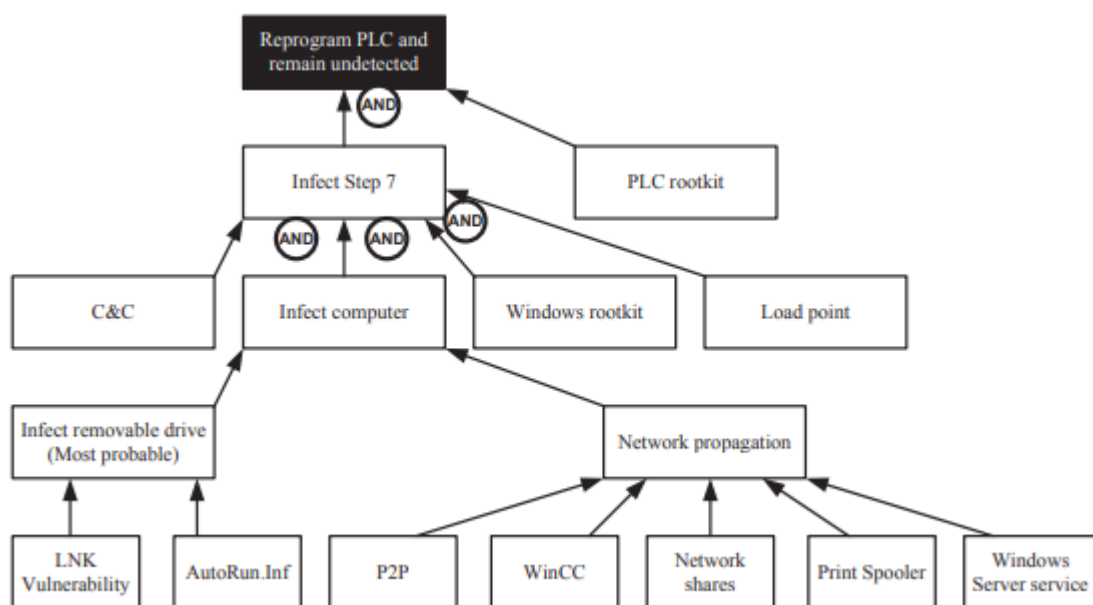


Figure 5: Stuxnet attack tree example (Ola Flaten, 2014)

In the above example a leaf represents not only an attacker action but also a potentially vulnerable entry point on which it could have leveraged on.

4.1.3. GhostNet attack tree example

GhostNet was a cyber-espionage network uncovered in March 2009. The network consisted of 1295 infected computers in 103 different countries; up to 30 % of them high-value targets (Markoff, 2009). The attack was very complex and accomplished in several steps distributed over a long period. A tree modelling could appear as following (Ola Flaten, 2014):

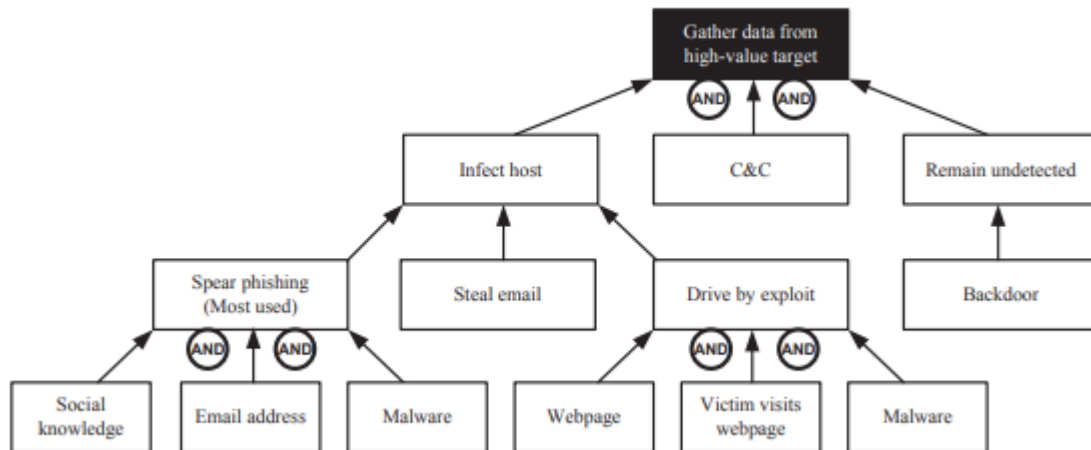


Figure 6: GhostNet - Attack tree example (Ola Flaten, 2014)

4.1.4. Dynamic attack tree

The original attack tree does not describe if the basic steps (attacker actions) are performed in sequence or as alternative of each other, for example if an attacker needs to scan the system before to exploit a vulnerability. The dynamic attack tree (Florian Arnold, 2015) tries to model also this aspect inserting an order indication (an arrow between leaves or inside the OR or AND symbols). The Stuxnet attack tree would then represented as following:

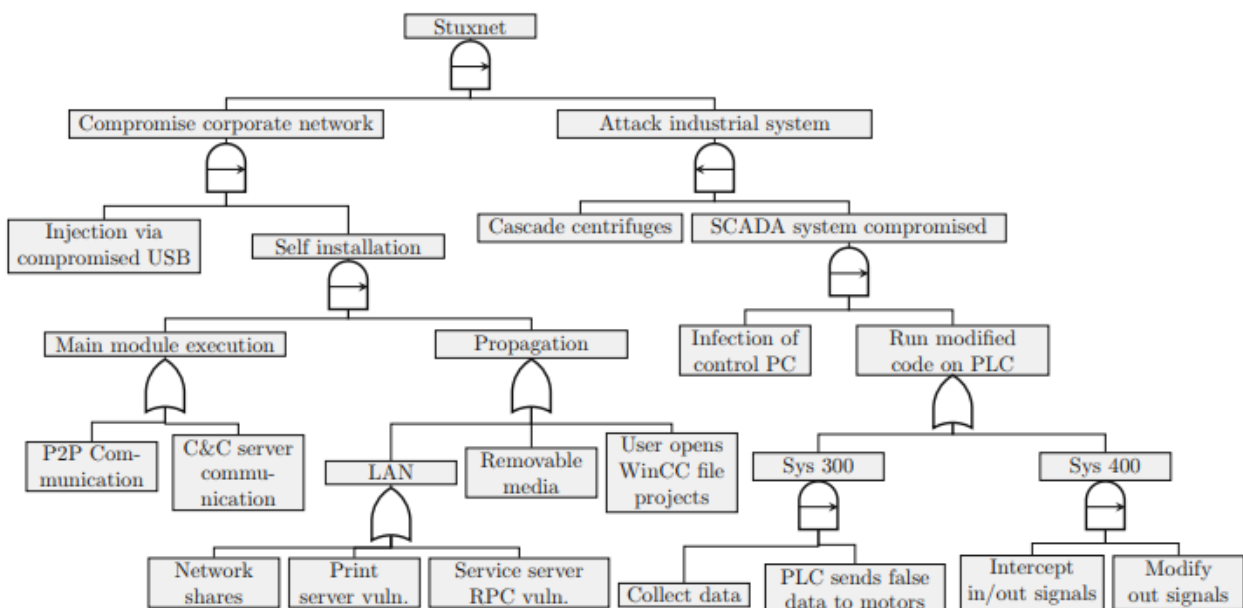


Figure 7: Dynamic tree example: Stuxnet (Florian Arnold, 2015)

Beside graphical aspects the dependencies represented in the tree help to more rapidly assess the attack and to identify the probability of the various attack paths in the tree. The long-term loop can likely get most benefit from this approach detecting early indicators of APT attacks as it works on a larger set of data collected over time and can make a deeper analysis of it.

Another simple example (Ludovic Piètre-Cambacédès, 2010) of attack tree is the case of RAS malicious access threat. In this case, in addition to probable attack steps sequences, there are the indicators of the max time expected to be spent in each step. E.g. an attacker can decide to try for 1 hour a password cracking attack then, before getting noticed by the short-term loop, changes strategy and tries to exploit SW vulnerabilities. The attack would then likely be detected in the long-term loop instead as it considers also timing aspects.

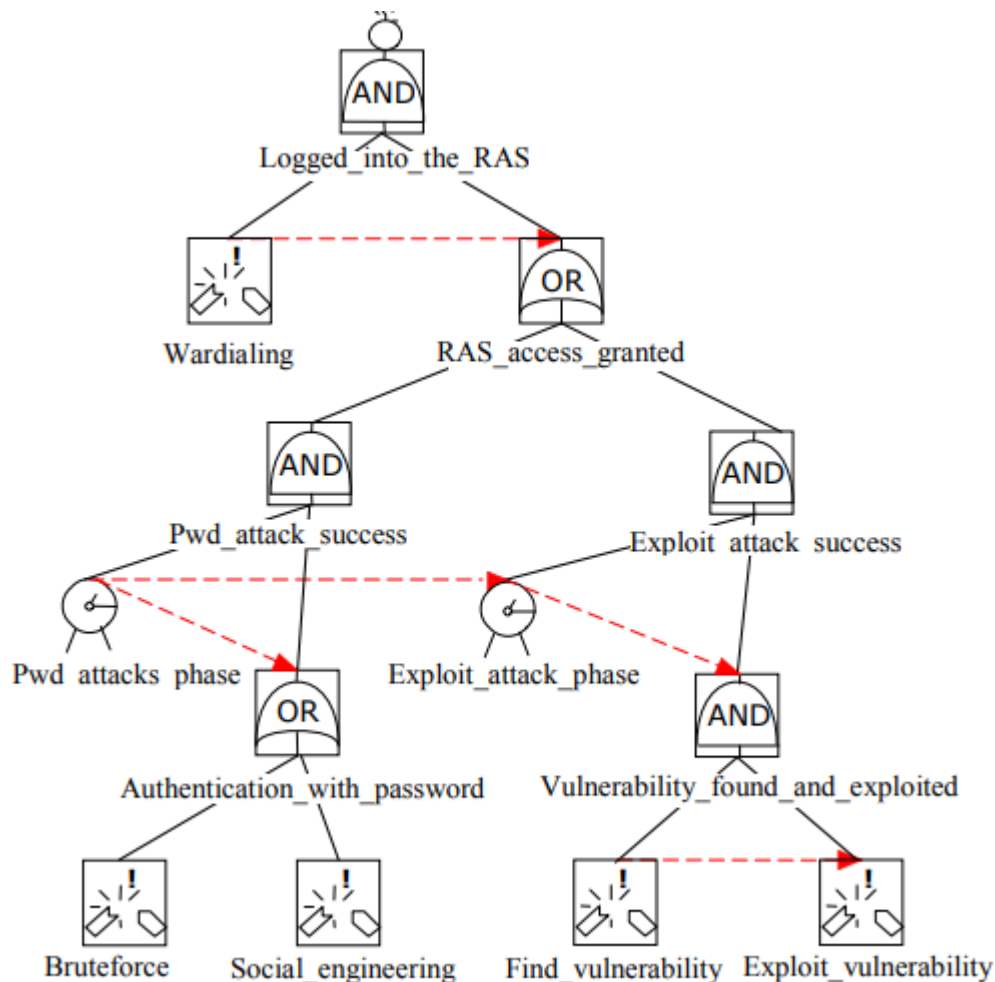


Figure 8: Dynamic Attack Tree: RAS malicious access example (Ludovic Piètre-Cambacédès, 2010)

4.1.5. Combined attack tree

Multiple threat scenarios can have common steps in an attack path and can influence each other. Such case can be analysed using combined attack trees which are able to represent cross-connections between attack trees. A simple example is reported below (Marco Gribaudo, 2015).

Note: The addition of each step attack probability makes it possible to assess both single vulnerability issues aspects and whole attack sequences.

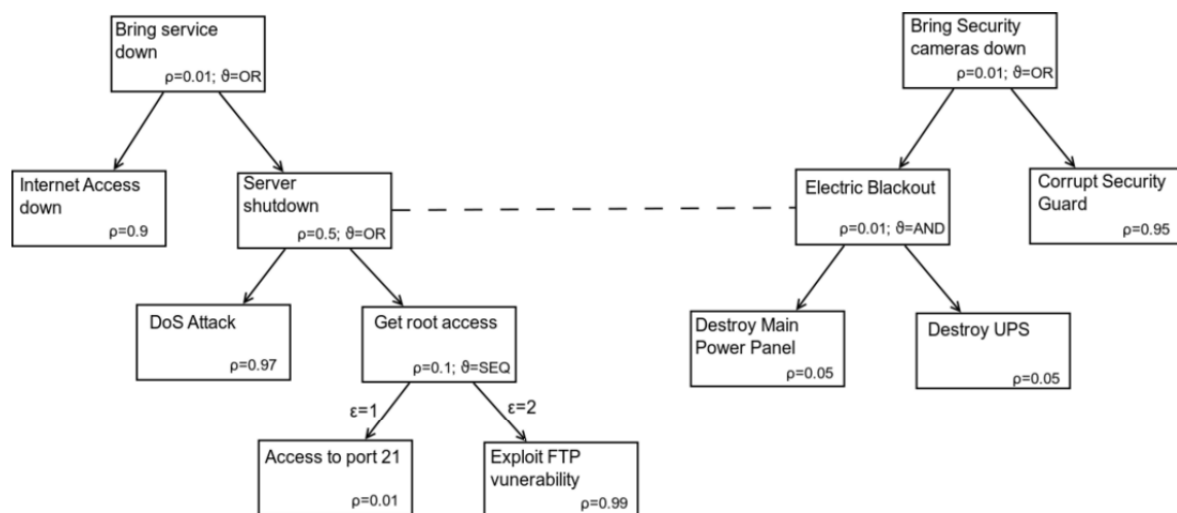


Figure 9: Simple example of combined attack tree (Marco Gribaudo, 2015).

In the same paper (Marco Gribaudo, 2015) the much more complicate example shown in Figure 10 is described. It connects different threat scenarios from railway, maritime system, data networks and biochemical areas. It is very interesting in the long-term loop context as it gives an example of a complex cyber-physical risks assessment where a threat can be a combination of heterogeneous domains threats or, on the converse, where a single threat in a domain can impact other cyber and physical domains.

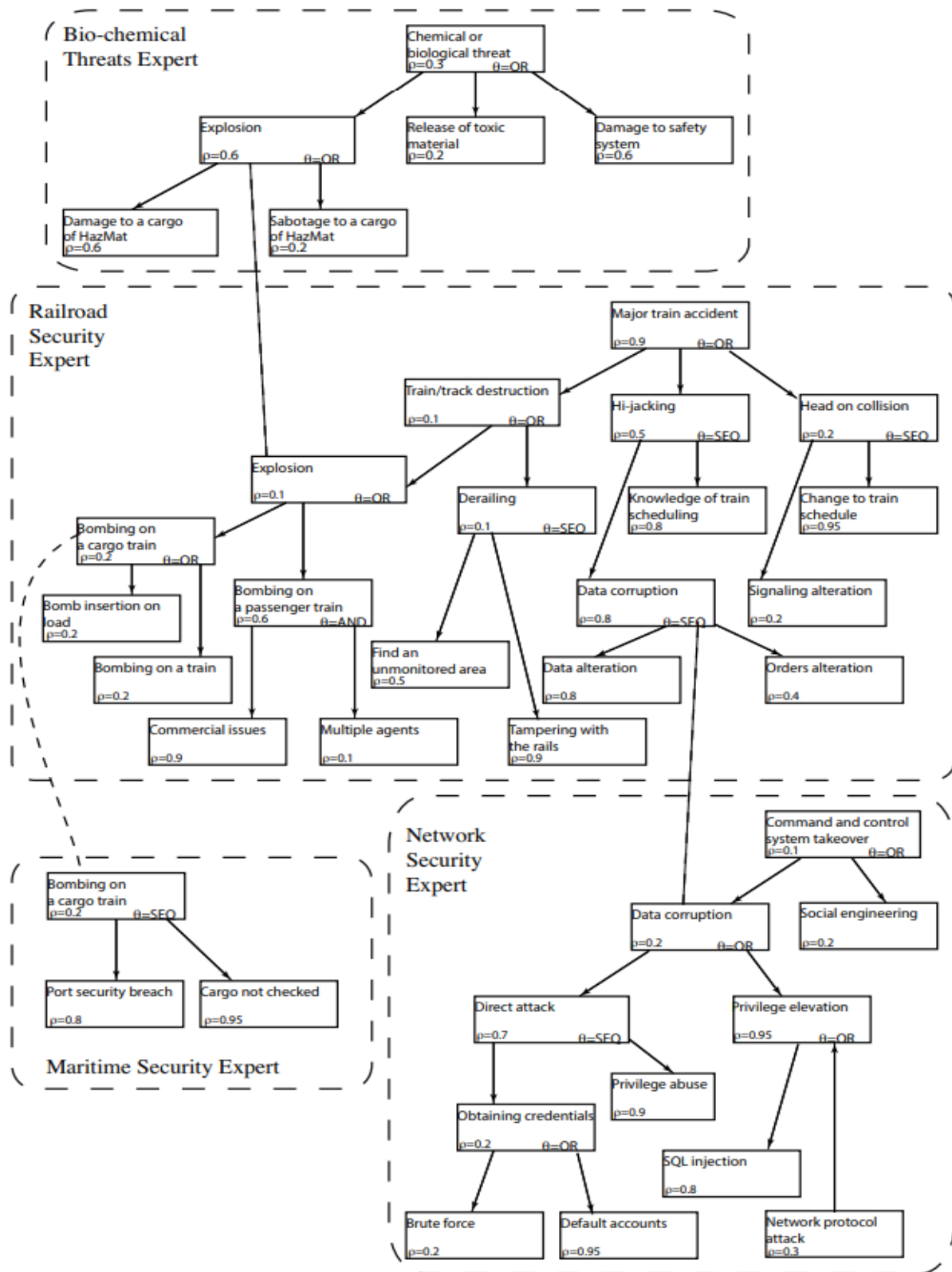


Figure 10: Multiple Domain Combined Cyber-Physical Attack tree (Marco Gribaudo, 2015)

4.1.6. Attack-Defence tree

A further improvement in risk assessment could be achieved if the attack tree model is cross-connected with the defence tree i.e. the set of related countermeasures, mitigations and recovery actions already in place in the system to assess. An example comes from (Barbara Kordy, 2012) work.

In below figure the goal is to protect the company data confidentiality. The green box represents the main security areas and tools except for the ones connected with dotted lines which are specific countermeasures to the attacker actions (red circles).

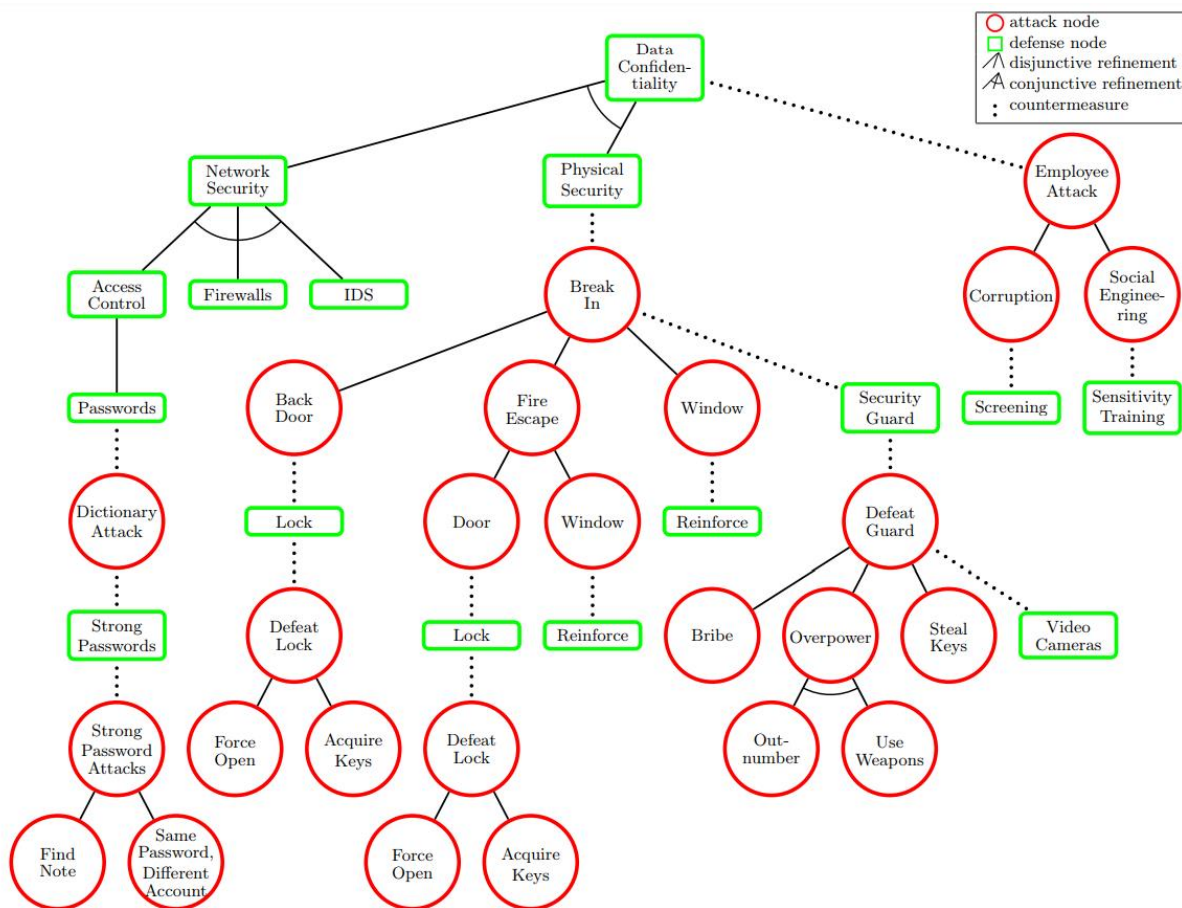


Figure 11: Attack-Defence trees (Barbara Kordy, 2012)

4.2. Honeypots

A honeypot is a set of physical, HW and/or SW modules simulating legitimate interactions with external users while they are instead separate entities not performing any real operation and/or handling real data.

The honeypot is a long time used technique to detect and analyse sophisticated intrusions while keeping the real system safe. Moreover, it provides the possibility to observe an attack over time enabling the system both to learn about new threats and to assess known ones. Since it collects detailed historical information, the technique is particularly useful for the long-term loop purposes.

As from its name “honeypot” it is made very attractive for an attacker by appearing as the most vulnerable part of the system to protect (e.g. with many open ports, no secure protocols required, unpatched SW).

However, honeypots must not be considered self-contained but always integrated in a security system which correlates/complements the information with other security monitoring sources and decides mitigation actions.

There are many different honeypots. Below a classification based on four main characteristics:

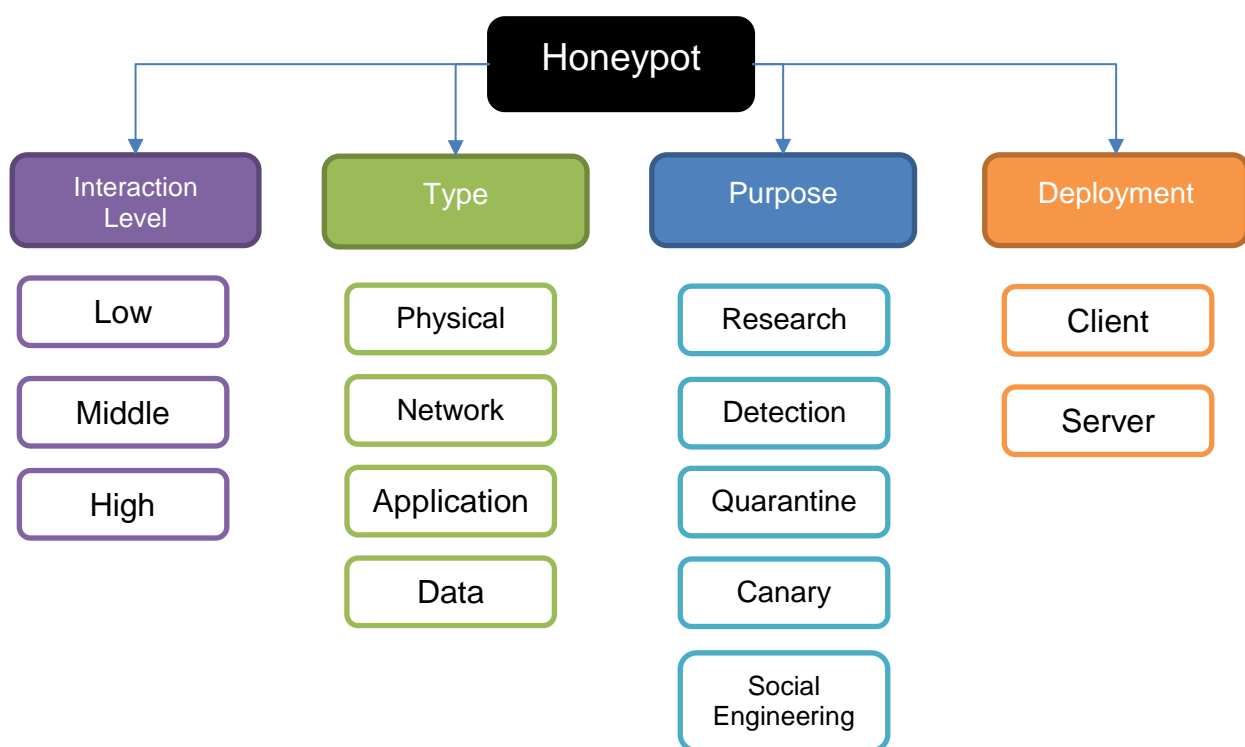


Figure 12: Honeypot characteristics

Interaction level is intended as the capability of simulate part or all the system functionalities. More in detail:

- **Low Interaction:** Only the minimal functions are emulated e.g. login access, secure protocol authentication (e.g. SSH), file system navigation and the related events logged/sent to a central SIEM
- **High Interaction:** it mimics the whole system behaviour and can be difficult to detect by an attacker unless analysing the actual effects on the real system before and after an interaction. It is very effective in finding new types of threats especially the APT's which require a long observation time without revealing the honeypot nature
- **Medium Interaction:** it usually characterizes honeypots which simulate well one or few behaviours of the system targeting specific kind of attacks.

There is a distinction in the way the honeypot is deployed and works. In particular:

- **Server Honeypot:** it is installed as additional component/s of the server system and mimics its behaviour in serving the requests from an external user or device. It is the most common honeypot deployment and can act passively, detecting anomaly traffic, or actively, reacting against the attacker e.g. re-directing the malicious traffic to a sink or mimicking a dummy answer that leads the attacker to another system/honeypot for further analysis
- **Client Honeypot:** when a client application is the target to be protected, e.g. a mobile application holding personal data, the honeypot can mimic its client behaviour towards all the available servers. The purpose is to verify if one of the servers is malicious by observing if any of them try to perform an attack e.g. encryption level bid-down, malware injection, client session hijacking or sensitive data exfiltration. A typical example in the web client application area is the honeypot variants based on web spiders or web crawlers.

The following subsections describe the type and purpose characteristics together with some examples.

4.2.1. Physical Honeypots

The honeypot is a physical entity, HW/SW host or just a sensor simulator acting as a legitimate appliance providing some services alone (e.g. an info kiosk or a fake temperature/gas sensor). The honeypot is made attractive by making it visual evident and easy physically accessible (e.g. usually placed in a publicly accessible place with exposed physical ports or reachable via internet).

An example is the deployment of a fake radio antenna or, more commonly, a Wi-Fi access point presented as legitimate entry point of a company's infrastructure network access, which instead connects the user to network/host honeypot. To make it attractive it has high signal strength (higher than any other in the area) and no or a weak protection protocol like EAP, WPA, WPA2.

A war-dialling attack will then be discovered by letting the attacker access to the fake access point and assigning it a dummy IP. The attacker is not able to go any further (e.g. internet connection is not available and no/limited internal network is visible). On defence side, the access event is collected and notified to one or more security systems according to the local security strategy.

As example, this kind of honeypot can be easily built from a Linux popular open source router:

- <https://www.linuxjournal.com/content/wi-fi-mini-honeypot>

Another example is the Bluetooth honeypot in bank payment context:

- https://www.researchgate.net/publication/224114802_Securing_Bluetooth-based_payment_system_using_honeypot

In the IoT domain any electrical appliance (e.g. a washing machine, oven, fridge, security camera, etc.) can expose an access port (usb, ethernet) or be reachable by internet. It is becoming a common strategy to modify one of them to act as a honeypot and put it in a location where it is expected to be attacked.

One example was published in a recent news where a fake electric substation installed in the electrical network of an energy operator resulted in being a target of many hacker attacks in just few days:

- <https://www.zdnet.com/article/hackers-found-and-cracked-this-fake-electricity-substation-network-in-just-two-days/>

The above example demonstrated the feasibility of the attack and allowed the operator both gathering information and ranking the risk of actually interesting targets for an attacker.

A scalable IoT honeypot physical framework is also envisioned in the SIPHON architecture (J. D. Guarnizo, 2017).

4.2.2. Network Honeypots

The network honeypots are the most common case, since IDS (Intrusion detection systems) were the first to use them. They are basically realised as specific devices attached to the network (usually in DMZ) and available to internet (e.g. as router, access authentication/authorization systems, proxy etc.) or they can just be a SW daemon on top of existing network SW appliances offering a common protocol service handler like for SNMP, SSH, IPSEC etc.

A popular open source project:

- <http://www.honeyd.org/concepts.php>

and, scaling up the example, a whole network can be used as honeypot:

- <https://www.honeynet.org/>

A list of SSH, telnet open source honeypots:

- <https://linuxsecurity.expert/security-tools/honeypots>

4.2.3. Application Honeypots

The application honeypots simulate a specific application behaviour which, due to its sensitivity (e.g. bank payment access, energy appliances manager), require a high level protection.

It is specialized on the features the application provides with a high interaction level because it must gather meaningful information while the trapped attacker must think as long as possible to interact with a legitimate server.

The application honeypots can also act as preliminary checker and proxy the request to the user only once the honeypot has verified there is no actual attack risk.

Some examples from open source communities:

- <http://conpot.org/> CONPOT a honeypot for ICS/SCADA system
- <https://github.com/mushorg/snare> SNARE and advanced honeypot for Web applications

4.2.4. Data Honeypots

Data Honeypots are dummy documents which contain fictitious sensitive data (e.g. password, identities, credit cards numbers, and firewall configuration info) in clear text and stored in various places both in highly restricted and unrestricted areas across the system to protect.

The access to this data is monitored by agents or by the application used for reading the data (database engine, content management system, word application configured ad hoc) or a centralized controller. Each access event is reported and analysed, eventually correlated with other security events e.g. like the tentative to use the password for accessing a service.

Sometime the honeypot password data are made easily available to an attacker with the intent to invite it to use the credential to access to a network or application honeypot.

Some examples:

HoneyDoc - detect access to dummy doc claimed as sensitive

- <https://github.com/jgcreator/honeydoc>
- <https://www.blackhillsinfosec.com/bugging-docx-files-using-microsoft-word-part-1/>

HoneyToken and HoneyBits – spread fake access credentials/token across the system in order to lure the attacker towards your honeypots or monitor the failed access tentative using these credentials.

- <https://www.symantec.com/connect/articles/honeytokens-other-honeypot>
- <http://www.eurecom.fr/en/publication/1275/download/ce-pougfa-030914b.pdf>
- <https://github.com/0x4D31/honeybits>

4.2.5. Research Honeypots

The research honeypots are used for information gathering only. They are usually specialised in finding new threats via a deep analysis of data and usage of AI aided tools. They can be located either in the front of the system (e.g. DMZ area, Firewall) or inside it to detect complex internal attacks. As it requires a high computation capability, in many cases they are limited to collect, filter, transform, compress data and send them to a centralized system which perform real-time and batch computation.

A typical application are the honeypot agents distributed on internet and connected to 24/7 centre of big security companies like Symantec, fSecure, TrendMicro, etc. with the purpose to discover new malware infections and APT flow patterns worldwide.

The result is summarized in worldwide threat status dashboards like <http://www.digitalattackmap.com>

As it works on large and historical set of data a research honeypot fits well our long-term loop goals.

4.2.6. Production Honeypots

The purpose is to fast detect/prevent/react threats in real-time. These kinds of honeypots are usually integrated or complementary to the IDS/IPS infrastructure of the system to protect.

The objective is to serve as a system entity attracting initial attacks and let the security system react before the attack reaches real modules or sensitive data. Considering RESISTO architecture, production honeypots are mostly candidate for deployment in the short-term loop.

There are many example of IDS integrated honeypot. The most common ones are based on the popular open source SNORT IDS <https://www.snort.org/>. Below are two articles about integration examples:

- http://www.academia.edu/1074906/Honeypot_IDS_SNORT_Intrusion_Detection_System
- <https://ieeexplore.ieee.org/document/7409013>

An interesting commercial honeypot and IDS system for windows is:

- <http://www.keyfocus.net/kfsensor/>

4.2.7. Quarantine Honeypots

The quarantine honeypot is a specialised host/SW instance that serves, as first entry point, a request from a user, sends back a dummy reply/ack but proxies it to the real system only after a configurable time interval and/or a safety analysis/run in an isolated environment.

This is usually done for requests which do not require a complex user interaction like a file deliver request where the sender just gets back an ack while the received file is instead put in quarantine for malware scan and observation period before actual processing.

The honeypot is nowadays particularly popular for usage in e -mail antispam systems. Some examples and architectural description are given at the following links:

- https://www.researchgate.net/publication/229042409_Quarantine_Net_design_and_application

Open source:

- <https://github.com/msurguy/Honeypot>
- <https://github.com/ianlandsman/Honeypot>

Commercial:

- <https://www.symantec.com/connect/forums/analyse-email-quarantine>
- https://www.ibm.com/support/knowledgecenter/en/SSFS6T/com.ibm.apic.devportal.doc/topic_portal_honeypot.html
- <https://www.ostraining.com/blog/coding/honeypot/>

4.2.8. Canary Honeypots

The canary is a recurring pattern in security defence techniques. The name is derived from the use of canary birds by miners who left a living bird at the mine entrance to detect toxic gas presence. If the canary died they got aware of the danger.

In honeypot context the canary is a digital application/host/HW and any malfunctioning behaviour is used as an indicator of an on-going attack or data manipulation. This means that all the traffic shall pass first via the canary application and then to the real system. The real application needs then to monitor the canary honeypot key performance indicators and stop processing or performing other corrective actions if they go below a certain threshold.

The advantage of this technique is that the honeypot can be a fake instance of the same SW used for real systems, but working on dummy data. Therefore there is no need to develop an interaction facade with a dedicate application mimics. Below are several examples and source links:

- <https://www.oreilly.com/library/view/applied-network-security/9780124172081/xhtml/CHP012.html>
- <http://docs.opencanary.org/en/latest/>
- <https://github.com/thinkst/opencanary>

A simple but powerful variant of canary honeypot is the “canarytoken” i.e. a honey token which can be embedded in a document (e.g. word, pdf, web page) and automatically check and report to a central monitoring system if it is accessed. It can be seen as an application of honeydoc explained above.

An interesting example of a canary token generator can be found at the following link:

<https://canarytokens.org/generate>

4.2.9. Social Engineering Honeypots

The evolution of social network applications on a large scale widens the possibilities to use them as honeypot also to collect information. The model consists in creating fake social profiles to interact with others in order to gather information and/or to detect anomaly behaviours.

They can be very powerful to prevent many cybersecurity attacks, however, the usage of this kind of honeypot must be very careful and supervised by officers as it heavily involves collection of personal identifiable information. Another main risk is that can be easily used also by an attacker.

We can distinguish two types of honeypots:

A “server model” where a social profile or site is made attractive to receive interactions from targeted persons and gather information for cybersecurity purposes.

A “client model” where a honeypot profile tries to interact with targeted profiles to both gather information but also to discover impersonations (e.g. as said before social honeypot used by the attacker).

The artificial intelligence is playing a fundamental role in enhancing the level of interaction of social engineering honeypots which can replicate many human behaviours in online interactions (e.g. via chatbot)

4.3. Penetration test assessment

A penetration testing (pentest) is a combination of techniques that considers various issues of the systems and tests, analyses, and gives solutions. It is based on a structured procedure that performs penetration testing step-by-step.

The National Cyber Security Center, describes penetration testing as the following: "A method for gaining assurance in the security of an IT system by attempting to breach some or all of that system's security, using the same tools and techniques as an adversary might." (NCSC, 2017)

Penetration tests are a component of a full security audit.

The process typically identifies the target systems and a particular goal, then reviews available information and undertakes various means to attain that goal.

Penetration testing is not, however, a straightforward process. It is often very technical in nature and uses very challenging methods and processes and tools.

Furthermore, organizations have reported a number of difficulties when conducting penetration tests, which include:

- Determining the depth and breadth of coverage of the test
- Identifying what type of penetration test is required
- Managing risks associated with potential system failure and exposure of sensitive data
- Agreeing the targets and frequency of tests
- Assuming that by fixing vulnerabilities uncovered during a penetration test the systems will be secure.

4.3.1. Goals

The goals of a penetration test vary depending on the type of approved activity for any given engagement with the primary goal focused on finding vulnerabilities that could be exploited by a nefarious actor and informing the client of those vulnerabilities along with recommended mitigation strategies.

Undertaking a series of penetration tests will help test security arrangements and identify improvements. When carried out and reported properly, a penetration test can give knowledge of nearly all technical security weaknesses and provide the information and support required to remove or reduce those vulnerabilities.

Research has shown that there are also other significant benefits to organisations through effective penetration testing, which can include:

- A reduction in ICT costs over the long term
- Improvements in the technical environment, reducing support calls
- Greater levels of confidence in the security of IT and OT environments
- Increased awareness of the need for appropriate technical controls.

4.3.2. Types of Penetration Tests

Network Pentest - The primary objective for a network penetration test is to pro-actively identify exploitable vulnerabilities in networks, systems, hosts and network devices (i.e.: routers, switches). Network penetration testing will reveal real-world opportunities for hackers to be able to compromise systems and networks in such a way that allows for unauthorized access to sensitive data or even take-over systems for malicious/non-business purposes.

Web App Pentest - A web application security test focuses only on evaluating the security of a web application. The process involves an active analysis of the application for any weaknesses, technical flaws, or vulnerabilities. Any security issues that are found will be presented to the system owner, together with an assessment of the impact, a proposal for mitigation or a technical solution.

Wireless Networks Pentest - The goal of a Wireless pentest is to identify all exploitable vulnerabilities and misconfigurations in all wireless networks (Wi-Fi, 2G, 3G, 4G), on all stack levels. This type of penetration test usually involves in-depth knowledge of multiple technologies and the use of specifically crafted tools in order to find flaws on all layers of the communications stack. For a telecommunications operator, this type of pentest is conducted on the physical transport layers on the IP layer, on the signalling and control layer and finally on the application layer. Specific technologies and protocols such as SS7, SIGTRANS, Diameter are in the extended scope of this type of Pentest.

OT & PSIM (Physical) Pentest – The goal of a OT Pentest is to evaluate the security of the attack surface, identify all exploitable vulnerabilities and audit all misconfigurations. The audit should consist of procedures and techniques necessary to discover and identify network elements such as Controller Machines, PLCs, End Device (sensors, valves, pumps etc.) and to assess how the OT network is segregated from the rest of the (IT) networks.

In realistic environments, this pentest will usually be conducted on 'live' production networks as most organisations seldom have a quality assurance or test environment for OT deployments,

During and OT Pentest, the auditor will usually perform several checks against the attack surface such as:

- discovery of devices and equipment with factory credentials enabled and unchanged;
- PLCs that allow access from any device, not only from authorized machines (whitelisting);
- misconfigured network segregation (i.e – the auditor can reach the OT network from the IT networks);
- Internet access availability on the controller machines;
- Clear text services running on the OT network;
- Password Policy implementation;
- Patch level and minimum security requirements for the controller machines;

A basic toolkit for assessing OT networks in particular and industrial networks in general can consist of open-source software such as smod – a ModBus pentesting framework, plcscan, mbtget and various NMAP scripts – for discovery and fingerprinting of PLC devices and plcinject – a tool for code injection in PLC devices.

Social Engineering – the scope of a Social Engineering pentest is to ascertain an organization's level of vulnerability to social engineering exploits by attempting to scam the organization's employees or collaborators. A Social Engineering pentest is designed to test employee's adherence

to the organization's security policies, practices and procedures as defined by the management and the resulting report of this type of pentest should provide the organization with information on how easy it is for an intruder to convince employees to break security rules and regulations and to divulge or exfiltrate information not meant to exit the organization's security perimeters.

A Social Engineering test can be conducted as part of a broader scope penetration testing or audit and should replicate the efforts that real-world attackers use.

The most common methods and techniques for testing are phishing exploits and attempts to gather password information from employees by impersonating IT staff.

4.3.3. Penetration Testing Stages

Penetration tests are usually performed in stages with the principal 5 stages as follows (Scarfone, 2008):

Reconnaissance - The act of gathering important information on a target system. This information can be used to better attack the target. For example, open source search engines can be used to find data that can be used in a social engineering attack.

Scanning - Uses technical tools to further the attacker's knowledge of the system. For example, Nmap can be used to scan for open ports.

Gaining Access - Using the data gathered in the reconnaissance and scanning phases, the attacker can use a payload to exploit the targeted system. For example, Metasploit can be used to automate attacks on known vulnerabilities.

Maintaining Access - Maintaining access requires taking the steps involved in being able to be persistently within the target environment in order to gather as much data as possible.

Covering Tracks - The attacker must clear any trace of compromising the victim system, any type of data gathered, log events, in order to remain anonymous.

4.3.4. Penetration Testing Activities

Planning & Preparation

Planning and preparation starts with defining the goals and objectives of the penetration testing.

The common objectives of penetration testing are:

- To identify the vulnerability and improve the security of the technical systems.
- Have IT security confirmed by an external third party.
- Increase the security of the organizational/personnel infrastructure.

Reconnaissance

Reconnaissance includes an analysis of the preliminary information. The tester starts by analysing the available information and, if required, requests for more information such as system descriptions, network plans, etc. This step is the passive part of a penetration test. The sole objective is to obtain a complete and detailed information of the systems.

Discovery

In this step, a penetration tester will most likely use the automated tools to scan target assets for discovering vulnerabilities. These tools normally have their own databases giving the details of the latest vulnerabilities. However, tester discover

- Network Discovery – Such as discovery of additional systems, servers, and other devices.
- Host Discovery – It determines open ports on these devices.
- Service Interrogation – It interrogates ports to discover actual services which are running on them.

Analysing Information and Risks

In this step, tester analyses and assesses the information gathered before the test steps for dynamically penetrating the system. Because of larger number of systems and size of infrastructure, it is extremely time consuming. While analysing, the tester considers the following elements –

- The defined goals of the penetration test.
- The potential risks to the system.
- The estimated time required for evaluating potential security flaws for the subsequent active penetration testing.

Active Intrusion Attempts

This is the most important step of a penetration test. This step entails the extent to which the potential vulnerabilities that was identified in the discovery step which possess the actual risks. This step must be performed when a verification of potential vulnerabilities is needed. For those systems having very high integrity requirements, the potential vulnerability and risk needs to be carefully considered before conducting critical clean up procedures.

Final Analysis

This step primarily considers all the steps conducted (discussed above) till that time and an evaluation of the vulnerabilities present in the form of potential risks. Further, the tester recommends to eliminate the vulnerabilities and risks. Above all, the tester must assure the transparency of the tests and the vulnerabilities that it disclosed.

Report Preparation

Report preparation must start with overall testing procedures, followed by an analysis of vulnerabilities and risks. The high risks and critical vulnerabilities must have priorities and then followed by the lower order.

However, while documenting the final report, the following points needs to be considered:

- Overall summary of penetration testing.
- Details of each step and the information gathered during the pen testing.
- Details of all the vulnerabilities and risks discovered.
- Details of cleaning and fixing the systems.
- Suggestions for future security.

4.3.5. Penetration Testing Programme

A Penetration Testing Programme can be represented by a logical diagram of specific activities, based on one or more methodologies, regulatory, legal and business requirements, see Figure 13.

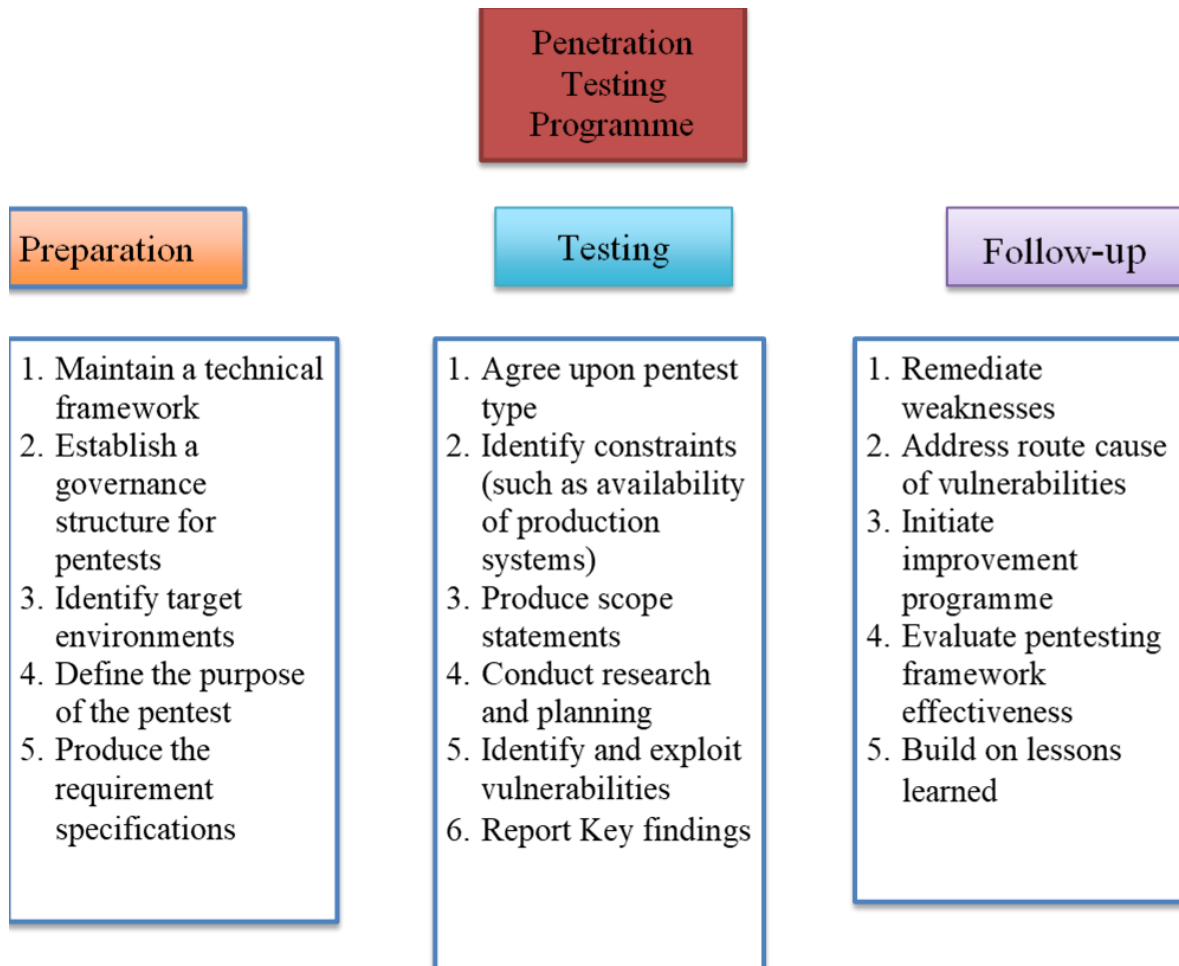


Figure 13: Logical diagram for penetration testing programme.

4.3.6. Penetration Testing Methodologies

Several public-domain, open-source or commercial methodologies and standards can be used to build a penetration testing framework. One organisation can use more than one such methodology, according to their specific needs.

OWASP Testing Guide:

https://www.owasp.org/index.php/Penetration_testing_methodologies

PCI DSS Testing Guide:

https://www.pcisecuritystandards.org/documents/Penetration_Testing_Guidance_March_2015.pdf

NIS 800-115:

<http://csrc.nist.gov/publications/nistpubs/800-115/SP800-115.pdf>

OSSTMM:

<http://www.isecom.org/research/>

4.4. MITRE ATT&CK

MITRE started this project in 2013 with the objective to document common tactics, techniques based on real-world observations (Strom B. E., 2018). Initially MITRE started the observation to document the advanced persistent threats use against Microsoft Windows enterprise network, actually in the ATT&CK Matrix (an extraction in figure below), we can find also tactics and techniques used against Linux and Mac OSX platform.

ATT&CK stands Adversarial Tactics, Techniques, and Common Knowledge and is a knowledge base and a model for attacker behaviour in several phases of attack lifecycle.

With this framework we can describe and define a characterization of adversary behaviour, after initially access to our protected system, gaining access via a successful exploit.

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise	AppleScript	bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Data Destruction
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	BITS Jobs	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Encrypted for Impact
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	Binary Padding	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Connection Proxy	Data Encrypted	Defacement
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data Staged	Custom Command and Control Protocol	Data Transfer Size Limits	Disk Content Wipe
Replication Through Removable Media	Control Panel Items	AppInit DLLs	Application Shimming	CMSTP	Credentials in Files	File and Directory Discovery	Logon Scripts	Data from Information Repositories	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Structure Wipe
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Bypass User Account Control	Clear Command History	Credentials in Registry	Network Service Scanning	Pass the Hash	Data from Local System	Data Encoding	Exfiltration Over Command and Control Channel	Endpoint Denial of Service
Spearphishing Link	Execution through API	Authentication Package	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Network Share Discovery	Pass the Ticket	Data from Network Shared Drive	Data Obfuscation	Exfiltration Over Other Network Medium	Firmware Corruption
Spearphishing via Service	Execution through Module Load	BITS Jobs	Dylib Hijacking	Compile After Delivery	Forced Authentication	Network Sniffing	Remote Desktop Protocol	Data from Removable Media	Domain Fronting	Exfiltration Over Physical Medium	Inhibit System Recovery
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Exploitation for Privilege Escalation	Compiled HTML File	Hooking	Password Policy Discovery	Remote File Copy	Email Collection	Domain Generation Algorithms	Scheduled Transfer	Network Denial of Service
Trusted Relationship	Graphical User Interface	Browser Extensions	Extra Window Memory Injection	Component Firmware	Input Capture	Peripheral Device Discovery	Remote Services	Input Capture	Fallback Channels		Resource Hijacking
Valid Accounts	InstallUtil	Change Default File Association	File System Permissions Weakness	Component Object Model Hijacking	Input Prompt	Permission Groups Discovery	Replication Through Removable Media	Man in the Browser	Multi-Stage Channels		Runtime Data Manipulation

Figure 14: Extract of ATT&CK Matrix™ - <https://attack.mitre.org/matrices/enterprise/>

The MITRE ATT&CK, as written before, is a behavioural model that is composed of several components. Tactics, represented by columns of the matrix, techniques represented by every single cell on matrix (Strom B. E., 2017).

Other elements that are present within ATT&CK model are Groups and Software.

Group - represent Threat actors, including they techniques and used software

Software - malware and utilities linked to techniques

Tactics represent the tactical goals that an adversary could be reached by using techniques reported within the matrix. The tactics are the headers of ATT&CK matrix, see figure below for list of tactics.

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion		
Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact

Figure 15: ATT&CK tactics

4.4.1. Sample of Tactic - Credential Access

From MITRE website we can explore for example one tactic and related techniques.

[Credential Access ID: TA0006](#)

Credential access represents techniques resulting in access to or control over system, domain, or service credentials that are used within an enterprise environment (The MITRE Corporation, 2019). Adversaries will likely attempt to obtain legitimate credentials from users or administrator accounts (local system administrator or domain users with administrator access) to use within the network. This allows the adversary to assume the identity of the account, with all of that account's permissions on the system and network, and makes it harder for defenders to detect the adversary. With sufficient access within a network, an adversary can create accounts for later use within the environment.

ID	Name	Description
T1098	Account Manipulation	Account manipulation may aid adversaries in maintaining access to credentials and certain permission levels within an environment. Manipulation could consist of modifying permissions, modifying credentials, adding or changing permission groups, modifying account settings, or modifying how authentication is performed. These actions could also include account activity designed to subvert security policies, such as performing iterative password updates to subvert password duration policies and preserve the life of compromised credentials. In order to create or manipulate accounts, the adversary must already have sufficient permissions on systems or the domain.
T1139	Bash History	Bash keeps track of the commands users type on the command-line with the "history" utility. Once a user logs out, the history is flushed to the user's <code>.bash_history</code> file. For each user, this file resides at the same location: <code>~/.bash_history</code> . Typically, this file keeps track of the user's last 500 commands. Users often type usernames and passwords on the command-line as parameters to programs, which then get saved to this file when they log out. Attackers can abuse this by looking through the file for potential credentials.
T1110	Brute Force	Adversaries may use brute force techniques to attempt access to accounts when passwords are unknown or when password hashes are obtained.
T1003	Credential Dumping	Credential dumping is the process of obtaining account login and password information, normally in the form of a hash or a clear text password, from the operating system and software. Credentials can then be used to perform Lateral Movement and access restricted information.
T1081	Credentials in Files	Adversaries may search local file systems and remote file shares for files containing passwords. These can be files created by users to store their own credentials, shared credential stores for a group of individuals, configuration files containing passwords for a system or service, or source code/binary files containing embedded passwords.
T1214	Credentials in Registry	The Windows Registry stores configuration information that can be used by the system or other programs. Adversaries may query the Registry looking for credentials and passwords that have been stored for use by other programs or services. Sometimes these credentials are used for automatic logons.

Figure 16: Techniques of Credential Access tactics

The adversary emulation is the process used for applying cyber threat intelligence to a technology domain by miming the behaviour about specific adversaries by emulating how they obtain the "crown of jewels".

The model proposed by MITRE using ATT&CK threat model focusing on an adversary's post-compromise detection and focused on adversary behaviour more than IOCs.

Other models or approaches generally based on network security (always a valid approach) are not useful to detect a modus operandi of the adversary, or for example, creates a several paths to obtain a specific objective, such as access to a sensitive information stored within our systems and verify subsequently how our sensors react to a specific solicitation.

Advantage that we can obtain using the ATT&CK knowledge base is verify the capacity of sensors deployed on RESISTO project to detect the anomalies or attacks and verify how RESISTO platform performing in several cases of attack.

4.5. Usage of the tools within RESISTO

The tools and methods described in the previous subsections are industry-accepted and applied by telecom partners in the consortium. They could potentially be used in RESISTO to support the risk

and resilience management process of the LTCL. They would be mainly implemented/executed by the operators of the testbeds, but the results would contribute to the risk and resilience analysis of the LTCL, in particular to step 4 “Disruptions identification”. The implementation of the tools in the risk and resilience management process is shown preliminary in Figure 17 in section 5 and will be further discussed in the final report D3.2 “Risk and resilience management process for cyber-physical threats of telecom CI” of Task 3.1. A short summary of the feedback given by the operators and technical partners for each tool is given in the following:

1. Attack trees are a way to describe an attack, in the LTCL they can be used to identify situations that the organization has no defence process in place. Each attack should have countermeasures in place.
2. Honeypots are used to stimulate attacks and to identify them. They are to be in place in the network live to collect attacks and need to be included by the operators in their testbeds. For the LTCL, honeypots could be used to collect elements, to be crossed with other sources, in order to prevent attacks. The results can be used to verify that potential attackers actually try to access the networks. Each operator, in particular if their network is connected to the internet, should deploy honey pots and perform a penetration to trigger them.

Honeypots could also provide early warning events for the STCL. No RESISTO components are foreseen in the DoW to detect events based on honeypots, but a honeypot component could be put in place by operators sending events to RESISTO correlator.

3. Penetrations tests are a typical tool for threat identification and support both the vulnerability disclosure and the risk and resilience assessment analysis. They should be used by each end-user to test their infrastructure off-line, providing the results for the LTCL analysis.
4. Telecom providers should constantly monitor the information in the “MITRE ATT&CK” knowledge base on potential attacks in order to develop and update threat models for risk assessment of their networks. “MITRE ATT&CK” will be used for some check in the TIM testbed. This will be described in more detail in D2.8.

4.6. Requirements traceability

In task 2.1 (D2.1), the operators and technical experts of the consortium collected a large number of requirements that the RESISTO platform shall, should or could comply with – the modal verb stating the level of technical readiness of the platform. 38 of them have been labelled “SHALL”, which means the implementation of those requirements is mandatory for a TRL7 prototype. “SHOULD” requirements are mandatory for TRL 8/9 and “COULD” requirements are not mandatory at any level but could improve performance and functions of the platform.

During the design and development phase of the RESISTO platform, all “SHALL” requirements have been and will be traced and monitored. If a requirement cannot be covered, the reason for that will be analysed and appropriate recovery actions will be identified.

The goal is a traceability matrix at the end of the project, where the completeness of the implementation of the requirements can be demonstrated together with the information about why, where and how they were implemented.

To ease the complexity of that task, the tracing of requirements will be carried out in each relevant deliverable to come, where the “SHALL”-requirements will be matched with the methods, tools, cases etc. described in that document.

In general, the main requirements for the LTCL are given by RES_FUN_0570 and RES_FUN_0070, but there are contributions as well to other requirements such as RES_FUN_0700 or RES_FUN_0005 (see also D3.2 of T3.1).

A particular emphasis is given to the tools and methods described in this chapter. A matching to corresponding requirements is shown in *Table 2*.

Mandatory Requirements (D2.1, chapter 6.3)		
Requirement Identity Code	Requirement Description	Related to method / tool (D3.4)
RES_FUN_0005	RESISTO shall exploit the outcomes of the cyber security and the physical security systems of the TLC infrastructures (if existing).	4.2 Honeypots
RES_FUN_0070	RESISTO shall suggest to the operator the necessary steps to mitigate the effect of a cyber/physical attack.	4.1 Attack trees
RES_FUN_0570	The <i>Risk and resilience assessment analysis</i> shall also take into consideration network single point of failure nodes, using network metrics such as:	4.3 Penetration Test Assessment; 4.4 MITRE ATT&CK
	✓ Link state protocol databases for alternative IGP routes	
	✓ BGP secondary paths for EGP routes	
	✓ HSRP/VRRP/GLBP statuses for gateway redundancy	
RES_FUN_0670	The <i>Vulnerability Disclosure Framework</i> shall be able to provide users with functionalities to define the scope for testing, rewards for different types of threats.	4.3 Penetration Test Assessment
RES_FUN_0680	The <i>Vulnerability Disclosure Framework</i> shall be able to allow Security Researchers to submit findings.	4.1 Attack trees
RES_FUN_0700	The <i>Vulnerability Disclosure Framework</i> shall be able to help Security Researchers and users to monitor vulnerabilities reported through the whole cycle:	4.3 Penetration Test Assessment; 4.1 Attack trees
	✓ report the finding,	
	✓ confirm/reject/request information from the security research	
	✓ notify the stakeholders,	
	✓ patch the finding,	
	✓ confirm from the security researcher that the issue was fixed,	
	✓ reward the security researcher, if appropriate.	

Table 2 : Mandatory requirements related to the tools and methods described in chapter 4 of this document.

5. WEB-APP FOR RISK AND RESILIENCE ASSESSMENT

As described in chapter 2, a web application was developed to access the tabular input for the risk and resilience management process. In Figure 17, a graphical representation of the management process loop is shown with its main inputs and tools currently considered. In comparison to previous versions of the process presentation, the tools discussed in chapter 4 were added, referred to as “Testbed tools”. As mentioned in section 4.5 the tools mostly contribute to the disruption identification in step 4, but partially they may also provide some feedback for potential mitigation options in step 8.

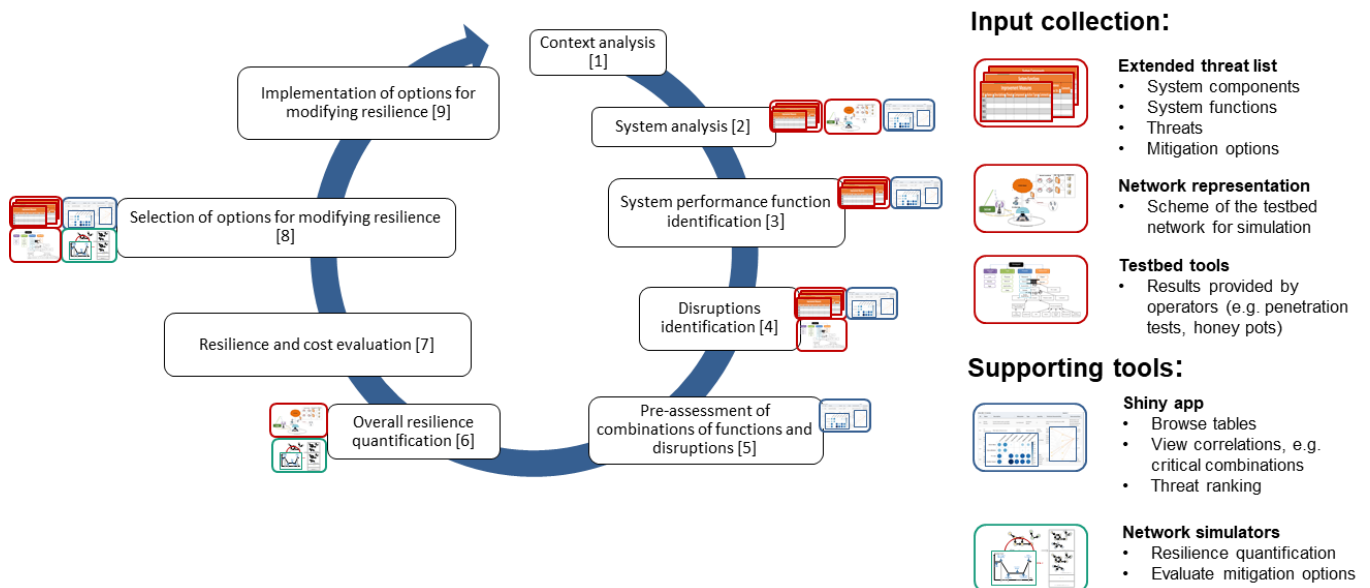


Figure 17: Risk and resilience management process based on (Häring, 2017). The usage of the tabular Excel inputs and graphical network representations is indicated by orange boxes and the support by the Shiny app and simulation tools by blue or green boxes, respectively.

The app is developed in the statistical computing language R using the Shiny package for web-applications (Chang, 2017).

The app is structured by a dashboard that allows to choose between the main features of the app. The Excel inputs are collected as separate files from different partners and the first option is to choose which file to use. It is planned to update the files with more specific information for each testbed-use case example explored in WP7-WP9.

In general, the tables are structured such that each row corresponds to one item of the corresponding category, e.g. one threat or one component. A unique identifier (ID) is assigned to each item, by using numbered abbreviations of the names of the tables as shown in Table 3.

More information about the structure and contents of the tables is given in the deliverables D2.2/D2.3 and D2.4/D2.5 of WP2.

ID	Table
SC1, SC2, ...	1. System Components
SF1, SF2, ...	2. System Functions
T1, T2, ...	3. Threats
IM1, IM2, ...	4. Improvement Measures

Table 3: Identifier (ID) assignment for the Excel tables.

In the following a short description of the main features of the app is given:

View Tables: Browse the information of the four tables (1. System Components, 2. System Functions, 3. Threats, 4. Improvement Measures). All contents of the tables are printed and can be searched for specific strings, see Figure 18.

Connections: Visualize the linkages between the tables by plotting the IDs and connections between them, see Figure 19. Each table corresponds to one layer or column of the plot. By default, all tables are included, but optionally individual tables/layers can be removed. By clicking on one item, information about the item and the directly connected items is printed below the plot.

Correlations: Plot correlation matrices for a set of two chosen tables, see Figure 20. Several columns of the tables can contribute to the correlation and the contribution strength of each contribution can be modified (default values are provided). It should be noted that the matrices are not normalized and do not correspond to the mathematical definition of correlations, but rather represent the connection strengths between items.

Threat Ranking: Perform a score calculation to allow for a ranking of the risks, see Figure 21. Several columns of the third table (3. Threats) provide input to rank the threats, e.g. frequency and economic impact of the threats. A score model is used to compute the score for each threat based on these inputs. The model definition is not unique and can be modified by the user. It is predefined to $FQ \cdot (EI + SI)$, i.e. frequency (FQ) multiplied with the sum of economic impact (EI) and impact on society (SI). Most relevant inputs are filled by characters from pre-defined drop-down menus, e.g. low, medium or high economic impact, and need to be translated to numerical values for the score calculation. Default values for this are provided, but can be changed by the user. The results are plotted as a sorted bar plot. In addition to the computed score, also the contributing entities (e.g. FQ, EI and SI) can be shown in the plot.

The app can be easily extended by further options, for example to present additional information and input retrieved from the network simulators.

The Excel input files should be updated on a regular basis to ensure that the information provided by the app is up to date. Most structural changes of the input files do not require significant changes in the code of the app, but slight modifications might be necessary e.g. to account for additional inputs used by the score ranking.

Running the app locally on a machine requires to install the free software R plus the necessary packages (shiny, shinydashboard, shinycssloaders, shinyjs, ggplot2, readxl, dplyr, stringr, DT, plotly, corrplot). To deploy it on the web, the best option is to install Shiny Server on a Linux server (RStudio, 2017).

A more precise description of the implementation of the app in the RESISTO platform shall be given Task 3.1 and in WP6.

Treat List

operator:

Select table:

- ☒ 1. System components (SC)
- ☐ 2. System functions (SF)
- ☐ 3. Threats
- ☐ 4. Improvement measures (IM)
- ☒ Remove empty columns

View Tables

Connections

Correlations

Threat Ranking

Show 10 entries

ID	Name	Description	Subsystem	Type	Quantity	Technical characteristics	Interconnections
SC1	Border Routers	Carrier Grade routers, provides resources access to subscribers	Core Network	Hardware Device	3	CISCO Carrier Grade Routers, 9000-Series	Workstations and Servers, Network Security Equipment, FO Infrastructure
SC2	FO Infrastructure	Fiber Optics Infrastructure	Optical Network	Interconnection	7548 km owned FO	Buried or aerial installation fiber optic cable. Transport technologies used are: DWDM or Gigabit Ethernet over fiber.	Border Routers, MSC, Radio Infrastructure
SC3	Mobile Switching Centers (MSC)	Primary service delivery nodes for GSM/CDMA, responsible for routing voice calls and SMS as well as other services	Core Network	Hardware Device	3 MSCS/7MGW	Ericsson MSCS: circuit-switched calling mobility management and GSM services to the mobile phones Ericsson MGW: conversion between different transmission and coding technique	FO Infrastructure, Border Routers
SC4	Radio Infrastructure (BTS, BSC, RNC, NodeB)	Provides radio connectivity for legacy (2G + 3G) and 4G services (voice and data)	Radio Network	Hardware Device	N/A		Border Routers, FO Infrastructure
SC5	Network Security Equipment (IPSs, FWs)	Deployed network security infrastructure including Firewalls, IPS, WAFs etc.	Core Network	Hardware Device		5: Mobile Services 2: Fixed Internet Services for Corporate customers Fortinet Next-Gen Firewalls with UTM capabilities	Border Routers, Workstations and Servers, Applications
SC6	Workstations and Servers	All servers, internal and public-facing, all end-points in one of the Microsoft Security Domains	Internal Network	Hardware Device	N/A	Microsoft Windows PCs, Microsoft Windows Servers and various CentOS/RHEL Servers running on bare iron or in VMs	Border Routers, Network Security Equipment, Microsoft Security Domain
SC7	Microsoft Security Domain	All devices, users, policies and data in one of the Microsoft Windows Security Domains	Internal Network	Software Tool	x	MSAD + Windows Professional workstations	Network Security Equipment, Workstations and Servers, Border Routers
SC8	Business Applications	Applications such as SSO/Multi Authentication tool, Databases, Internal Webservers (Intranet), Billing Apps, Monitoring apps, VPN access etc.)	Applications	Software Tool	N/A	Various Business Apps based on technologies such as Databases, Database Connectors, Java, APIs etc.	Workstations and Servers, Microsoft Windows Security Domain, Border Routers
SC9	Equipment Shelters	Build structures that houses and provides weather and human-tampering protection to sensitive equipment	Radio Network	Built structure	N/A	Built structures that houses various equipment	Radio Infrastructure, FO Infrastructure
SC10	Mobile Core Network		Core Network	Hardware Device	11		FO Infrastructure, Radio Infrastructure

Showing 1 to 10 of 10 entries

Previous 1 Next

Figure 18: Start screen of the Shiny app. The main panel in black on the left allows to switch between the main features of the app. The option of viewing the tables is selected by default. It allows to browse all tables of the Excel inputs and searching them.

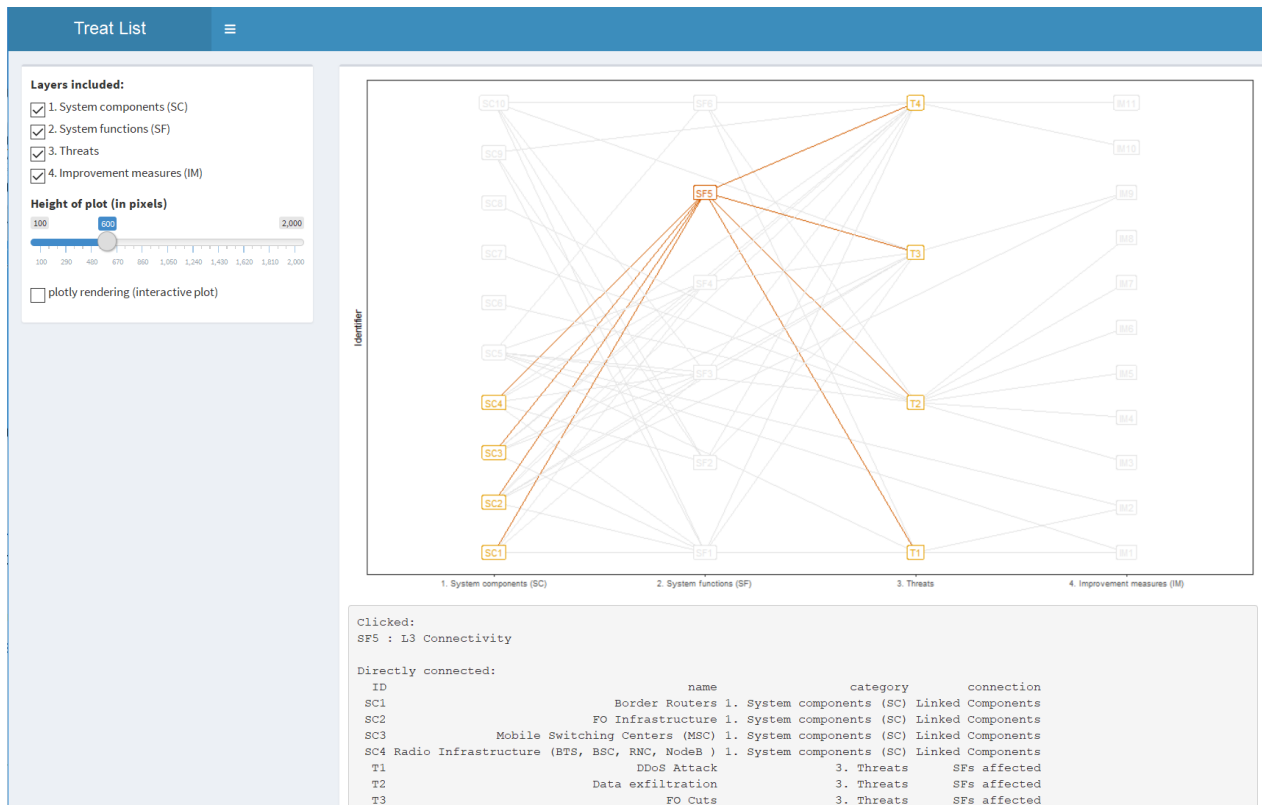


Figure 19: Connections option of the Shiny app, visualizing the connections between the items. In the plotted example SF5 was clicked and information about this system function and the connected system components and threats is printed below the plot.



Figure 20: Correlation option for the Shiny app, printing connection strength matrices for two chosen tables, e.g. critical combinations of threats and system functions. See main text for details about the strength computation

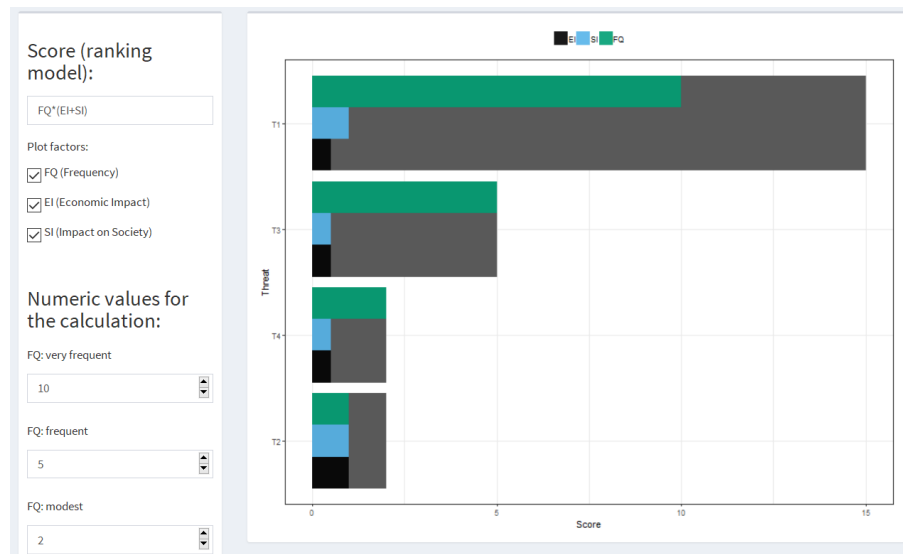


Figure 21: Threat Ranking option of the Shiny app. The user can define a score model based on the inputs in the threats table to rank the threats according to this model. Further details are given in the main text.

6. SUMMARY

Main goal of the task T3.2 “Methods/Plans for joint cyber-physical security management process”, of which the status is presented in this report, is the collection of fast and flexible methods for the risk and resilience management in telecommunication infrastructures.

A short description of the risk and resilience management process was given in chapter 2 to provide an overview of the main path to be followed by WP3.

Existing standards and handbooks were collected in chapter 3, since the methods following in chapter 4 should refer to these as far as possible.

The list of known and applied methods is presented in chapter 4. These methods are well accepted and used in industry and comply with the standards and handbooks from chapter 3.

In addition, a fast and flexible tool based on a tabular assessment that was developed for the RESISTO project is presented in chapter 5.

This report was written at the end of runtime of the task. The contents from the intermediate report D3.3 were reviewed and updated (for a list of modifications see Introduction) to ensure completeness. New contents were added mainly to chapter 4. A special emphasis was given to the question if and how the tools could be used in RESISTO (see section 4.5).

The deployment of tools from chapter 4 needs to be done by the operators of the testbeds. Further specification on the execution of the tools and methods needs to be provided in D2.8 and during the platform implementation in WP6-WP9 (i.e. which tools will be run on which testbed by the operators). In general, it is planned that the results from the tools will be shared and provide additional input for the LTCL, i.e. for the disruption identification step of the risk and resilience management process.

REFERENCES

- Barbara Kordy, S. M. (2012). Attack–Defense Trees. *the Journal of Logic and Computation* 2012.
- Chang, W. a. (2017). shiny: Web Application Framework for R. R package version 1.0. . *R Found. Stat. Comput., Vienna*. [https://CRAN.R-project.org/package= shiny](https://CRAN.R-project.org/package=shiny) (accessed 12 Feb. 2018).
- Chen, T. a.-N. (2011). Lessons from stuxnet. *Computer*, pp. 91-93.
- Disterer, G. (2013). ISO/IEC 27000, 27001 and 27002 for information security management. *Journal of Information Security*, 4, p. 92.
- European Union Agency for Cybersecurity. (2018). *ENISA Threat Landscape Report 2017*. European Union Agency for Cybersecurity.
- Florian Arnold, D. G. (2015). Sequential and Parallel Attack Tree Modelling. In F. K. Coen Van Gulijk, *SAFECOM 2015 Workshops, ASSURE, DECSoS. ISSE, ReSA4CI, and SASSUR, Delft, The Netherlands, September 22, 2015*. Springer International.
- Häring, I. e. (2017). Towards a Generic Resilience Management, Quantification and Development Process: General Definitions, Requirements, Methods, Techniques and Measures, and Case Studies Resilience and Risk. In I. Linkov, & J. M. Palma-Oliveira (Eds.), *Resilience and Risk* (Vol. 6, pp. 21-80). Dordrecht: Springer Netherlands.
- ISO. (2013a). *ISO/IEC 27001:2013*. Retrieved from ISO: <https://www.iso.org/standard/54534.html>
- ISO. (2013b). *ISO/IEC 27002:2013*. Retrieved from ISO: <https://www.iso.org/standard/54533.html>
- J. D. Guarnizo, A. T. (2017). Siphon: Towards scalable high-interaction physical honeypots. *Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security*. ACM, (pp. pp. 57–68.).
- Ludovic Piètre-Cambacédès, M. B. (2010). Beyond attack trees: dynamic security modeling with Boolean logic Driven Markov Processes (BDMP). *Conference Paper*.
- Marco Gribaudo, M. I. (2015). Exploiting Bayesian Networks for the Analysis of Combined Attack Trees. *Electronic Notes in Theoretical Computer Science* 310 (2015) , 91–111.
- Markoff, J. (2009). Vast spy system loots computers in 103 countries. *The New York Times*, 29.
- NCSC. (2017). *Penetration Testing - Advice on how to get the most from penetration testing*. Retrieved from <https://www.ncsc.gov.uk/guidance/penetration-testing>
- Ola Flaten, M. S. (2014). How Good are Attack Trees for Modelling Advanced Cyber. *Norwegian Information Security Conference 2014 (NISK-2014)*.
- RStudio. (2017, October). *Deploying Shiny apps to the web*. Retrieved from Shiny: <https://shiny.rstudio.com/articles/deployment-web.html>
- RStudio. (2019, December). *Shiny*. Retrieved from <https://shiny.rstudio.com/>
- Scarfone, K. a. (2008). Technical guide to information security testing and assessment. *NIST Special Publication*, pp. 2-25.
- Schneier, B. (1999). Attack trees: Modeling security threats. . *Dr. Dobb's Journal*, 24(12):21–, 24(12):21–.
- Strom, B. E. (2017). *Finding cyber threats with ATT&CK-based analytics*. Technical report.

Strom, B. E. (2018). *MITRE ATT&CK™: Design and philosophy*. Technical report.

The MITRE Corporation. (2019). *Credential Access*. Retrieved from MITRE ATT&CK:
<https://attack.mitre.org/tactics/TA0006/>