

# **RESISTO:**

## **D3.3\_Methods for cyber-physical security management for telecom CI**



# RESISTO

## D3.3 – METHODS FOR CYBERPHYSICAL SECURITY MANAGEMENT FOR TELECOM CI

<b>Document Manager:</b>	Mirjam Fehling-Kaschek	Fraunhofer	Editor
--------------------------	------------------------	------------	--------

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

<b>Document ID N°:</b>	RESISTO_D3.3_190516_02	<b>Version:</b>	2.0
<b>Deliverable:</b>	D3.3	<b>Date:</b>	16/05/2019
		<b>Status:</b>	APPROVED

<b>Document classification</b>	<b>Public</b>
--------------------------------	---------------

Approval Status	
<b>Prepared by:</b>	Mirjam Fehling-Kaschek (Fraunhofer)
<b>Approved by: (WP Leader)</b>	Mirjam Fehling-Kaschek (Fraunhofer)
<b>Approved by: (Coordinator)</b>	Federico FROSALI (LDO)
<b>Advisory Board Validation (Advisory Board Coordinator)</b>	NA
<b>Security Approval (Security Advisory Board Leader)</b>	Alberto BIANCHI (LDO)

## CONTRIBUTING PARTNERS

Name	Company / Organization	Role / Title
Mirjam Fehling-Kaschek, Jörg Finger	Fraunhofer	Scientific Researcher
Giuseppe Amato	TEI	
Lucian Enescu, Octavian Echim, Ioan Constantin	ORO	Information Security Experts

## 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
1.0	12.12.2018	All	All	First full draft with contributions from EMI, TEI, ORO
1.1	19.12.2018	All	All	WP review: comments from OTE, TEI and EMI
1.2	21.12.2018	All	All	Release for SAB review
2.0	16.05.2019	All	All	Final release

## 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 delle Officine Galileo 1 – Campi Bisenzio (FI) – 50013 – Italy

Tel.: +39 055 5369640, Fax: +39 055 5369640

E-Mail: frederico.frosali@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 summarizes the 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.

## CONTENTS

<b>1. INTRODUCTION [EMI]</b>	<b>10</b>
<b>2. RISK AND RESILIENCE MANAGEMENT [EMI]</b>	<b>12</b>
2.1. Input collection via Excel template	12
2.2. Inferring critical combinations	13
<b>3. REVIEW OF EXISTING STANDARDS AND HANDBOOKS [ORO]</b>	<b>14</b>
3.1. Standards	14
3.1.1. ISO/IEC 27000 family	14
3.2. Frameworks, Handbooks and Guidelines	17
3.2.1. Plan-Do-Check-Act (PDCA) cycle	17
3.2.2. NIST Framework for Improving Critical Infrastructure Cybersecurity	18
3.2.3. NIST SP 800-100, Information Security Handbook: A Guide for Managers	18
3.2.4. NIST Computer Security Incident Handling Guide	18
3.2.5. Centre for Internet Security (CIS)	18
3.2.6. Open Web Application Security Project (OWASP)	19
3.2.7. Open Source Security Testing Methodology Manual (OSSTMM)	19
3.2.8. ENISA Recommendations to IT Industry	19
3.2.9. Microsoft Security Development Lifecycle	19
<b>4. COLLECTION OF ASSESSMENT METHODS AND TOOLS [TEI, ORO]</b>	<b>20</b>
4.1. Deductive approaches [TEI]	20
4.1.1. Attack trees	20
4.1.2. Stuxnet attack tree example	21
4.1.3. GhostNet attack tree example	21
4.1.4. Dynamic attack tree	22
4.1.5. Combined attack tree	23
4.1.6. Attack-Defence tree	26
4.2. Honeypots [TEI]	26
4.2.1. Physical Honeypots	28
4.2.2. Network Honeypots	29
4.2.3. Application Honeypots	29
4.2.4. Data Honeypots	30
4.2.5. Research Honeypots	30
4.2.6. Production Honeypots	31
4.2.7. Quarantine Honeypots	31
4.2.8. Canary Honeypots	31
4.2.9. Social Engineering Honeypots	32
4.3. Penetration test assessment [ORO]	33
4.3.1. Goals	33
4.3.2. Types of Penetration Tests	34

4.3.3. Penetration Testing Stages .....	34
4.3.4. Penetration Testing Activities .....	34
4.3.5. Penetration Testing Programme .....	36
4.3.6. Penetration Testing Methodologies .....	37
<b>5. WEB-APP FOR RISK AND RESILIENCE ASSESSMENT [EMI] .....</b>	<b>38</b>
<b>6. SUMMARY [EMI] .....</b>	<b>43</b>
6.1. Next steps .....	43

## Table of figures:

Figure 1: RESISTO logical architecture (see Deliverable D2.6 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. ....	10
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. ....	12
Figure 3: Plan-do-check-act cycle .....	17
Figure 4: Physical Safe - Attack tree (Schneier, 1999) .....	20
Figure 5: Stuxnet attack tree example (Ola Flaten, 2014) .....	21
Figure 6: GhostNet - Attack tree example (Ola Flaten, 2014) .....	22
Figure 7: Dynamic tree example: Stuxnet (Florian Arnold, 2015) .....	22
Figure 8: Dynamic Attack Tree: RAS malicious access example (Ludovic Piètre-Cambacédès, 2010) .....	23
Figure 9: Simple example of combined attack tree (Marco Gribaudo, 2015). ....	24
Figure 10: Multiple Domain Combined Cyber-Physical Attack tree (Marco Gribaudo, 2015) .....	25
Figure 11: Attack-Defence trees (Barbara Kordy, 2012) .....	26
Figure 12: Honeypot characteristics .....	27
Figure 13: Logical diagram for penetration testing programme .....	37
Figure 14: Risk and resilience management process based on [1]. 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 .....	38
Figure 15: 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. ....	40
Figure 16: 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. ....	41
Figure 17: 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 .....	41
Figure 18: 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. ....	42



## ABBREVIATIONS

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

## 1. INTRODUCTION [EMI]

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 WP3 is the definition of the long term control loop of the RESISTO platform.

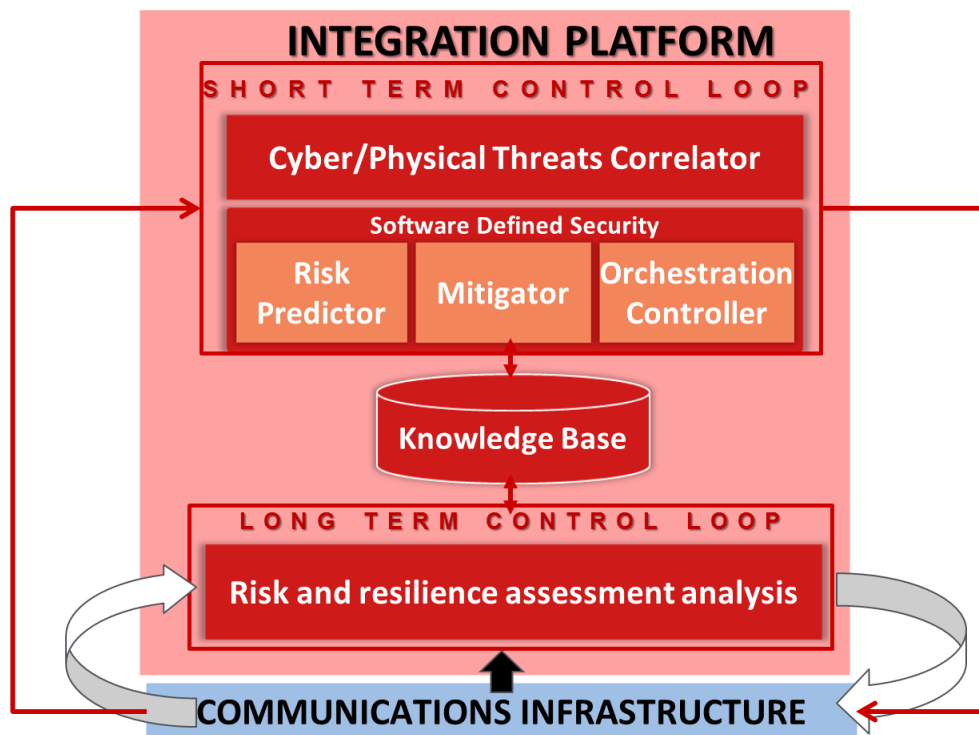


Figure 1: RESISTO logical architecture (see Deliverable D2.6 for more information). Aim of the task described in this report is to identify and evaluate methods for the fast and flexible risk and resilience assessment in the Long Term Control Loop.

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

This report summarizes the status of T3.2 at half run-time. 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.

The report is structured as follows:

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.

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

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

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.

Chapter 6 summarizes this report and provides an outlook to the next steps to be followed within T3.2.

## 2. RISK AND RESILIENCE MANAGEMENT [EMI]

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 1.

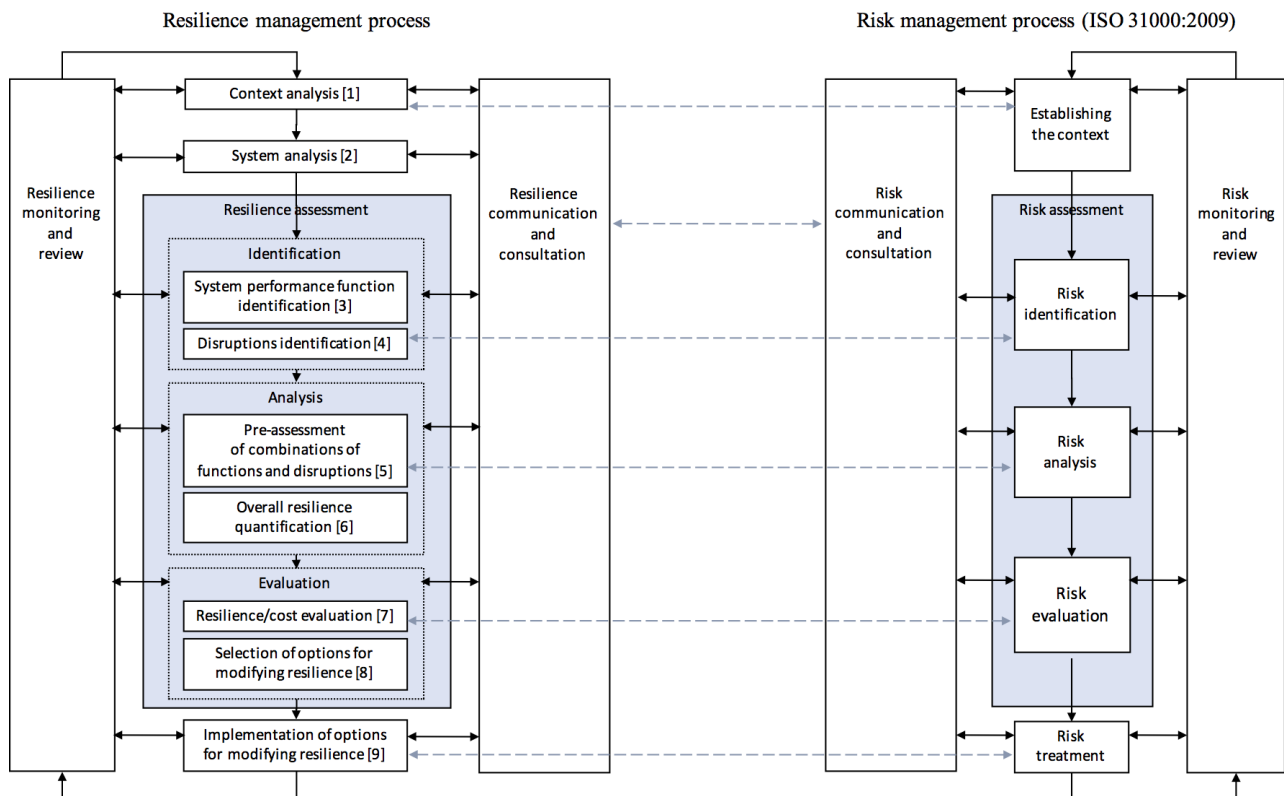


Figure 2: Risk and resilience management processes (Häring, 2017). The risk management process (right) follows the definition of ISO 31000 (2009) Risk management – Principles and guidelines.

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 will be given in the deliverable D3.1 of T3.1.

### 2.1. Input collection via Excel 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<sup>1</sup> web-app was developed, supporting a fast and flexible risk and resilience assessment. The app is described in Section 5.

## 2.2. Inferring critical combinations

Critical combinations refers to a set of system (performance) functions and threats for which critical resilience issues are prognosticated. Their determination and evaluation is a major step in the risk and resilience management process (see step 5 in Figure 1).

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 Section 5.

---

<sup>1</sup> Shiny is a package for the development of web applications in the programming language R: <https://shiny.rstudio.com/>

### 3. REVIEW OF EXISTING STANDARDS AND HANDBOOKS [ORO]

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"**<sup>2</sup>

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)

---

<sup>2</sup> <https://www.iso.org/standard/54534.html>

**ISO/IEC 27002:2013 — Information technology — Security techniques — Code of practice for information security controls (second edition)<sup>3</sup>**

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

---

<sup>3</sup> <https://www.iso.org/standard/54533.html>

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 Framework 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.

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

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

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

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

### 3.2.4. NIST Computer Security Incident Handling Guide

This publication 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:

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

### 3.2.5. Centre for Internet Security (CIS)

CIS Benchmark:

Hardening guidelines - safeguard operating systems, software and networks that are most vulnerable to cyber-attacks.

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

CIS Controls:

A series of 20 foundational and advanced cybersecurity actions, where the most common attacks can be eliminated.

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

### 3.2.6. Open Web Application Security Project (OWASP)

Top 10 Most Critical Web Application Security Risks: It represents a broad consensus about the most critical security risks to web applications.

[https://www.owasp.org/index.php/Category:OWASP\\_Top\\_Ten\\_Project](https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project)

OWASP Mobile Top 10 Risks:

[https://www.owasp.org/index.php/OWASP\\_Mobile\\_Security\\_Project#tab=Home](https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=Home)

OWASP Top 10 Proactive Controls: The OWASP Top Ten Proactive Controls 2018 is a list of security techniques that should be included in every software development project.

[https://www.owasp.org/index.php/OWASP\\_Proactive\\_Controls](https://www.owasp.org/index.php/OWASP_Proactive_Controls)

OWASP Penetration testing methodologies:

[https://www.owasp.org/index.php/Penetration\\_testing\\_methodologies](https://www.owasp.org/index.php/Penetration_testing_methodologies)

### 3.2.7. Open Source Security Testing Methodology Manual (OSSTMM)

OSSTMM 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.

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

### 3.2.8. ENISA Recommendations to IT Industry

In this short paper, ENISA 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.

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

### 3.2.9. Microsoft Security Development Lifecycle

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

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

## 4. COLLECTION OF ASSESSMENT METHODS AND TOOLS [TEI, ORO]

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

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

Below are some examples.

### 4.1. Deductive approaches [TEI]

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:

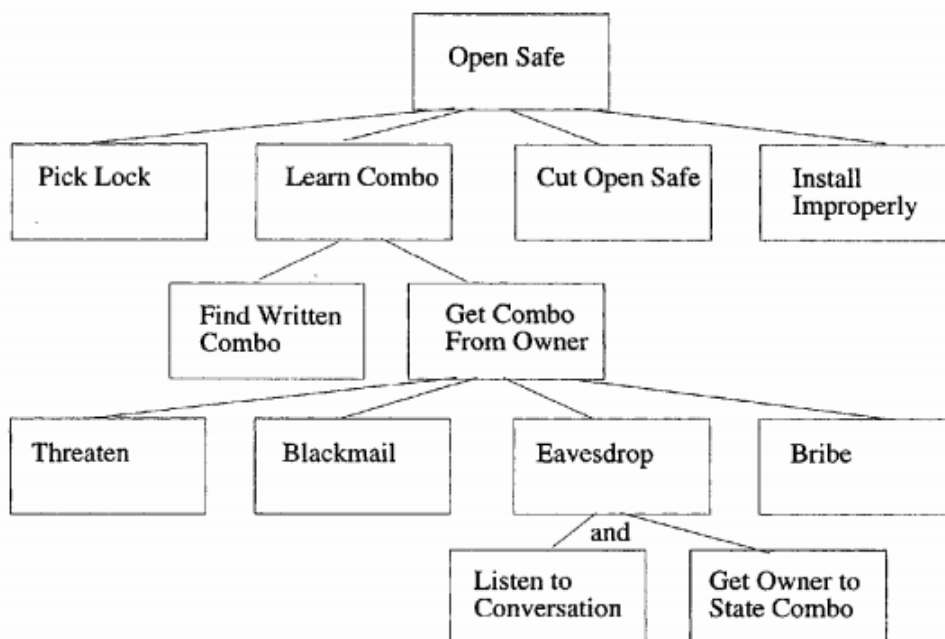


Figure 4: Physical Safe - Attack tree (Schneier, 1999)

#### 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. 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):

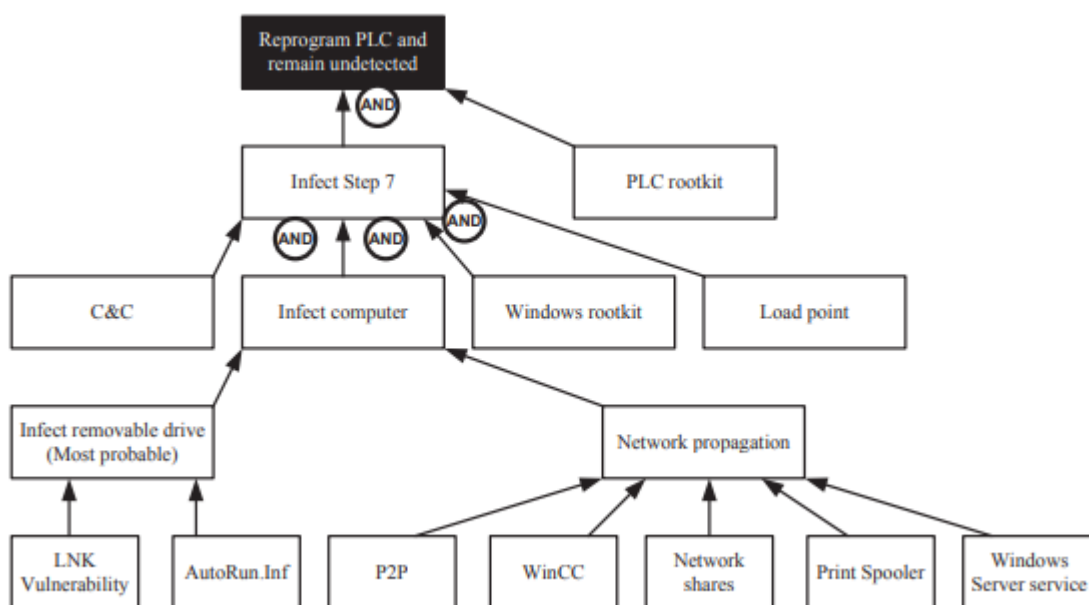


Figure 5: Stuxnet attack tree example (Ola Flaten, 2014)

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. 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):

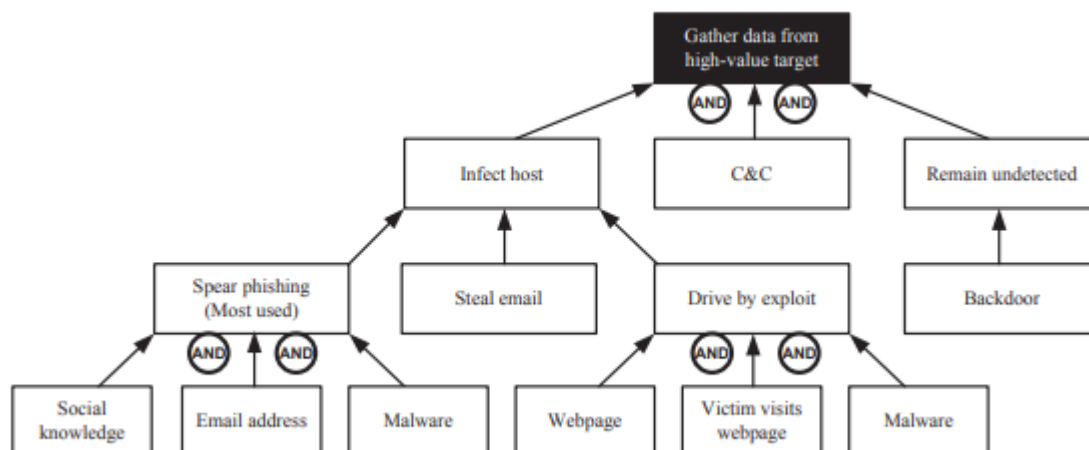


Figure 6: GhostNet - Attack tree example (Ola Flaten, 2014)

#### 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:

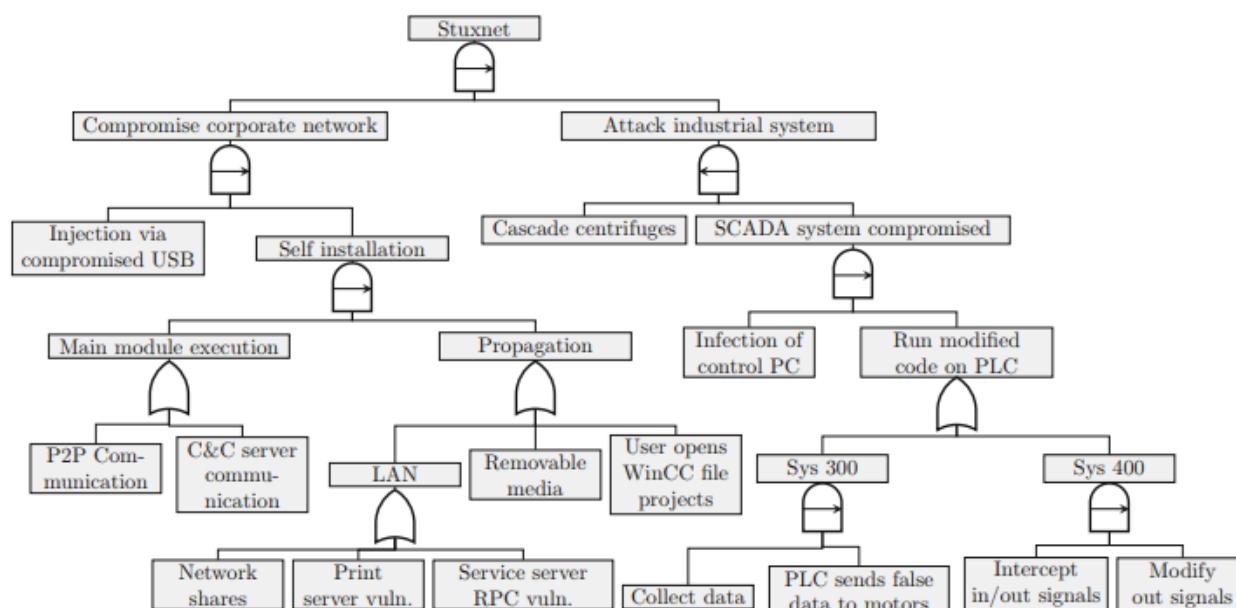


Figure 7: Dynamic tree example: Stuxnet (Florian Arnold, 2015)

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.

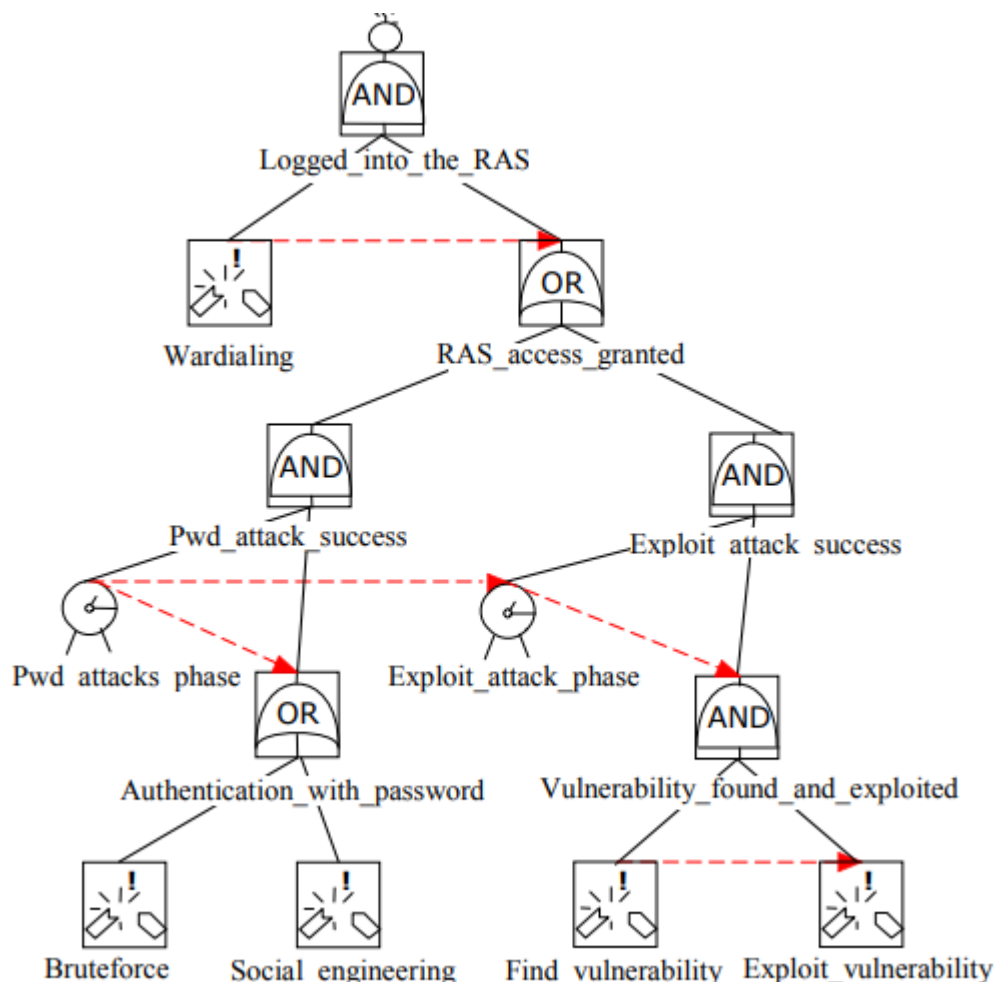


Figure 8: Dynamic Attack Tree: RAS malicious access example (Ludovic Piètre-Cambacédès, 2010)

#### 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.

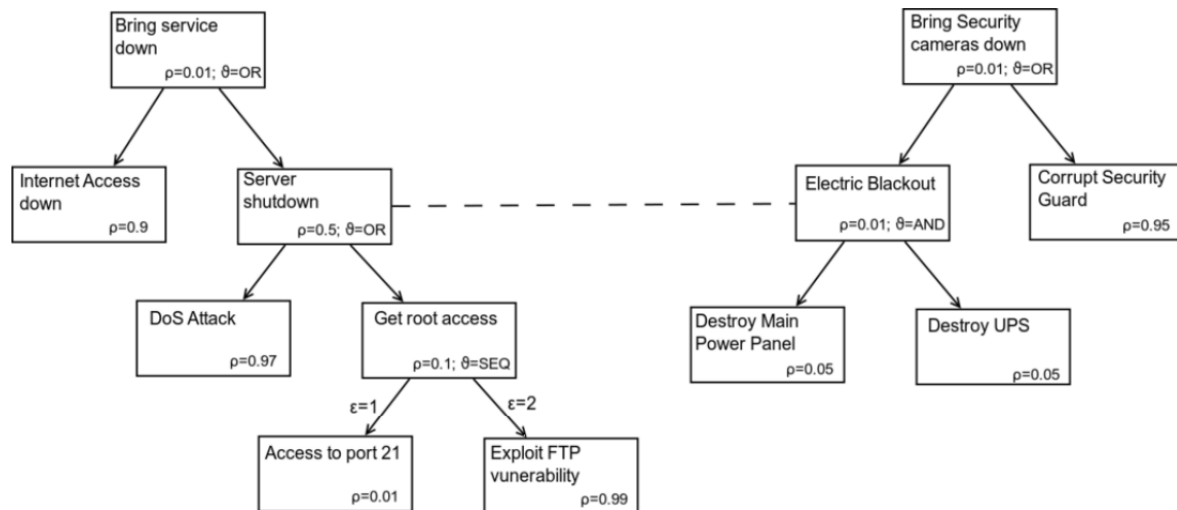


Figure 9: Simple example of combined attack tree (Marco Gribaudo, 2015).

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.



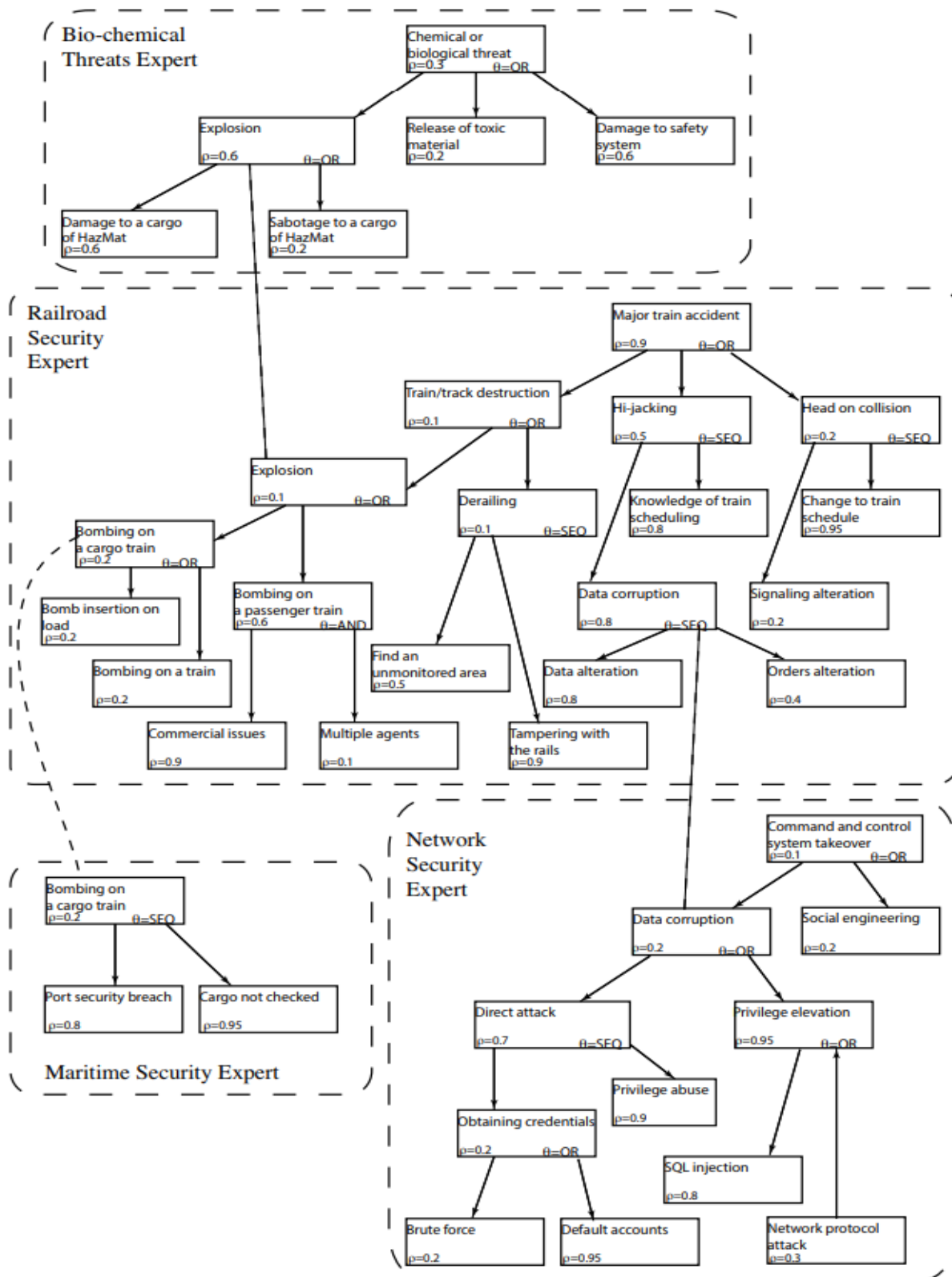


Figure 10: Multiple Domain Combined Cyber-Physical Attack tree (Marco Gribaudo, 2015)

#### 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).

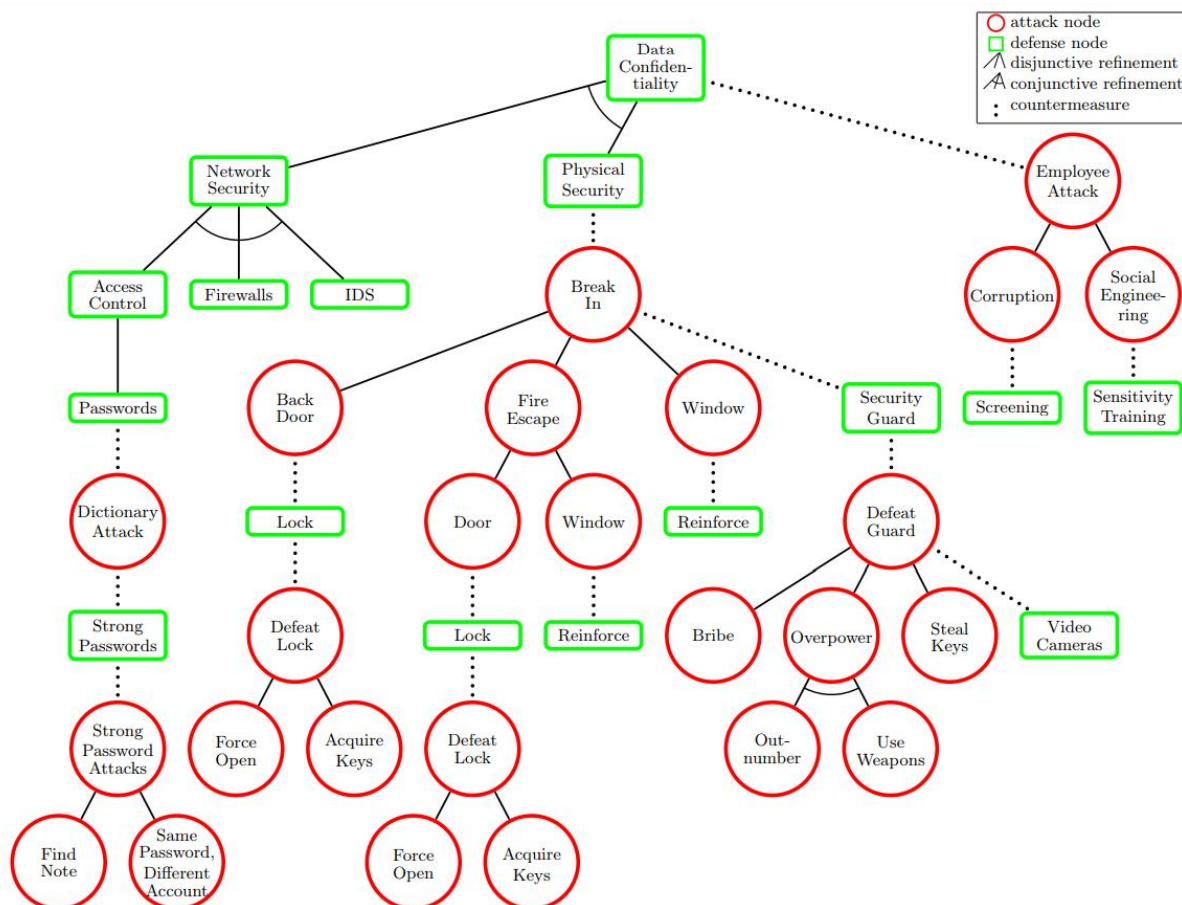


Figure 11: Attack-Defence trees (Barbara Kordy, 2012)

## 4.2. Honeypots [TEI]

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:

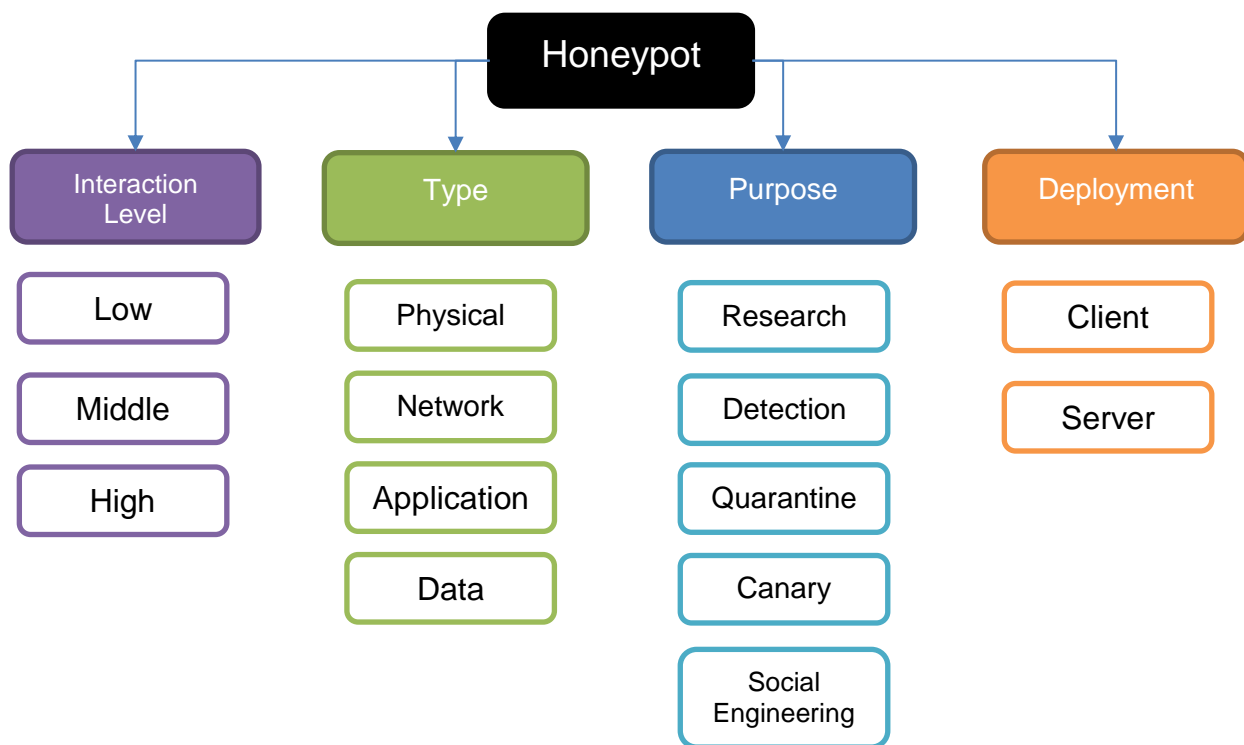


Figure 12: Honeypot characteristics

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](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 Honey pots

Data Honey pots 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>
- <https://attack.mitre.org/>

#### 4.2.5. Research Honey pots

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](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 antis spam systems. Some examples and architectural description are given at the following links:

- [https://www.researchgate.net/publication/229042409\\_Quarantine\\_Net\\_design\\_and\\_application](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.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 [ORO]

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."

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**

#### **Social Engineering**

#### 4.3.3. Penetration Testing Stages

Penetration tests are usually performed in stages with the principal 5 stages as follows:

**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.

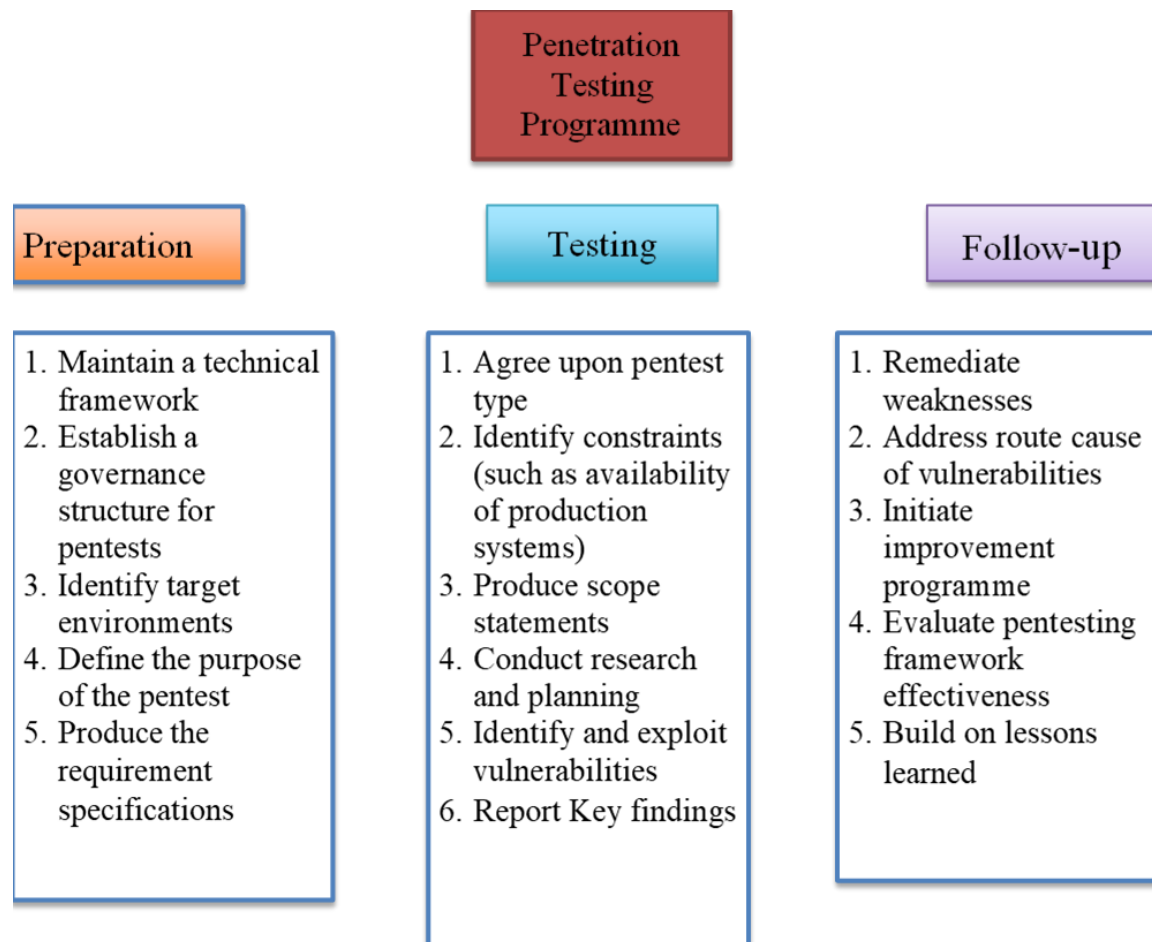


Figure 13: Logical diagram for penetration testing programme.

#### 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](https://www.owasp.org/index.php/Penetration_testing_methodologies)

PCI DSS Testing Guide:

[https://www.pcisecuritystandards.org/documents/Penetration\\_Testing\\_Guidance\\_March\\_2015.pdf](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/>

## 5. WEB-APP FOR RISK AND RESILIENCE ASSESSMENT [EMI]

As described in Section 2, a web application was developed to access the tabular input for the risk and resilience management process. In Figure 14Figure 1, a graphical representation of the management process loop is shown with its main inputs and tools currently considered.

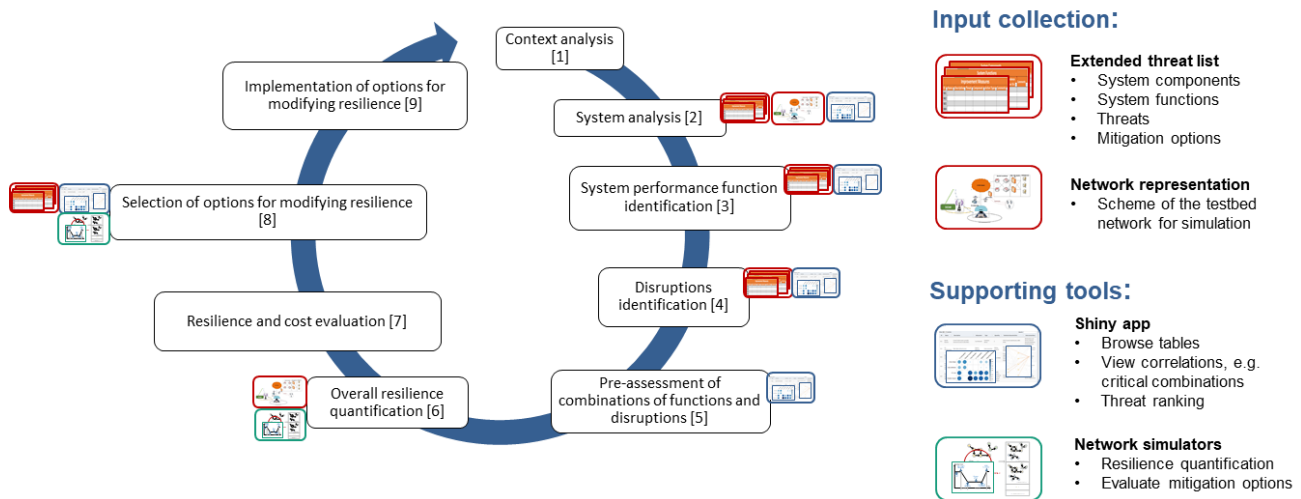


Figure 14: Risk and resilience management process based on [1]. The usage of the tabular Excel inputs and graphical network representations is indicated by orange boxes and the support by the Shiny app and simulation tools by blue or green boxes, respectively.

The app is developed in the statistical computing language R using the Shiny package for web-applications<sup>4</sup>.

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. The combination of the files into one main input file is planned as soon as at least four completed Excel files are returned by the partners.

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

<sup>4</sup> <https://shiny.rstudio.com/>

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

Table 1: 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 15.

**Connections:** Visualize the linkages between the tables by plotting the IDs and connections between them, see Figure 16. 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 17. 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 18. 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.<sup>5</sup>

The screenshot shows the start screen of the Shiny app. On the left is a sidebar with a dark blue header 'Treat List' and a menu with options: 'operator:' (with a dropdown), 'View Tables' (selected), 'Connections', 'Correlations', and 'Threat Ranking'. The main panel has a light blue header with 'Show 10 entries' and a search bar. Below the header is a table with 10 columns: ID, Name, Description, Subsystem, Type, Quantity, Technical characteristics, and Interconnections. The table lists 10 entries (SC1 to SC10) representing various network components. At the bottom of the table, it says 'Showing 1 to 10 of 10 entries' and has 'Previous', '1', and 'Next' navigation buttons.

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, 9800-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, WAFs)	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 Webserver (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

Figure 15: 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.

<sup>5</sup> Several options for deploying shiny apps are listed here at <https://shiny.rstudio.com/articles/deployment-web.html>



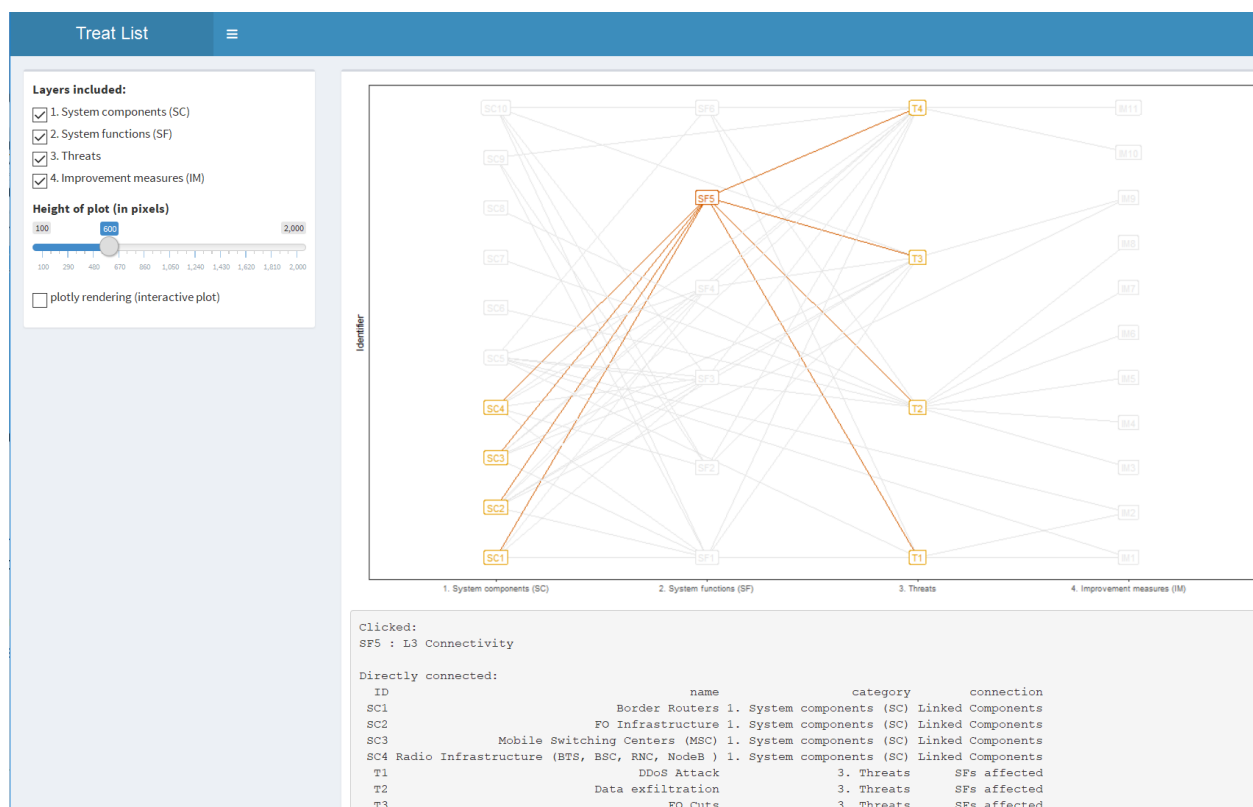


Figure 16: Connections option of the Shiny app, visualizing the connections between the items. In the plotted example SF5 was clicked and information about this system function and the connected system components and threats is printed below the plot.



Figure 17: 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

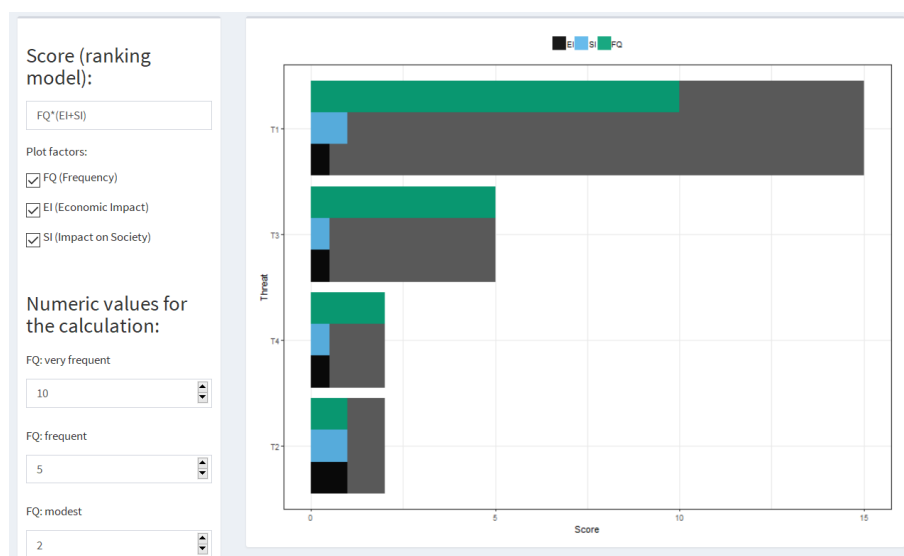


Figure 18: Threat Ranking option of the Shiny app. The user can define a score model based on the inputs in the threats table to rank the threats according to this model. Further details are given in the main text.

## 6. SUMMARY [EMI]

Main goal of the task, 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 Section 2 to provide an overview of the main path to be followed by WP3. In the future, the identified methods of this task should be associated with the corresponding resilience management steps.

Existing standards and handbooks were collected in Section 3, since the methods should reference as far as possible to these.

The list of known and applied methods is presented in Section 4. In addition, a fast and flexible tool based on a tabular assessment that was developed for the RESISTO project is presented in Section 5.

This deliverable was written at mid-runtime of the task and further contributions are expected to be added to the final version of the report at the end of the runtime of T3.2. At this point, the focus was set on the collection of material (standards and methods), meaning that the evaluation and decision making about which tools should be integrated into the RESISTO platform is still outstanding.

### 6.1. Next steps

In general, both the collection of standards and handbooks and the collection of methods needs to be further checked for completeness by all technical partners.

As mentioned above, it should be determined at which steps of the risk and resilience management the methods listed in Section 4 contribute. Also, the reference to the standards and handbooks by the collected methods needs to be addressed. The Shiny app implementation needs to be finalized and also here compliance with existing standards should be checked.

## References

- Barbara Kordy, S. M. (2012). Attack–Defense Trees. *the Journal of Logic and Computation* 2012.
- Disterer, G. (2013). ISO/IEC 27000, 27001 and 27002 for information security management. *Journal of Information Security*, 4, p. 92.
- 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.
- 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.
- Ola Flaten, M. S. (2014). How Good are Attack Trees for Modelling Advanced Cyber. *Norwegian Information Security Conference 2014 (NISK-2014)*.
- Schneier, B. (1999). Attack trees: Modeling security threats. . *Dr. Dobb's Journal*, 24(12):21–, 24(12):21–.