Call identifier: H2020-ICT-2016 - **Grant agreement no**: **732907**

Topic: ICT-18-2016 - Big data PPP: privacy-preserving big data technologies

Deliverable 5.4

MHMD Security Infrastructure

Due date of delivery: October 31th, 2018

Start of the project: 1st November 2016
Ending Date: 31st October 2019

Partner responsible for this deliverable: CNR

Version: 7.0

Document Classification: Public

Title	MHMD Security Infrastructure
Deliverable	D5.4
Reporting Period	M24
Authors	Enrico Cambiaso (CNR), Ivan Vaccari (CNR), Maurizio Aiello (CNR), E. Punta (CNR), Alexandre Flament (GNUBILA)
Work Package	WP5
Security	Public
Nature	Report
Keyword(s)	Cyber-security; project architecture; state-of-the-art security; standardization; post-project exploitation.

Document History

Name	Remark	Version	Date
Enrico Cambiaso (CNR), Ivan Vaccari (CNR)	First Version	1.0	August 1st, 2018
Maurizio Aiello (CNR)	Second Version	2.0	September 4th, 2018
Enrico Cambiaso (CNR)	Third Version	3.0	September 28th, 2018
Elisabetta Punta (CNR)	Fourth Version	4.0	October 5th, 2018
Enrico Cambiaso (CNR)	Fifth Version	5.0	October 26th, 2018
	Submitted for review	5.0	October 26th, 2018
Minos Garofalakis	Review on the final version	6.0	November 13th, 2018
Enrico Cambiaso (CNR), Ivan Vaccari (CNR)	Final version produced	7.0	November 15th, 2018

List of Contributors

Name	Affiliation
Enrico Cambiaso	CNR
Ivan Vaccari	CNR
Maurizio Aiello	CNR
Elisabetta Punta	CNR
Alexandre Flament	GNUBILA

List of reviewers

Name	Affiliation
Omiros Metaxas	ATHENA
Minos Garofalakis	ATHENA
Mirko De Maldè	LYNKEUS

Abstract

This document is aimed at providing an input for the development of the security of the MHMD platform, by providing relevant security aspects to consider and guidelines and approaches to adopt in order to guarantee protection of the system. The document may also represent an input for the hacking challenge activities executed in MHMD WP9, relatively to the (allowed) intrusion of the system by external entities, with the aim at identifying security and privacy breaches.

The document goal is to define the MHMD Security Infrastructure, including a set of security guidelines and aspects to consider during the development of the project. Such result is achieved by considering three different aspects: the current version of the MHMD architecture, its components and how they interact; a deep study of state-of-the-art attacks and protection systems and approaches for the different contexts characterizing the MHMD architecture; the potential consideration of security and privacy standards and regulations, for consideration in order to provide additional security of the system and its data.

Also, the design of the ad-hoc MHMD distributed Intrusion Detection System (MHMD-dIDS) aimed to protect the MHMD system from cyber-threats is proposed, by deeply analysing the current architecture and the components of the system, in order to identify anomalies that may be related to malicious activities (like cyber-attacks or data leaks). Thanks to the proposed Intrusion Detection System framework, an innovative protection scheme will be able to improve security of the MHMD platform.

By adopting this approach, the proposed security infrastructure provides a general security solution to adopt in order to cover the different aspects of the platform.

1 Introduction

This document is mainly focused on security issues of the internal MHMD modules, such as storage systems, web applications and APIs. Concerning secure and privacy preserving data processing, as well as privacy preserving static data publishing (through anonymization), such topics are also addressed in the D5.1 (techniques, guarantees) and D5.7 (APIs, usage flow and deployment) deliverables of the MHMD project.

This document represents an input for both the development of the security of the MHMD platform, by providing security aspects to be considered and guidelines to implement a secure system, and the public hacking challenge scope of MHMD WP9, by providing useful information on the platform and related cyber-security aspects. The aim of the document is to describe the MHMD Security Infrastructure (MSI) that has been designed in order to protect the entire MHMD system from cyber-attacks. The MSI is the result of the deep study and the consequent integration of different aspects, integrated with the design of an ad-hoc distributed intrusion detection framework. In particular, the MHMD architecture, its components and interactions were studied in detail, in order to contextualize the MSI to the current version of the system. Also, state-of-the-art threats and protection approaches were considered (in conjunction with well-known defence systems), in order to identify the most important threats for the different contexts we have identified, with relative protections. Finally, security standards should be considered from the developers of the MHMD system to provide additional protection to the whole framework, in order to consider real security applications and best practices in the cyber-security context.

Although other security frameworks are available in literature [Parker, 2012; Jouini, 2016; Chang, 2016], the proposed approach provides us the ability to define an ad-hoc security framework for the MHMD platform. A graphical view of the adopted approach is reported in Figure 1.

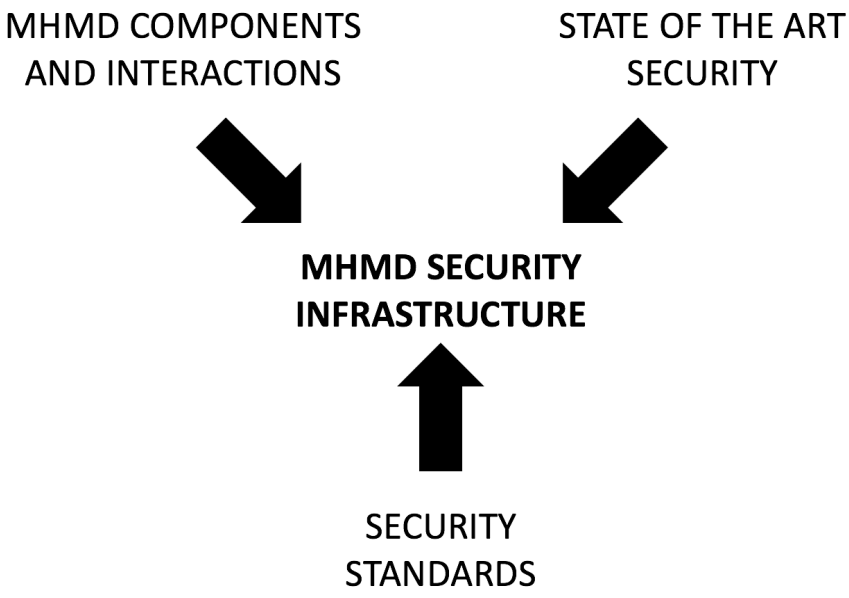

Figure 1: The approach adopted to define the MHMD Security Infrastructure

By adopting such approach, we can provide a security infrastructure focused on the MHMD context, by considering state-of-the-art security aspects, and evaluating the possibility to comply to existent and idenfied standards and regulations.

Considering an ICT based system, different contexts may be investigated in order to define the MSI:

- Physical: it concerns components protection from the physical point of view, for instance, to protect servers and rooms hosting the components [Baker, 2016], or to restrict access to personnel.
- Network: it is related to protection of the interactions between the components, by avoiding abuses from malicious network users, both internal and external of the organization(s).
- Database: given the potential sensibility of stored data, it is aimed at avoiding data exploitation and leakages, in order to guarantee integrity and availability of the system and users' data confidentiality.
- Blockchain: it concerns protection of the blockchain component of the platform from exploitation for malicious purposes, for instance by the injection of malicious software/smart contracts/transactions.
- Services: it regards back-end services protection, a crucial aspect since they represent the contact point between front-end components, directly accessible to the user, and storage information or other components.
- Web: it is relative to web services and applications protection, representing a delicate aspect to consider, since web hosts both provide direct communication with the user and access to data.
- Mobile: it is relative to the mobile security topic, covering different aspects like mobile access from remote or from the internal of the organization (BYOD) or (public or private) mobile applications interacting with the system[1].
- Desktop: it refers to the desktop security topic, covering different aspects like workstations placed inside an organization, or desktop applications interacting with the system[2].

In addition, it is also important to consider Software and Hardware contexts, that can be transversely applied to the other contexts. In particular, the software context includes aspects like secure software development or cryptography and encryption considerations, while the hardware context includes hardware security aspects and one-time-password devices.

The reported contexts can be grouped as reported in Figure 2, showing a categorization based on four main categories: *infrastructure* security, relatively to security aspects for the network infrastructure of the system; *data* security, relatively to security and privacy aspects of stored information; *back-end* security, concerning security of back-end services and applications; *front-end* security, relative to security aspects of applications directly accessed by the user.

[1] In the following, given the context of the MHMD platform, we will focus on security of the mobile application.
[2] In the following, since the MHMD platform in the current version is not composed of such components, we will not consider the desktop context in deep.

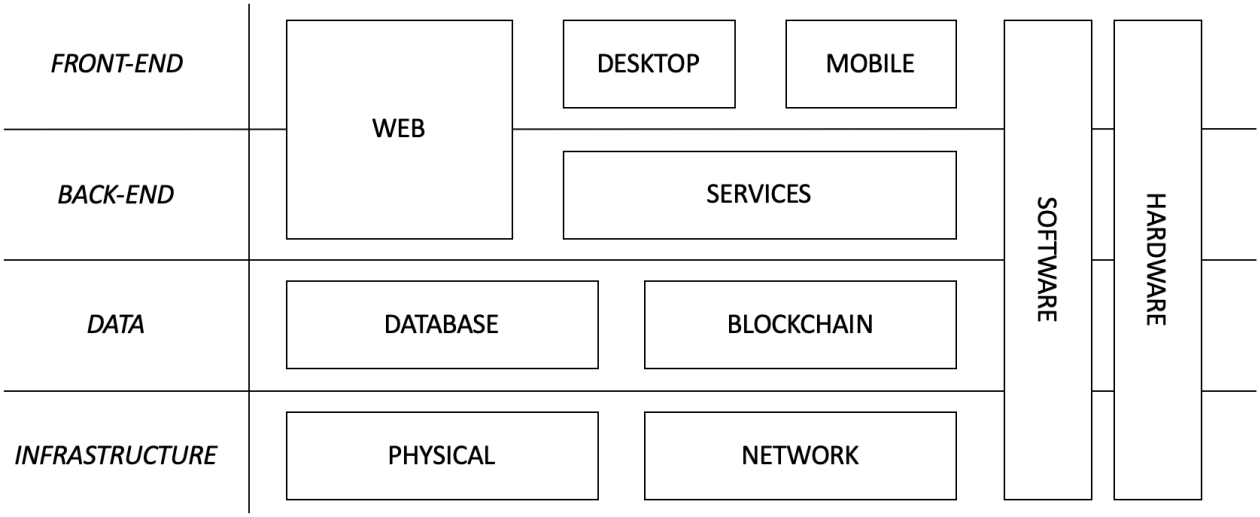

Figure 2: The contexts considered for the protection of an ICT system like the MHMD one

In general, although it depends on the kind of attack that is perpetrated, an attack involves the subsequent execution of specific steps [Hoque, 2014]. Such steps begin with a preparation to the attack and usually ends with a clean-up of the proofs of the attack. Hence, it is important to consider that before the attack is executed, the malicious user runs an information gathering phase, that is not supposed to cause the effects of the attack to the system, but only to retrieve valuable information for the attacker. Such information may be useful to detect an attack earlier/in advance, in function of the executed threat, without making it effective.

2 System architecture and components

Generally speaking, the aim of the MHMD project is to provide an interconnection system between researchers and data, by allowing researchers to get information from data providers who agreed to share their data, through a blockchain based infrastructure as source of trust and by respecting the General Data Protection Regulation (GDPR). Such retrieval/sharing activities are supposed to be accomplished automatically, when possible.

On the MHMD platform, the typical scenario of a researcher is the following one: first, the researcher browses the data by accessing a Central Catalog; then, once the data choice is made, the researcher orders the data, in order to get it/them; finally, the researcher trusts the MHMD platform to share data with given consent. Analogously, the typical scenario for a hospital is the following one: if the hospital wants to share its data, a MHMD Driver is installed and one or more data sources are configured; hence, the MHMD Driver will index the data locally, in background. An individual inside a hospital is supposed to record data into the MHMD system, by making it available to a provided consent.

The current architecture of the MHMD system is reported in Figure 3.

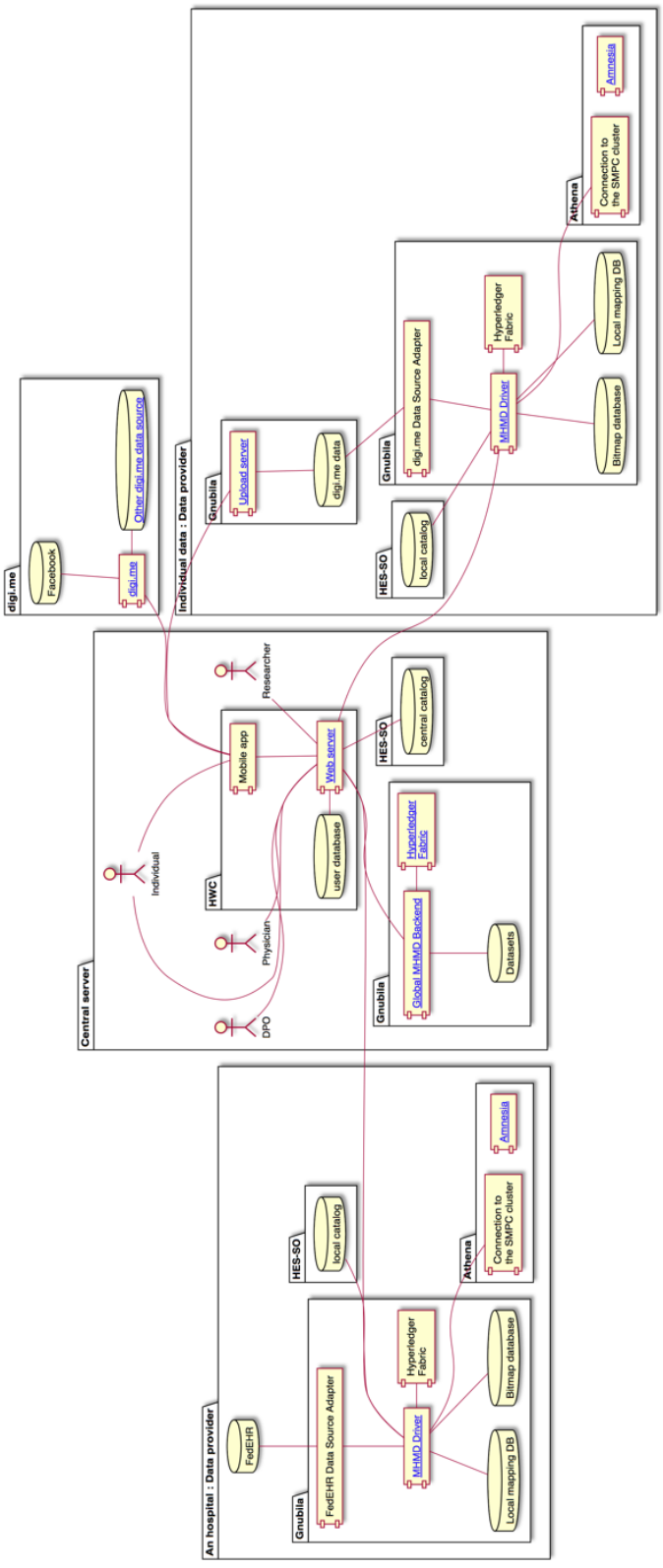

Figure 3: The architecture of the MHMD system (updated to Sept 4, 2018)[3]

As it is possible to notice, the system is made up of several components, that communicate/interact using different protocols. If we consider for instance external attacks, the most relevant components are the following ones:

- Web Server: Web Server is the main contact point for users, since all users interaction pass through it; the Web Server is supposed to manage user authentication and it relies on the Central Catalog and the Global MHMD backend to create studies, the Global MHMD backend to provide links to the datasets, to follow study status, the MHMD Drivers to record data (users are linked to one organization), and the Upload Server to provide a way to upload individual data from the Mobile App.
- Mobile App: provides users with the ability to interact with the system from their mobile devices. When the user wishes to share some personal data on the MHMD platform, the mobile application will pick up the data using the digi.me application and will use the web server as an entry point for the MHMD platform, first by authenticating to it, then requesting a session token; at this point, the data will be uploaded through the received session token to the upload server; once the data is received, the token will be deleted from the upload server.
- Upload Server: the upload server has the aim at managing the personal data, received from the mobile application, in order to store it[4], apply the *right to be forgotten* approach[5], and provide a `DataSourceAdapter` to link the data to an MHMD driver[6].
- Global MHMD Backend: this component represents the central component for researchers[7]. The aim of this component is to create a study contract (a Javascript code executed by the MHMD Driver), to create a study (also linking it to a study contract), to monitor the study contract (that may be in *defined/processing/ready/downloaded* status), to download the dataset related to a study, and to apply the *right to be forgotten* principle[8]. Studies and relative definitions are stored on the blockchain, by adopting appropriate RSA public/private keys.
- Central Catalog: this Catalog does not include personal data and includes instead metadata; it can potentially be publicly available; Central Catalog data is linked to personal data through the Local Catalog.
- Local Catalog: this catalog represents the back-end catalog for each data provider. In order to build the Local Catalog, a background process on the MHMD Driver is supposed to browse all data from all data sources, to store new retrieved information on the Local Catalog.
- MHMD Driver: the purpose of this component is not to store the data, but, instead, to give access to them according to provided consent; this component is repeated on each hospital integrated into the system and gives a hospital the possibility to connect to the MHMD platform. A background process on the MHMD Driver monitors the MHMD blockchain, looking for new studies. By connecting to the Local Catalog, requests formulated inside the retrieved studies are interpreted, to fetch data

[3] Please note that, although not visible in the current image, a communication line between the Web Server and the Upload Server should be considered.

[4] This feature is still not implemented to date, in the current version of the platform.

[5] This feature is still not implemented to date, in the current version of the platform.

[6] This feature is still not implemented to date, in the current version of the platform.

[7] In the current version of the architecture, there is only one Global MHMD Backend, specifically located on the architecture.

[8] Currently, this process is not entirely defined.

items references. Since the MHMD Driver has access to the existing consent for each data item, by mapping such information with the study definition, it is possible to filter out the data items that does not match the consent, hence preventing leakage of data without proper consent.

3 State-of-the-art security

As previously mentioned, in order to define a security infrastructure for the entire platform, it is crucial to analyse state-of-the-art security on the involved contexts, also assessing that the scenario of the cyber-attacks is going through a continuous evolution. Just to give an example, in recent years, the Advanced Persistent Threat (APT) phenomenon has appeared, concerning attacks originally targeting military domains, which instead nowadays represents an important threat to any industry and organization [Chen, 2014], because their effects are distributed on long periods of time [Brewer, 2014].

In the issue of cyber-security, it is also important to consider the logic behind an attack. In general, the cause of a cyber-attack may derive from various causes, including cyber-criminal activities, aimed at collecting money from the execution of cyber-attacks [Moore, 2009], cyber-terrorism, carrying out digital terrorism activities [Stohl, 2006], the phenomenon of hatavism, aimed at the “cyber-rebellion” [Jordan, 2007], insider threats, aimed at carrying out a sort of revenge against the organization [Colwill, 2009], or cyber-warfare activities, involving state-to-state digital warfare [Nicholson, 2012].

Considering different aspects relevant to the MHMD context, in this section we will illustrate the state-of-the-art on different security topics.

3.1 Encryption

Encryption history begins thousands of years ago, with the first reported documents dated around 1900 BC and found in Egypt. Although later well-known approaches such as the Caesar cipher were introduced [Oktaviana, 2016], modern encryption techniques begin after 1945, with the computing era [Pandya, 2015]. Encryption is nowadays a fundamental topic in the cyber-security context, since it provides data and communications protection, guaranteeing secure transfers.

In the encryption field, different techniques are investigated by [Justin, 2012], focusing on image encryption, information encryption, double encryption and Chaos-based encryption methods, and presenting and studying the effects of different encryption algorithms for each scenario.

Concerning encryption algorithms, two different approaches are possible: symmetric key, or asymmetric/public key. In the first case, the same key is used either to encrypt and decrypt data. Conversely, public key cryptography makes use of two separate keys to encrypt and decrypt data [Chandra, 2014]. Such keys are commonly known as private and public keys, and they are strictly connected. By comparing symmetric and asymmetric algorithms, [Agrawal, 2012] states that symmetric based solutions are in general faster than asymmetric ones, since the memory requirements is reduced. Considering the two encryption approaches, [Tripathi, 2014] proposes a comparison between symmetric and asymmetric approaches, by investigating different encryption algorithms. As a result, the Blowfish algorithm is the most secure algorithm for symmetric encryption purposes, while RSA is found to be the better solution for asymmetric scenarios. Similarly, [Mitali, 2014] proposes a study of popular encryption algorithms, by comparing them in function

of their advantages and disadvantages, and well as their execution times. Results show that also in this case the Blowfish algorithm provides better performance compared to the alternatives, while the least performant algorithm is 3DES. In another work, [Singh, 2013] presents a study of popular encryption algorithms like RSA, DES, 3DES and AES, also reporting a survey on encryption techniques and approaches. By comparing the algorithms, unlike the previous results, AES results the most performant one.

In general, although a single and generic rule on the algorithm to be adopted is not easy to define, since it depends on the real scenario, [Caviglione, 2017] investigates different encryption algorithms, by analysing the energy consumption of these ones (green computing). By considering several protocols and different scenarios, based on the data volume to be encrypted/decrypted, or the key length, a general set of guidelines for the adoption of security algorithms is provided, for each considered context, by also discouraging the adoption of insecure algorithms such as Data Encryption Standard (DES) one [Singh, 2013].

By investigating the literature in the encryption field, a recent trend concerns the study of homomorphic encryption schemes [Smart, 2010; Ogburn, 2013], providing operations on the encrypted data as if they were decrypted. In this context, a fully homomorphic encryption scheme is hard to define, due to the computational needs required to run such algorithm [Gentry, 2012]. Considering homomorphic encryption schemes, [Fontaine, 2007] presented a state-of-the-art of different solutions, by discussing parameters, performances and security issues of different approaches, by also discussing the advantages of homomorphic schemes, in comparison to “standard” symmetric and asymmetric approaches. In general, although several attempts to propose fully homomorphic encryption schemes are found in literature [Canetti, 2017; Brakerski, 2014; Dijk, 2010; Smart, 2010], their adoption should be evaluated in deep, in order to guarantee proper functioning of the entire system, even under stress conditions.

Another important aspect to consider concerns post-quantum cryptography, referring to the definition and investigation of cryptographic algorithms resistant to attacks perpetrated through quantum algorithms [Bernstein, 2009]. Although quantum computing is still not considered an actual threat [Valiron, 2015], in this context, several well-known and currently widely adopted public key algorithms are considered potentially not secure [Mavroeidis, 2018], since they are vulnerable to the so-called Shor’s algorithm [Ekert, 1996]. Therefore, in view of a long-term adoption of the system, the adoption or migration (e.g. through a soft fork of the blockchain component) to quantum resistant cryptographic algorithms should be considered [Perlner, 2009].

3.2 Network and communications security

A computer network may be usually composed of different components, such as servers, workstations, virtual hosts, accessory, IoT and mobile devices, but also network firewalls, switches and accelerators, wired, wireless, VPN, DMZ and VLAN networks, honey nets, etc. By analysing the MHMD architecture shown in Figure 3, it is possible to notice that different networks are present, interacting among themselves in order to make the overall system working correctly. Independently from the nodes that form the MHMD platform, in this section of the document we analyse network security. In this context, although the mobile app component may be related to a wireless connectivity, that may be potentially exposed to well-known cyber-threats [Nakhila, 2015; Singh, 2014; Welch, 2003] and to bring-your-own-device (BYOD) considerations [French, 2014], our focus is on the attacks that may be executed on an ICT network. In this context, there are essentially two kind of attacks: from one side, well-known/exploit-based attacks, often characterized by a Common Vulnerabilities and Exposures number (CVE) [Tidwell, 2001] and related to an available exploit code.

From the other side, novel/0-day attacks are unknown to the public and their exploitation is usually limited to the malicious user, or a restricted number of users [Bilge, 2012].

Considering the kinds of attacks, different taxonomies are available in literature [Igure, 2008; Hansman, 2005; Mirkovic, 2004; Cambiaso, 2013]. Among the wide range of attacks, some of the investigated categories are described in the following:

- Denial of service attacks: denial of service attacks (DoS) are aimed at preventing legitimate users to access a service or network [Hansman, 2005]. In this case, different approaches are possible, exploiting the different stacks of the ISO/OSI model: first generation attacks are mainly characterized by flooding the victim with high amount of network traffic [Mirkovic, 2004], while last generation threats are characterized by low attack bandwidth requirements [Cambiaso, 2013]. The execution of a coordinated DoS attack with the same aim from more than a single attacking node is known as distributed denial of service (DDoS). In this case, a botnet of attacking nodes is involved [Farina, 2016]. In history, different DDoS attacks were perpetrated against large companies [Bhardwaj, 2018], also adopting recently introduced methodologies exploiting communication protocols such as the DNS protocol to perpetrate a distributed amplification and reflection denial of service attack (DRDoS) [Deka, 2017; Ryba, 2015].
- Malicious software: such tools are commonly known as trojan horses, time-bomb, logic-bomb, spyware, virus, malware, worm, etc. [Landwehr, 1994]. Concerning trojan horses, a term introduces by Dan Edwards in 1972 [Anderson, 1972] and later refined [Landwehr, 1994], they refer to software programs that appear as useful and legitimate tools, but exploit user's rights for unintended purposes. A trojan horse can replicate itself by adopting different approaches: in case of software replication, we talk of a virus, while a code replicating on system's processes and files is commonly known as a worm; finally, in case of replication on the network, malwares are involved. Injected malicious software may exploit specific vulnerabilities and bugs of the hosting machine, for instance, to make the attacker gain administrator privileges [Ou, 2005]. Nowadays, it is therefore important to consider protection from malicious software, since they could spread in different ways [Zhu, 2012] and they could compromise system functioning or security and privacy.
- Packet forging and replay attacks: packet forging attacks are executed to compromise the identity of the source of the data, to craft and send network packets to a recipient, for malicious purposes [Namratha, 2016]. Replay attacks are a similar attack, sending data previously acquired (if needed, with minor changes) in a later moment in time, to induce some sort of behavior on the victim [Goyal, 2010; Pries, 2008]. Such kind of threats is strictly related to man-in-the-middle (mitm) attacks, executed by placing the attacker between the sender and the recipient of the message, from a physical or logical point of view [Conte, 2016], to spoofing attacks, executed by passively eavesdropping exchanged network packets to derive valuable information [Goyal, 2010], and to fingerprinting attacks, executed to passively infer information on the exchanged data [Abe, 2016].
- Covert channels: covert channels consist in the adoption of methods able to bypass security policies or leak sensitive data outside of an organization [Aiello, 2016]. This kind of threats is particularly dangerous, since it makes use of legitimate protocols (such as DNS, or HTTPS) to incapsulate malicious packets (tunnelling). Moreover, although convert channels refer to the insider threat problem [Colwill, 2009], in case of an infection of an internal server of the platform, it is theoretically possible for the attacker to behave in a malicious way. Hence, it is important to consider protection

from this kind of threats, for instance, by deploying ad-hoc algorithms or protection systems [Sheridan, 2015] to monitor network behaviour.

Other kind of attacks can be executed for specific purposes. For instance, phishing [Wu, 2006] (often involved to inject malicious software on a host) or spam [Wood, 2016] attacks are important threats for any organization. Nevertheless, given the context of the MHMD platform, we prefer to focus on network attacks technicalities, not investigating in detail social engineering aspects of the cyber-security context, since it depends on the experience and knowledge of the personnel involved in the systems and of the users of the platform.

3.3 Blockchain and transactions security

An important component of the MHMD platform is composed by the blockchain system. Such component is integrated in the architecture and it represents a core aspect for sensitive data management. By analysing the blockchain, it is composed of several nodes communicating on the network and sharing data. Hence, three different aspects should be analysed: security of communications and the entire blockchain network (strictly related to Section 3.2), users security and privacy, and data security. Concerning data security, blockchain data is represented by transactions, where a single transaction is structured in a specific way, as designed by the developers of the system.

Previous works on the topic mainly focus on crypto-currency contexts, often characterized by a public blockchain without a central control [Lin, 2017]. Concerning blockchain network and components security, an important and well-known attack is composed by the majority attack. In particular, in this case, if the enemy controls more than 51% of the nodes (or the resources, such as computing power, in function of the mining algorithm adopted), it is possible to mine blocks quicker, hence having the authority to discard/accept specific blocks, hence including falsified transactions [Eyal, 2018]. Although this vulnerability mainly affects public blockchains, it may become relevant even for private ones, since mining hosts may be targeted by cyber-criminals, that may take control of the entire network and pass unobserved even for months or even years [Nagle, 2014]. Another relevant problem affecting public blockchain is presented by forks [Back, 2014]. In this case, a disjointed agreement of blockchain software and protocol upgrades, that may lead to network split, hence, to the existence of two (or more) different blockchains sharing the first set of blocks [Lin, 2017]. Instead, the investigation of well-known distributed denial of service (DDoS) attacks against a blockchain based network system is accomplished in [Park, 2017]. In this case, a volumetric DDoS attack is considered, by overwhelming network resources of the targeted nodes. This kind of threats, also considering low-rate approaches [Cambiaso, 2013], should be avoided, in order to avoid a malicious user to dismantle the entire network or parts of it. Similar attacks exploit the Border Gateway Protocol (BGP), by executing hijacking activities [Apostolaki, 2017; Natoli, 2017]. In this case, the aim of the attacker is to accomplish network traffic interception, by exploiting the BGP routing protocol, e.g., to re-route network traffic to a node under the control of the attacker. For instance, in case of redirection of mining pools, it would be possible to “steal” crypto-coins from the targeted hosts. Although this attack is a serious threat, it requires the attacker to control the BGP routing process. Also, strong authentication of the two communicating parties may be able to protect from such attack, since the bogus node would not be able to authenticate on the network (although previous knowledge of the nodes is required). Concerning instead security of blockchain nodes, the eclipse attack allows an attacker to take control of incoming and outgoing connections of a given target, hence potentially isolating the victim from the other peers [Heilman, 2015]. Such kind of control allows an

attacker to manipulate the victim's view of the blockchain or to induce the victim to behave differently from expected [Li, 2017].

Relatively to blockchain users' security, [Park, 2017] considers a scenario where the attacker targets the personal encryption keys adopted by the users. In this case, the physical host used by the attacker is usually exploited, by installing malicious software able to leak the private keys of the user, hence allowing the attacker to impersonate the victim [Barber, 2012]. Hardware wallets were introduced to protect from private keys theft [Hurlburt, 2014]. Other studies suggest to strengthen authentication and access to sensitive information, for instance by adopting multi-factor authentication methods [Mann, 2017]. Blockchain users privacy is also investigated in [Androulaki, 2013], studying the adoption of clustering algorithms on the Bitcoin network and proving that information of a significative percentage of users can be identified. [Koshy, 2014] tries instead to correlate crypto-wallets to IP addresses, by analyzing real-time transactions traffic. Since such kind of attacks are extremely serious and should not be underestimated, during the design and development process of a new blockchain based system, it is important to guarantee users security and privacy.

Concerning instead blockchain data security aspects, an important topic is characterized by smart contracts. Smart contracts are scripts that are stored in the blockchain, characterized by a unique address and triggered by binding a transaction to the smart contract address [Christidis, 2016]. [Li, 2017] analysed the security of smart contracts, by studying their exploitation by cyber-criminals. In this context, the criminal smart contracts (CSC) phenomenon refer to the injection of smart contracts specifically designed to leak confidential information or to trigger real-world crimes [Juels, 2016]. As an example of smart contract exploitation, in 2016, cyber-criminals targeted the DAO smart contract, exploiting a recursive call vulnerability to stole crypto-money [Bahga, 2016]. Similarly, in 2014, transactions mutability exploitation lead to stole a large amount of crypto-money to MtGox, one of the most important crypto-trading services of the time [Decker, 2014].

By analysing smart contracts vulnerabilities in detail, [Atzei, 2017] introduces different smart contracts affected by vulnerabilities, based on a classification of the vulnerabilities that affect smart contracts, by focusing on Ethereum smart contracts.

Vulnerability	**Cause**	**Level**
Call to the unknown	The called function does not exist	Contract source code
Out-of-gas end	Fallback of the callee is executed	
Exception disorder	Irregularity in exception handling	
Type casts	Type-check error in contract execution	
Re-entrancy vulnerability	Function is re-entered before termination	
Field disclosure	Private value is publisher by the miner	
Immutable bug	Alter a contract after deployment	EVM bytecode
Ether lost	Send ether to an orphan address	

Stack overflow	The number of values in stack exceeds 1024	
Unpredictable state	State of the contract is changed before invoking	Blockchain mechanism
Randomness bug	Seed is biased by malicious miner	
Timestamp dependence	Timestamp of block is changed by malicious miner	

Table 1: A classification of smart contracts vulnerabilities

It is therefore important to analyse and identify in advance possible weaknesses on the smart contract, hence, on the implemented blockchain, in order to avoid exposure to vulnerabilities and exploitation from external entities.

3.4 Services security

Concerning Internet services security, an historically important topic concerns cloud security [Almorsy, 2016]. In this context, several works focus on identifying security issues on different cloud solutions, such as infrastructure-as-a-service (IaaS) [Gonzales, 2017], platform-as-a-service (PaaS) [Hussain, 2017], and software-as-a-service (SaaS) [Tiwari, 2016]. Nevertheless, in this context it is also important to guarantee other security aspects like access methodologies [Thanh, 2015] or cloud data protection [Rao, 2015]. Simultaneously, it is also worth mentioning the recent evolution of the cloud paradigm, in favour of the fog computing [Thota, 2018]. Although the MHMD platform can in practice be represented as a cloud-based platform, since several Internet services are web based (through web services or interfaces), web applications security assumes a crucial role.

Considering the security of web applications, even if the denial of service field is characterized by the appearance of the last generation attacks aimed at targeting web application daemons [Cambiaso, 2015], standing to [Scott, 2002], application level web security can be categorized in three different topics: attacks to web forms, Structured Query Language (SQL) attacks, and cross-site scripting (XSS). Concerning web forms attacks, they are strictly related to the phishing threat, exploiting web forms to retrieve sensitive user data [Wu, 2006b]. Considering such three threats categories, [Scott, 2003] extended the Security Policy Description Language to evaluate the efficiency of a protection system aimed at protecting from such attacks. The concept behind this "three threats" distinction can be still applied today. Indeed, [Dwivedi, 2014] proposes a study of the top three web application vulnerabilities, including injection, XSS, and authentication and session exploitation, also discussing possible protection systems to counter such threats.

Regarding SQL attacks, the most important threat is represented by SQL injection attacks (SQLi), a specific type of injection attack aimed at entering strings leading to the execution of illegal queries on the underlying database systems [Pawar, 2015]. [Halfond, 2006; Alwan, 2017] investigate the different types of SQLi attacks, by also presenting and discussing the advantages and disadvantages of different protection approaches. In a similar work, [Verma, 2015] categorizes both SQLi vulnerabilities and attacks, by also reporting some sample implementations of malicious queries. The work also analyses protection approaches, stating that the positive tainting approach, implemented on the Web Application SQL injection Preventer (WASP) tool [Halfond, 2008], results the best protection system, since in the executed tests it is able to block attacks without generating false positives.

By combining a study of both SQLi and XSS attacks and vulnerabilities, [Johari, 2012] presents and discusses various types of threats, with relative protection systems, and reporting the possible weaknesses leading to such attacks, including partial implementations, adoption of complex frameworks, and runtime overheads. XSS attacks, concerning the injection of client-side scripts into web pages accessed by other users [Vogt, 2007], are also investigated in [Jayamsakthi, 2008], presenting a survey of vulnerabilities related to cross-site scripting attacks. Concerning protection from XSS attacks, [Gupta, 2017] proposes a categorization of the threats on the (i) persistent, (ii) non-persistent, and (iii) DOM based categories, by also proposing a process to verify if a website is vulnerable to XSS attacks, and proposing a survey of state-of-the-art protection systems, by concluding that a single defense system able to protect from each kind of XSS does not exist.

3.5 Applications security

Concerning applications security, the topic is strictly related to the software security concept, related to "the idea of engineering software so that it continues to function correctly under malicious attack" [McGraw, 2004]. This concept is extremely important, since it involves programming and architectural choices, that, due to their nature, are prone to the injection of software vulnerabilities. In particular, bad software programming may lead to the exposure of software bugs, with consequent exposure to cyber-attacks [Bellissimo, 2006]. It is therefore crucial to consider application security as a key aspect for the security of the overall system. Considering the MHMD platform, the application security field is limited to web security, mobile app security and secure smart contract development. Considering web security, it has already been deeply investigated in Section 3.4, while secure smart contract aspects are already analysed in Section 3.3.

Concerning instead the Android ecosystem, it is a well-known fact that it is particularly exposed to malware [Merlo, 2014], not only because of its open-source nature, but also in virtue of the common "copy-and-paste" approach adopted by many developers, unwillingly injecting vulnerabilities on their mobile applications (apps) [Fisher, 2017]. Considering mobile malware, [Faruki, 2015] investigates the Android security topic, by discussing attack techniques and relative protection. In this context, signature-based methods aren't enough to efficiently identify malicious applications (apps), while anomaly-based approaches should be also considered [Faruki, 2015].

An important aspect in the Android apps security topic concerns app permissions. Each Android app declares a set of permissions required to run the underlying code. For instance, the geolocation permission is required to access user's location for navigation purposes, while the Internet permission is required to communicate on the net. Although it may seem a good approach, the permission-based access control mechanism implemented in Android is vulnerable to privilege escalation attacks, for instance by executing attacks exploiting vulnerable software components [Heuser, 2016]. The detection of privilege escalation attacks accomplished by exploiting interactions between apps is accomplished in [Sadeghi, 2015], while an exhaustive study on Android permission is accomplished in [Armando, 2015], stating that security policies over permissions is not supported by the operating system, hence, they are implemented in the logic of the application, with consequent potential exposure to security breaches. Also, by focusing on BYOD contexts, [Costa, 2018] considers automatic security assessment of apps configurations.

In particular, in order to analyse an app two different approach are generally possible: static and dynamic code analysis. While static code analysis scans/interprets the application code without installing it, dynamic code analysis runs the app in an isolated environment (sandbox), hence analysing the behaviour of the app [Blasing, 2010]. Static code analysis is investigated in [Bonett, 2018], proposing a framework to evaluate

different static analysis tools. Concerning dynamic code analysis, [Petsas, 2014] studies the different techniques to hinder it, while [Visas, 2014] focuses on the detection of sandbox. Static and dynamic code analysis approaches are combined in [Spreitzenbarth, 2015], by adopting machine learning techniques for automatic analysis of apps. Similarly, [Blasing, 2010] combines the approaches by proposing AASandbox, an ad-hoc Android application sandbox.

It is therefore important to properly evaluate security aspects of mobile applications, in order to prevent exposure to vulnerabilities and consequent exploitation from malicious users.

3.6 Data security

Concerning data security, tokenization methods [Paryasto, 2014] refer to the process to replace sensitive data with non-sensitive equivalent, commonly identified with the term *token*. This approach, requiring an isolated and secure tokenization server, is in general adopted to provide data protection. Although the term data security is often also coupled with data privacy aspects [Chen, 2012; Stahlberg, 2007], database forensics [Guimaraes, 2010; Frühwirt, 2010], encryption [Pandey, 2015] or stenography [Artz, 2001] and other data hiding methodologies [Moon, 2007], our focus in the following is on database protection, since databases represent a crucial element for the MHMD platform.

Database protection is strictly related to SQL injection threats [Chandrashekhar, 2015], a topic already covered in Section 3.4. Being SQLi an important threat to database systems, the protection from such kind of threats is scope of several research works. In this context, [Dawle, 2017] proposes indeed a database Intrusion Detection System (IDS), working by logging activities of intruders performing SQL injection attacks. Data is protected by the AES encryption algorithm and results prove that it is possible to prevent SQLi attacks. Database protection is indeed strictly related to data encryption. In this context, [Singh, 2015] analyses in depth different database encryption techniques, comparing them and evaluating their advantages and disadvantages. [Basharat, 2012] investigates instead how encryption can be adopted to provide security, considering a multi-layer approach, and stating that database encryption should be coupled with proper algorithms able to guarantee data integrity. Concerning database authentication security, [Kumar, 2016] investigates database authentication and proposes ad-hoc authentication policies and management rules, coupled with data analysis, in order to guarantee database security. [Hamilton, 2017] analyses instead the possibility to adopt multi-factor authentication [Dasgupta, 2017] on database systems. Other approaches make use of watermarking techniques to manage data access [Zhang, 2006] and integrity [Bhattacharya, 2010]. In the database security context, it is also important to analyze system logs and records in order to identify possible abuses. In this context, [Chaudhari, 2015] focuses on the identification of auditing records from database management systems (DBMS) logs directories in order to identify suspicious activities. A similar approach is proposed in [Panda, 1999], considering logs analysis to reconstruct a corrupted database. [Zuo, 2004] proposes instead a distributed database damage assessment by considering both a centralized and peer-to-peer approach.

Relatively to database management systems protection, it is also important to consider NoSQL database, often associated to big data storage and analysis activities [Han, 2011]. In this context, two of the most popular NoSQL database management systems, Cassandra and MongoDB, are analysed in [Okman, 2011], identifying the most important security concerns of such systems. Particularly, both the systems do not properly implement data encryption, although recent implementations are able to solve the issue [Tian, 2014]. Another issue found in [Okman, 2011] is related to weak authentication support and exposure to SQL

injection and denial of service attacks. Therefore, as also stated in [Ron, 2015], NoSQL DBMS suffer the same security risks as standard SQL systems, and proper defence methodologies should be implemented.

3.7 Other security aspects

Although the mentioned topics are extremely important, it is also important to consider innovative (0-day) threats. In this context, in history, we have assisted to different relevant attacks like large scale threats targeting SCADA networks through Stuxnet [You, 2014], Internet of Things devices through the Mirai botnet [Kolias, 2017], or world-scale service providers through large-scale distributed DoS attacks [Mansfield-Devine, 2016]. Also, important vulnerabilities were published, relatively to both software libraries and tools, including the Heartbleed bug exploiting TLS secure connections [Durumeric, 2014], and hardware systems, exploited for instance through the Spectre and Meltdown flaws in modern micro-processors [Prout, 2018]. It is therefore important to consider that (i) continuous security updates should be executed, in order to protect the system from novel threats and vulnerabilities, and (ii) it is impossible to implement a complete and final protection system, since any system is potentially vulnerable to 0-day threats. Nevertheless, anomaly detection systems may be implemented on the system to identify unknown behaviours, potentially related to novel cyber-attacks [Ahmed, 2016; Bhuyan, 2014].

4 Concepts from security standards

In general, it is recognized that the regulation and standards documents contribute to the definition of appropriate approaches and methodologies able to guarantee system security, from both technological and organization point of views [Vorobiev, 2010]. With regard to security standards, several standards are available [Tofan, 2011]. Among the most important standards providers in the cyber-security contexts, we mention the International Standards Organization (ISO), providing standards in different contexts, including information security standards [Disterer, 2013; Tofan, 2011], the National Institute of Standards and Technology (NIST), publishing a framework on cyber-security [Chang, 2016], the European Union Agency for Network and Information Security (ENISA) [Chen, 2010], an European centre with advanced network and information security expertise, and the Open Web Application Security Project (OWASP) [Srinivasan, 2017], an online community sharing documents, tools and methodologies on cyber-security, focusing on web applications and mobile security. In this section of the document, we report the most relevant security standards for the MHMD platform, with the goal to consider them for further development or integration from the developers of the MHMD system.

When it comes to security and cyber-security, an important aspect to consider also concerns users' privacy. In this context, although different regulations influence different countries [Greenleaf, 2012], the General Data Protection Regulation (GDPR), widely considered for the development of the MHMD project, plays a crucial role in Europe, applying directly to management and processing of personal data related to the European territory or market [Albrecht, 2016]. Therefore, concerning users' data privacy, it is crucial to implement a system able to comply to the current regulations. In the privacy standards category, the NIST SP 800-53A framework "Assessing Security and Privacy Controls in Federal Information Systems and Organizations", focused on both security and privacy, "provides a set of procedures for conducting assessments of security controls and privacy controls employed within federal information systems and

organizations"[9]. Similarly, the ETSI TR 103 306 report "Cyber: Global Cyber Security Ecosystem" reports information on cyber-security aspects to consider to protect a network system[10].

Considering information security management, the ISO/IEC 27000-series of standards on "Information technology - Security techniques - Information security management systems" includes the ISO/IEC 27000 "Overview and vocabulary", the ISO/IEC 27001 "Requirements", and the ISO/IEC 27002 "Code of practice for information security controls". The application of such standards mostly concerns the context of an organization. In particular, the ISO/IEC 27002:2013 standard [Disterer, 2013] covers different security aspects concerning information security, such as personnel security, physical security, access control, cryptography policy management, system security management, or security incident management. This standard defines a set of controls to be implemented on the organization in order to guarantee adequate protection of all the components of the organization. A similar standard is the "Security framework of trust service providers" by ENISA, reporting details on how to accomplish risk assessment, analyse the risk of a threat to assess the impact, evaluate it and mitigate it. The ISO/IEC 15408 "Information technology - Security techniques - Evaluation criteria for IT security - Part 1: Introduction and general model" and ISO/IEC 18045 "Information technology - Security techniques - Methodology for IT security evaluation" standards provide instead valuable information on the analysis of software design vulnerabilities.

Finally, the ISO/IEC 27799 standard "Health informatics - Information security management in health using ISO/IEC 27002" provides a set of guidelines for information security organization and management, by supporting the interpretation and the implementation of the ISO/IEC 27002 standard previously mentioned. Given the context of the MHMD platform, such standard represents an important aspect to consider, especially in view of an integration on the project, aimed at providing certifications to the platform.

5 MHMD Security Infrastructure

In this section of the document we report the MHMD Security Infrastructure related to the system, able to guarantee adequate protection[11]. As mentioned above, the proposed MHMD security infrastructure considers three different factors: (i) the current/existing version of the MHMD platform, its components and their interactions, (ii) state-of-the-art solutions in the security context, by evaluating the aspects previously investigated, and (iii) the possible consideration of security and privacy related standards and regulations. Combining these factors, we are able to provide a security infrastructure capable of considering different contexts, focusing on the aspects that are critical for the MHMD platform, and also evaluating aspects for exploitation and scalability, through standardization procedures and regulations. Also, the security of the entire MHMD platform is improved thanks to the MHMD distributed Intrusion Detection System (MHMD-dIDS), an innovative Intrusion Detection System aimed to monitor and protect the entire system.

---

[9] More information can be found at the following URL (revision 4):
https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-53ar4.pdf

[10] More information can be found at the following URL (v1.1.1):
https://www.etsi.org/deliver/etsi_tr/103300_103399/103306/01.01.01_60/tr_103306v010101p.pdf

[11] Please note that at this stage of the project, hacking and privacy breaking activities are not completed. Therefore, security and privacy issues may affect the system.

5.1 Preliminary analysis

Referring to the threats in Section 3, we report in Table 2 a summary of current threats and possible solutions that can be implemented.

Context	Major threats	Countermeasures
Encryption	Data confidentiality breaking.	Adoption of strong encryption algorithms, end-to-end encryption.
Network	Cyber-attacks like denial of service, malicious software, packet forging and replay, covert channels.	Network segmentation, VPN, policy and access control, log analysis, adoption of secure communication protocol, network appliances (IDS, IPS, antivirus, network monitors, etc.), adequate personnel training.
Blockchain and transactions	Network attacks to the system (e.g. denial of service), exploitation/injection of malicious transactions.	Network protection, secure software development, adoption of "anti-majority" approaches.
Services	Exploitation of insecure communications, SQL injection, cross-site scripting.	Network protection, secure software development, tools and approaches to protect from SQLi and XSS attacks.
Applications	Exploitation of software bugs.	Secure software development and deep testing activities, user access and communication restrictions, adoption of strong and secure communications and strong data encryption.
Data	SQL injection, data leaks.	Access control and user restrictions, strong authentication methods, database IDS and log analysis, tokenization methods.

Table 2: Possible countermeasures to adopt for state-of-the-art major threats

In particular, concerning encryption solutions, further work may be directed on the evaluation of the adoption of (partial) homomorphic encryption schemes, or to consider the potential investigation of quantum resistant algorithms. Concerning instead blockchain and applications, further work may be directed on the consideration of the adoption of hardware wallets, hardware token or other approaches able to provide multi-factor authentication. Relatively to software-based systems in particular, it is instead crucial to implement security updates as soon as they are released, in order to prevent exploitation of known vulnerabilities.

Considering instead the security standards reported in Section 4, we will now focus in particular on the ISO/IEC 27002:2013 standard [Disterer, 2013], since it covers different security aspects concerning ICT based systems, independently on the context of the system. This standard defines the procedures to adopt for the management of key security aspect within an organization. For instance, by referring to the "Cryptography Policy Management" aspect (part 10 of the standard), part 10.2.1 concerns the implementation of cryptographic key policies, by considering for example key generation and distribution procedures. Similarly, the "Network Security Management" aspect (part 13 of the standard), an in particular to the part 13.1.3 of the standard, it refers to network segregation. In this context, it is important to provide appropriate segregation of the information systems, services and users. Another important aspect is part 13.2.3, concerning the protection of information sent using electronic messaging systems. Being MHMD a distributed platform, composed of several components owned by different entities, it is important to consider that to comply to a standard like ISO/IEC 27002, this would involve the compliance from all the parties involved in the architecture. The compliance to most of the mentioned standards and regulations should be investigated by the developers of the MHMD platform, although it may not be reached easily and/or it may even not be needed. Nevertheless, it is important to mention that standards compliance would improve security of the overall system, hence it should be seriously evaluated.

Finally, considering the MHMD platform in the current state, scope of Section 2, by referring to Figure 2, we find that almost all the contexts are considered, with the exception of the Physical (for our aim, physical attacks are not considered in deep, being the MHMD platform a distributed platform, and being the final focus hacking challenge activities) and the Desktop one (since, to date, no desktop apps interacting with the system are available).

5.2 MHMD security considerations

It is therefore important to consider the security aspects previously mentioned, for each context (see Table 2). For instance, in order to provide system protection against malicious software such as virus and trojan horses, proper defence systems should be implemented, including an accurate management of users' privileges and access control [Shan, 2012], ad-hoc appliances [Everett, 2005] and specific Intrusion Detection Systems (IDS), designed to identify both known and novel threats [Idika, 2015]. Simultaneously, it is crucial to define and adopt post-mortem analysis methodologies [Ardi, 2009], in order to properly manage incidents and guarantee service continuity. This topic is widely adopted in literature [Hilgers, 2014; Spyridopoulos, 2013; Chandia, 2009] and it is even analysed in the ISO/IEC 27002 (see part 16 and part 17).

In addition, it is crucial to deeply analyse the implemented procedures, in order to identify possible flaws and concepts to exploit. For instance, since the Web Server and the Upload Server agree on an upload token to adopt, they may exchange the token, or they may adopt a common token generation algorithm. In the first case, how is this token exchanged? By using which protocols? Is the communication secure? Is client/server authentication implemented? In the second case, how strong is such algorithm? What would it happen if it would be executed from a third entity? These are all questions that should be implemented.

Similarly, considering services security and the application layer in particular, it is important to work hard on the implementation aspects, by adopting a "security by design" development approach and accurately evaluating libraries adoption and software updates. [Scott, 2003] presents a set of tools to support programmers for the development of secure applications able to counter a wide range of common attacks to web applications. Considering smart contracts published on the blockchain, as previously mentioned,

[Atzei, 2017] provides important suggestions to prevent exploitation of vulnerabilities in smart contracts. The blockchain adoption in the healthcare sector is also investigated in [Azaria, 2016], also reporting details relatively to authentication, confidentiality, accountability and data sharing. Concerning instead encryption algorithms, as previously suggested, end-to-end encryption is suggested. In addition, [Caviglione, 2017] reports a set of secure algorithms to adopt and another set to discard, considering both the security level of the algorithm and green aspects related to power consumption.

In order to provide a secure system, it is therefore important to adopt both software and hardware state-of-the-art approaches, solutions and algorithms to protect the system from both known and unknown (0-day) threats. In addition, it is crucial to design and develop secure software, by adopting a "security by design" approach and to prevent exploitation of software bugs or security holes. Relatively to data storage, secure storage should be guaranteed, through the adoption of proper encryption methodologies and appropriate access control mechanisms. Finally, considering network communications, it is important to guarantee data confidentiality, availability and integrity, by implementing security countermeasures, for instance, to monitor network traffic to block running attacks.

5.3 MHMD distributed Intrusion Detection System

In order to protect the overall MHMD system and its components, it is important to deploy appropriate security modules, aimed to guarantee security of the entire platform. In this context, with the aim to enrich the security of the system, the MHMD distributed Intrusion Detection System (MHMD-dIDS) was designed, a security framework to deploy a distributed and multi-context Intrusion Detection System. The aim of this component is to detect intrusions on the platform, hence, exploitation of weaknesses by malicious users.

Given the sensibility of the data managed by the platform, it is important to identify unwanted activities executed by malicious users. The MHMD-dIDS is represented by a set of components integrated into the system and transparently analyzing the network traffic on sensitive locations. Considering for instance network traffic analysis, such transparent analysis may be accomplished by physically interrupting (and forwarding) the network traffic (the same approach usually adopted by a network firewall), or by working on mirrored traffic, hence, in a passive and less invasive way.

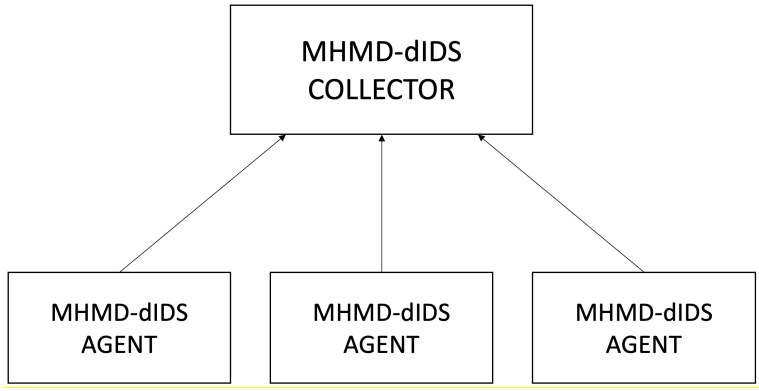

Figure 4: The high-level architecture of the MHMD-dIDS

As shown, the MHMD-dIDS is composed of two different kinds of components:

- MHMD-dIDS Collector: it is the main component of the MHMD-dIDS, aimed to provide a graphical interface (GUI) to the user. This component is supposed to collecting alert data from the secondary components distributed on the platform, in order to shown important messages to the user.
- MHMD-dIDS Agent: such components are replicated in the premises of each data provider and they are supposed to accomplish the following activities: (i) communication with the internal (distributed) modules of the data provider and/or network components of the system; (ii) run the (local) Intrusion Detection System on the data generated; (iii) send relative reports (alerts, mainly) to the MHMD-dIDS Collector component.

The adopted approach is based on a local data computation, accomplished inside of the data provided, by the MHMD-dIDS Agent components. This approach guarantees data privacy, since data or a representation of them is not exported outside of the local entity (data provider).

According to the definition reported in [Cambiaso, 2016], intrusion detection activities executed by the MHMD-dIDS Agent involve (i) the extrapolation and retrieval of data by adopting specific pre-defined metrics; (ii) data analysis, by elaborating the retrieved data and comparing it the scenario with an analogous one related to legitimate situation, as resulted, for instance, after previously executed training activities; and (iii) the characterization of the current situation as legitimate or anomalous, by adopting specific thresholds. By using the same components, the MHMD-dIDS Agents are composed of different components.

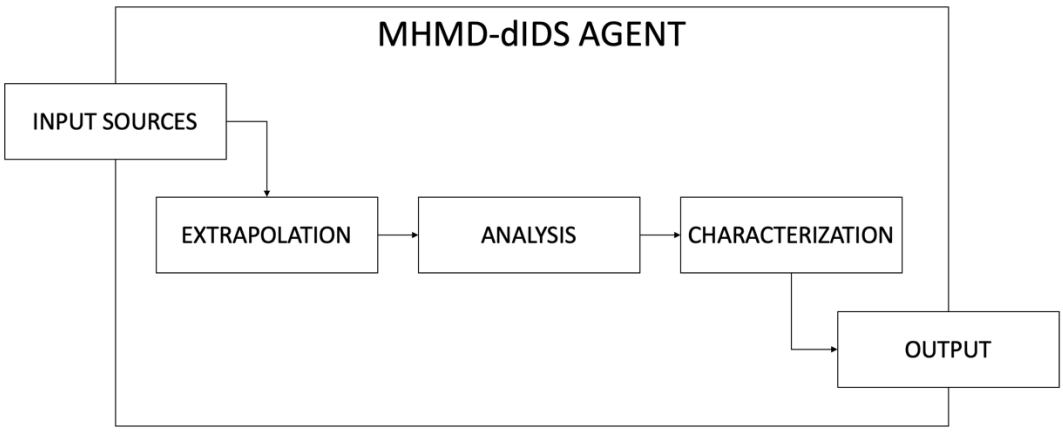

Figure 5: The high-level architecture of the MHMD-dIDS Agent

Particularly, the agent module is composed of different internal modules interacting among themselves: as previously mentioned, input sources are connected to network components of the MHMD platform and provide data useful for the three steps concerning data extrapolation, analysis, and characterization (as previously described); at this point, after data characterization is accomplished, the output module is aimed to forward resulted data to the MHMD-dIDS Collector. Such data contains alerts, hence, the fact that an anomalous scenario/situation is found on the system, with related data. Optionally, legitimate reports may be shared with the MHMD-dIDS Collector also. In addition, the output module may communicate with (local) network administrators (related to the data provider entity), for instance by notifying alerts to syslog servers or through email-based communications.

The adopted architecture provides the possibility to accomplish local replication of MHMD-dIDS Agents, inside of the same data provider. By adopting such approach, it is possible to monitor different data sources, hence making MHMD-dIDS a distributed and multi-contextual Intrusion Detection System. The evaluation of

such approach is not in the scope of the current document, but, given the scalability of the proposed architecture, it is possible to extend a single-context solution, by monitoring more input sources, hence providing the system the ability to identify different kinds of attacks and/or attacks targeting entities of different nature.

Thanks to the MHMD-dIDS component, the MHMD platform is enriched with a component designed to execute specific intrusion detection algorithms aimed to identify anomalies on the network. Since signature-based detection approaches are not always sufficient to identify cyber-attacks (e.g. in case of 0-day attacks or advanced threats), the conjunction with an anomaly-based Intrusion Detection System represents an important step in order to improve security of the overall system.

5.4 MHMD Security Infrastructure guidelines

In order to provide a security infrastructure, by making use of the previous considerations, in this section of the document we report a set of guidelines to adopt for the development of the system, in order to guarantee security of the overall MHMD system. The following list of rules/guidelines have to be considered, in order to protect the system in view of the penetration testing activities. It is worth mentioning that other optional guidelines, to be evaluated for future integration in the project, concern details on personnel security management and physical security aspects. In addition, all the aspects not explicitly mentioned in the following list should be considered optional.

1. Encryption
    a. All communications of the components are encrypted
    b. End-to-end encryption is adopted for communications
    c. All data stored by the component are encrypted
    d. Encryption/decryption keys are binded to a specific entity (e.g. data set, user, etc.)
    e. Encryption/decryption keys are not stored in memory for long time periods
    f. Adopted encryption algorithms do not include unreliable algorithms (such as DES and RC4)
    g. The adoption of “custom” encryption methods is accurately evaluated, from the security point of view (e.g. encryption of a portion of a file, double encryption using different algorithms, etc.)
2. Communications and network security
    a. Component’s hosts are placed in a dedicated and segmented network (e.g. DMZ), separated from other nodes (e.g. workstations, other server systems, etc.)
    b. Component communications are protected by a network firewall (e.g. allowed ports, network limits, filtering, IDS/IPS, etc.)
    c. Components connections adopt secure encrypted protocols (linked to 1.a)
    d. Adoption of user authentication methods, possibly combining both server-side and client-side authentication methods
    e. Adoption of strong authentication credentials
    f. Adoption of certificate pinning methodologies
    g. Anonymizing and/or tokenization servers are isolated from the network, except the (single) node they are supposed to communicate with
    h. Network communications are protected by a network firewall
    i. Network policy and access control rules are deployed
    j. A network Intrusion Detection and Prevention System is deployed on the network

   k. Critical nodes are replicated on the network, in order to improve availability
3. Blockchain and transactions security
   a. Protection of the blockchain network (see 2)
   b. Adoption of secure network designing approaches (e.g. to counter majority attacks)
   c. Adoption of secure smart contracts development approaches (security by design, identification of code vulnerabilities and bugs, testing, etc.)
4. Services security
   a. Services protection at the network level (see 2)
   b. Adoption of host or network tools/modules/approaches to counter well-known attacks (for instance, SQLi, XSS, etc.)
5. Mobile applications security
   a. Request of the minimum permissions possible
   b. Accurate evaluation of external interfaces (like intents or content providers)
6. Data security
   a. Adoption of access control and user restriction methods (principle of the least privilege)
   b. Adoption of strong user credentials
   c. Data encryption is accomplished (linked to 1.c)
   d. Anonymized and/or tokenized data are not accessible after anonymization/tokenization is accomplished
   e. Adoption of a database Intrusion Detection System (if a database is present)
   f. Implementation of data integrity procedures
   g. Consideration of data tokenization methods (where appropriate)
7. Software security
   a. Adoption of secure software development approaches (security by design, identification of code vulnerabilities and bugs, testing, etc.)
   b. Adoption of third-party well-known software/software libraries/modules/code snippets (if any), only if implemented by trusted developers
   c. Adoption of code obfuscation techniques
   d. Avoidance of sensitive debugging logs printed in output and available to the user, even if through dedicated consoles
   e. Adoption of a security by design approach to design the workflows implemented in the platform
8. Hosting and organization security
   a. An operating system with updated security modules is adopted
   b. The component host is not running un-useful services (e.g. not needed web service)
   c. The component host is not running other services external to the project
   d. Users and administrator accounts are protected by strong password
   e. No sensitive password or connection data are stored on the system
   f. Remote connection services (e.g. remote desktop, remote shell, file sharing, etc.) are not running, or their network access is restricted to the only nodes connecting to them
   g. Network logs are collected and maintained
9. Continuous security
   a. Encryption keys replacing is considered, in case of leak
   b. Security updates are constantly deployed on the system
   c. Security issues or data leaks are promptly reported

d. Sensitive data backup procedures are deployed
e. Network security controls are periodically executed
f. Network traffic monitoring activities are periodically accomplished

Bibliography

[Abe, 2016] Abe, K., & Goto, S. (2016). Fingerprinting attack on tor anonymity using deep learning. *Proceedings of the Asia-Pacific Advanced Network*, *42*, 15-20.

[Agrawal, 2012] Agrawal, M., & Mishra, P. (2012). A comparative survey on symmetric key encryption techniques. International Journal on Computer Science and Engineering, 4(5), 877.

[Ahmed, 2016] Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, *60*, 19-31.

[Aiello, 2016] Aiello, M., Mongelli, M., Cambiaso, E., & Papaleo, G. (2016). Profiling DNS tunneling attacks with PCA and mutual information. *Logic Journal of the IGPL*, *24*(6), 957-970.

[Albrecht, 2016] Albrecht, J. P. (2016). How the GDPR will change the world. *Eur. Data Prot. L. Rev.*, *2*, 287.

[Almorsy, 2016] Almorsy, M., Grundy, J., & Müller, I. (2016). An analysis of the cloud computing security problem. *arXiv preprint arXiv:1609.01107*.

[Alwan, 2017] Alwan, Z. S., & Younis, M. F. (2017). Detection and Prevention of SQL Injection Attack: A Survey. *International Journal of Computer Science and Mobile Computing*, *6*(8), 5-17.

[Anderson, 1972] Anderson, J. P. (1972). *Computer Security Technology Planning Study. Volume 2*. Anderson (James P) and Co Fort Washington PA.

[Androulaki, 2013] Androulaki, E., Karame, G. O., Roeschlin, M., Scherer, T., & Capkun, S. (2013, April). Evaluating user privacy in bitcoin. In *International Conference on Financial Cryptography and Data Security* (pp. 34-51). Springer, Berlin, Heidelberg.

[Apostolaki, 2017] Apostolaki, M., Zohar, A., & Vanbever, L. (2017, May). Hijacking bitcoin: Routing attacks on cryptocurrencies. In *Security and Privacy (SP), 2017 IEEE Symposium on* (pp. 375-392). IEEE.

[Ardi, 2009] Ardi, S., & Shahmehri, N. (2009, March). A post-mortem incident modeling method. In *Availability, Reliability and Security, 2009. ARES'09. International Conference on* (pp. 1018-1023). IEEE.

[Armando, 2015] Armando, A., Carbone, R., Costa, G., & Merlo, A. (2015, July). Android permissions unleashed. In *Computer Security Foundations Symposium (CSF), 2015 IEEE 28th* (pp. 320-333). IEEE.

[Artz, 2001] Artz, D. (2001). Digital steganography: hiding data within data. *IEEE Internet computing*, *5*(3), 75-80.

[Atzei, 2017] ATZEI, Nicola; BARTOLETTI, Massimo; CIMOLI, Tiziana. A survey of attacks on ethereum smart contracts (sok). In: Principles of Security and Trust. Springer, Berlin, Heidelberg, 2017. p. 164-186.

[Azaria, 2016] Azaria, A., Ekblaw, A., Vieira, T., & Lippman, A. (2016, August). Medrec: Using blockchain for medical data access and permission management. In *Open and Big Data (OBD), International Conference on* (pp. 25-30). IEEE.

[Bahga, 2016] Bahga, A., & Madisetti, V. K. (2016). Blockchain platform for industrial internet of things. *Journal of Software Engineering and Applications*, *9*(10), 533.

[Baker, 2016] Baker, P. R., & Benny, D. J. (2016). Physical Security Planning. In *The Complete Guide to Physical Security* (pp. 22-31). Auerbach Publications.

[Barber, 2012] Barber, S., Boyen, X., Shi, E., & Uzun, E. (2012, February). Bitter to better—how to make bitcoin a better currency. In *International Conference on Financial Cryptography and Data Security* (pp. 399-414). Springer, Berlin, Heidelberg.

[Basharat, 2012] Basharat, I., Azam, F., & Muzaffar, A. W. (2012). Database security and encryption: A survey study. *International Journal of Computer Applications*, *47*(12).

[Bellissimo, 2006] Bellissimo, A., Burgess, J., & Fu, K. (2006, July). Secure Software Updates: Disappointments and New Challenges. In *HotSec*.

[Bernstein, 2009] Bernstein, D. J. (2009). Introduction to post-quantum cryptography. In *Post-quantum cryptography* (pp. 1-14). Springer, Berlin, Heidelberg.

[Bhardwaj, 2018] Bhardwaj, K., Miranda, J. C., & Gavrilovska, A. (2018, July). Towards IoT-DDoS Prevention Using Edge Computing. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*. USENIX Association.

[Bhattacharya, 2010] Bhattacharya, S., & Cortesi, A. (2010). Database Authentication by Distortion Free Watermarking. In *ICSOFT (1)* (pp. 219-226).

[Bhuyan, 2014] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network anomaly detection: methods, systems and tools. *Ieee communications surveys & tutorials*, *16*(1), 303-336.

[Bilge, 2012] Bilge, L., & Dumitras, T. (2012, October). Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM conference on Computer and communications security* (pp. 833-844). ACM.

[Blasing, 2010] Blasing, T., Batyuk, L., Schmidt, A. D., Camtepe, S. A., & Albayrak, S. (2010, October). An android application sandbox system for suspicious software detection. In *2010 5th International Conference on Malicious and Unwanted Software (MALWARE 2010)* (pp. 55-62). IEEE.

[Block, 2014] Back, A., Corallo, M., Dashjr, L., Friedenbach, M., Maxwell, G., Miller, A., ... & Wuille, P. (2014). Enabling blockchain innovations with pegged sidechains. *URL: http://www. opensciencereview. com/papers/123/enablingblockchain-innovations-with-pegged-sidechains*.

[Bonett, 2018] Bonett, R., Kafle, K., Moran, K., Nadkarni, A., & Poshyvanyk, D. (2018, August). Discovering Flaws in Security-Focused Static Analysis Tools for Android using Systematic Mutation. In *27th {USENIX} Security Symposium ({USENIX} Security 18)* (pp. 1263-1280). USENIX} Association}.

[Brakerski, 2014] Brakerski, Z., & Vaikuntanathan, V. (2014). Efficient fully homomorphic encryption from (standard) LWE. *SIAM Journal on Computing*, *43*(2), 831-871.

[Brewer, 2014] Brewer, R. (2014). Advanced persistent threats: minimising the damage. *Network security*, *2014*(4), 5-9.

[Cambiaso, 2013] Cambiaso, E., Papaleo, G., Chiola, G., & Aiello, M. (2013). Slow DoS attacks: definition and categorisation. *International Journal of Trust Management in Computing and Communications*, *1*(3-4), 300-319.

[Cambiaso, 2015] Cambiaso, E., Papaleo, G., Chiola, G., & Aiello, M. (2015). Designing and modeling the slow next DoS attack. In *International Joint Conference* (pp. 249-259). Springer, Cham.

[Cambiaso, 2016] Cambiaso, E., Papaleo, G., Chiola, G., & Aiello, M. (2016). A Network Traffic Representation Model for Detecting Application Layer Attacks. *Int. J. Com. Dig. Sys*, *5*(1).

[Canetti, 2017] Canetti, R., Raghuraman, S., Richelson, S., & Vaikuntanathan, V. (2017, March). Chosen-ciphertext secure fully homomorphic encryption. In *IACR International Workshop on Public Key Cryptography* (pp. 213-240). Springer, Berlin, Heidelberg.

[Caviglione, 2017] Caviglione, L., Gaggero, M., Cambiaso, E., & Aiello, M. (2017). Measuring the energy consumption of cyber security. IEEE Communications Magazine, 55(7), 58-63.

[Chandia, 2009] Chandia, R., Gonzalez, J., Kilpatrick, T., Papa, M., & Shenoi, S. (2007, March). Security strategies for SCADA networks. In *International Conference on Critical Infrastructure Protection* (pp. 117-131). Springer, Boston, MA.

[Chandra, 2014] Chandra, S., Paira, S., Alam, S. S., & Sanyal, G. (2014, November). A comparative survey of symmetric and asymmetric key cryptography. In *Electronics, Communication and Computational Engineering (ICECCE), 2014 International Conference on* (pp. 83-93). IEEE.

[Chandrashekhar, 2015] Chandrashekhar, A. M., Ahmed, S. T., & Rahul, N. (2015). Analysis of Security Threats to Database Storage Systems. *International Journal of Advanced Research in data mining and Cloud computing (IJARDC)*, *3*(5).

[Chang, 2016] Chang, V., Kuo, Y. H., & Ramachandran, M. (2016). Cloud computing adoption framework: A security framework for business clouds. *Future Generation Computer Systems*, *57*, 24-41.

[Chaudhari, 2015] Chaudhari, M. R. R., & Bakal, J. W. (2015). Overview of Database Auditing for Oracle Database. *Auditing*, *4*(7).

[Chen, 2010] Chen, Y., Paxson, V., & Katz, R. H. (2010). What's new about cloud computing security. *University of California, Berkeley Report No. UCB/EECS-2010-5 January*, *20*(2010), 2010-5.

[Chen, 2012] Chen, D., & Zhao, H. (2012, March). Data security and privacy protection issues in cloud computing. In *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on* (Vol. 1, pp. 647-651). IEEE.

[Chen, 2014] Chen, P., Desmet, L., & Huygens, C. (2014, September). A study on advanced persistent threats. In *IFIP International Conference on Communications and Multimedia Security* (pp. 63-72). Springer, Berlin, Heidelberg.

[Christidis, 2016] Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things. *Ieee Access*, *4*, 2292-2303.

[Colwill, 2009] Colwill, C. (2009). Human factors in information security: The insider threat–Who can you trust these days?. *Information security technical report*, *14*(4), 186-196.

[Conte, 2016] Conti, M., Dragoni, N., & Lesyk, V. (2016). A survey of man in the middle attacks. *IEEE Communications Surveys & Tutorials*, *18*(3), 2027-2051.

[Costa, 2018] Costa, G., Merlo, A., Verderame, L., & Armando, A. (2018). Automatic security verification of mobile app configurations. *Future Generation Computer Systems*, *80*, 519-536.

[Czerwinski, 2018] Czerwinski, D., Nowak, J., & Przylucki, S. (2018, June). Evaluation of the Jammers Performance in the WiFi Band. In *International Conference on Computer Networks* (pp. 171-182). Springer, Cham.

[Dasgupta, 2017] Dasgupta, D., Roy, A., & Nag, A. (2017). Multi-Factor Authentication. In *Advances in User Authentication* (pp. 185-233). Springer, Cham.

[Dawle, 2017] Dawle, Y., Naik, M., Vande, S., & Zarkar, N. (2017). Database Security using Intrusion Detection System. *Database*, *2*(03), 01-06.

[Decker, 2014] Decker, C., & Wattenhofer, R. (2014, September). Bitcoin transaction malleability and MtGox. In *European Symposium on Research in Computer Security* (pp. 313-326). Springer, Cham.

[Deka, 2017] Deka, R. K., Bhattacharyya, D. K., & Kalita, J. K. (2017). DDoS Attacks: Tools, Mitigation Approaches, and Probable Impact on Private Cloud Environment. *arXiv preprint arXiv:1710.08628*.

[Dijk, 2010] Van Dijk, M., Gentry, C., Halevi, S., & Vaikuntanathan, V. (2010, May). Fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 24-43). Springer, Berlin, Heidelberg.

[Disterer, 2013] Disterer, G. (2013). ISO/IEC 27000, 27001 and 27002 for information security management. *Journal of Information Security*, *4*(02), 92.

[Durumeric, 2014] Durumeric, Z., Li, F., Kasten, J., Amann, J., Beekman, J., Payer, M., ... & Halderman, J. A. (2014, November). The matter of heartbleed. In *Proceedings of the 2014 conference on internet measurement conference* (pp. 475-488). ACM.

[Dwivedi, 2014] Dwivedi, V., Yadav, H., & Jain, A. (2014). Web Application Vulnerabilities: A Survey. *International Journal of Computer Applications*, *108*(1).

[Ekert, 1996] Ekert, A., & Jozsa, R. (1996). Quantum computation and Shor's factoring algorithm. *Reviews of Modern Physics*, *68*(3), 733.

[Everett, 2005] Everett, C. (2005). Fortinet—so far, so good. *Infosecurity Today*, *2*(1), 32-34.

[Eyal, 2018] Eyal, I., & Sirer, E. G. (2018). Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*, *61*(7), 95-102.

[Farina, 2016] Farina, P., Cambiaso, E., Papaleo, G., & Aiello, M. (2016). Are mobile botnets a possible threat? The case of SlowBot Net. *Computers & Security*, *58*, 268-283.

[Faruki, 2015] Faruki, P., Bharmal, A., Laxmi, V., Ganmoor, V., Gaur, M. S., Conti, M., & Rajarajan, M. (2015). Android security: a survey of issues, malware penetration, and defenses. *IEEE communications surveys & tutorials*, *17*(2), 998-1022.

[Fischer, 2017] Fischer, F., Böttinger, K., Xiao, H., Stransky, C., Acar, Y., Backes, M., & Fahl, S. (2017, May). Stack overflow considered harmful? the impact of copy&paste on android application security. In *Security and Privacy (SP), 2017 IEEE Symposium on* (pp. 121-136). IEEE.

[Fontaine, 2007] Fontaine, C., & Galand, F. (2007). A survey of homomorphic encryption for nonspecialists. EURASIP Journal on Information Security, 2007, 15.

[French, 2014] French, A. M., Guo, C., & Shim, J. P. (2014). Current Status, Issues, and Future of Bring Your Own Device (BYOD). *CAIS*, *35*, 10.

[Frühwirt, 2010] Frühwirt, P., Huber, M., Mulazzani, M., & Weippl, E. R. (2010, April). InnoDB database forensics. In *2010 24th IEEE International Conference on Advanced Information Networking and Applications* (pp. 1028-1036). IEEE.

[Gentry, 2012] Gentry, C., Halevi, S., & Smart, N. P. (2012, April). Fully homomorphic encryption with polylog overhead. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 465-482). Springer, Berlin, Heidelberg.

[Gonzales, 2017] Gonzales, D., Kaplan, J. M., Saltzman, E., Winkelman, Z., & Woods, D. (2017). Cloud-trust—A security assessment model for infrastructure as a service (IaaS) clouds. *IEEE Transactions on Cloud Computing*, *5*(3), 523-536.

[Goyal, 2010] Goyal, P., Batra, S., & Singh, A. (2010). A literature review of security attack in mobile ad-hoc networks. *International Journal of Computer Applications*, *9*(12), 11-15.

[Greenleaf, 2012] Greenleaf, G. (2012). The influence of European data privacy standards outside Europe: implications for globalization of Convention 108. *International Data Privacy Law*, *2*(2), 68-92.

[Guimaraes, 2010] Guimaraes, M. A., Austin, R., & Said, H. (2010, October). Database forensics. In *2010 Information Security Curriculum Development Conference* (pp. 62-65). ACM.

[Gupta, 2017] Gupta, S., & Gupta, B. B. (2017). Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art. *International Journal of System Assurance Engineering and Management*, *8*(1), 512-530.

[Halfond, 2006] Halfond, W. G., Viegas, J., & Orso, A. (2006, March). A classification of SQL-injection attacks and countermeasures. In *Proceedings of the IEEE International Symposium on Secure Software Engineering* (Vol. 1, pp. 13-15). IEEE.

[Halfond, 2008] Halfond, W., Orso, A., & Manolios, P. (2008). WASP: Protecting web applications using positive tainting and syntax-aware evaluation. *IEEE Transactions on Software Engineering*, *34*(1), 65-81.

[Hamilton, 2017] Hamilton, C., & Olmstead, A. (2017, December). Database multi-factor authentication via pluggable authentication modules. In *2017 12th International Conference for Internet Technology and Secured Transactions (ICITST)* (pp. 367-368). IEEE.

[Han, 2011] Han, J., Haihong, E., Le, G., & Du, J. (2011, October). Survey on NoSQL database. In *Pervasive computing and applications (ICPCA), 2011 6th international conference on* (pp. 363-366). IEEE.

[Hansman, 2005] Hansman, S., & Hunt, R. (2005). A taxonomy of network and computer attacks. *Computers & Security*, *24*(1), 31-43.

[Heilman, 2015] Heilman, E., Kendler, A., Zohar, A., & Goldberg, S. (2015, August). Eclipse Attacks on Bitcoin's Peer-to-Peer Network. In *USENIX Security Symposium* (pp. 129-144).

[Heuser, 2016] Heuser, S., Negro, M., Pendyala, P. K., & Sadeghi, A. R. (2016, February). DroidAuditor: Forensic Analysis of Application-Layer Privilege Escalation Attacks on Android (Short Paper). In *International Conference on Financial Cryptography and Data Security* (pp. 260-268). Springer, Berlin, Heidelberg.

[Hilgers, 2014] Hilgers, C., Macht, H., Muller, T., & Spreitzenbarth, M. (2014, May). Post-mortem memory analysis of cold-booted android devices. In *IT Security Incident Management & IT Forensics (IMF), 2014 Eighth International Conference on* (pp. 62-75). IEEE.

[Hoque, 2014] Hoque, N., Bhuyan, M. H., Baishya, R. C., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network attacks: Taxonomy, tools and systems. *Journal of Network and Computer Applications*, *40*, 307-324.

[Hurlburt, 2014] Hurlburt, G. F., & Bojanova, I. (2014). Bitcoin: Benefit or curse?. *IT Professional*, *16*(3), 10-15.

[Hussain, 2017] Hussain, S. A., Fatima, M., Saeed, A., Raza, I., & Shahzad, R. K. (2017). Multilevel classification of security concerns in cloud computing. *Applied Computing and Informatics*, *13*(1), 57-65.

[Idika, 2015] Osborn, M. (2015). Malware Detection Techniques. *International Journal of Computer (IJC)*, *18*(1).

[Jayamsakthi, 2008] Jayamsakthi Shanmugam, D. M. (2008). Cross Site Scripting-Latest developments and solutions: A survey. *Int. J. Open Problems Compt. Math*, *1*(2).

[Johari, 2012] Johari, R., & Sharma, P. (2012, May). A survey on web application vulnerabilities (SQLIA, XSS) exploitation and security engine for SQL injection. In *2012 International Conference on Communication Systems and Network Technologies* (pp. 453-458). IEEE.

[Jordan, 2007] Jordan, T. (2007). Online direct action: Hacktivism and radical democracy. In *Radical democracy and the internet* (pp. 73-88). Palgrave Macmillan, London.

[Jouini, 2016] Jouini, M., & Rabai, L. B. A. (2016). A security framework for secure cloud computing environments. *International Journal of Cloud Applications and Computing (IJCAC)*, *6*(3), 32-44.

[Juels, 2016] Juels, A., Kosba, A., & Shi, E. (2016, October). The ring of Gyges: Investigating the future of criminal smart contracts. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 283-295). ACM.

[Justin, 2012] John Justin, M., & Manimurugan, S. (2012). A survey on various encryption techniques. International Journal of Soft Computing and Engineering (IJSCE) ISSN, 2231, 2307.

[Kolias, 2017] Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. *Computer*, *50*(7), 80-84.

[Koshy, 2014] Koshy, P., Koshy, D., & McDaniel, P. (2014, March). An analysis of anonymity in bitcoin using p2p network traffic. In *International Conference on Financial Cryptography and Data Security* (pp. 469-485). Springer, Berlin, Heidelberg.

[Kumar, 2016] Kumar, B., & Al Hasani, M. H. S. (2016, October). Database security—Risks and control methods. In *Computer Communication and the Internet (ICCCI), 2016 IEEE International Conference on* (pp. 334-340). IEEE.

[Landwehr, 1994] Landwehr, C. E., Bull, A. R., McDermott, J. P., & Choi, W. S. (1994). A taxonomy of computer program security flaws. *ACM Computing Surveys (CSUR)*, *26*(3), 211-254.

[Li, 2017] LI, Xiaoqi, et al. A survey on the security of blockchain systems. Future Generation Computer Systems, 2017.

[Lin, 2017] LIN, Iuon-Chang; LIAO, Tzu-Chun. A Survey of Blockchain Security Issues and Challenges. IJ Network Security, 2017, 19.5: 653-659.

[Mansfield-Devine, 2016] Mansfield-Devine, S. (2016). DDoS goes mainstream: how headline-grabbing attacks could make this threat an organisation's biggest nightmare. *Network Security*, *2016*(11), 7-13.

[Mann, 2017] Mann, C., & Loebenberger, D. (2017). Two-factor authentication for the Bitcoin protocol. *International Journal of Information Security*, *16*(2), 213-226.

[Mavroeidis, 2018] Mavroeidis, V., Vishi, K., Zych, M. D., & Jøsang, A. (2018). The Impact of Quantum Computing on Present Cryptography. *arXiv preprint arXiv:1804.00200*.

[McGraw, 2004] McGraw, G. (2004). Software security. *IEEE Security & Privacy*, *2*(2), 80-83.

[Merlo, 2014] Merlo, A., Migliardi, M., & Fontanelli, P. (2014, July). On energy-based profiling of malware in android. In *High Performance Computing & Simulation (HPCS), 2014 International Conference on* (pp. 535-542). IEEE.

[Mirkovic, 2004] Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*, *34*(2), 39-53.

[Mitali, 2014] Mitali, V. K., & Sharma, A. (2014). A survey on various cryptography techniques. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 3(4), 307-312.

[Moon, 2007] Moon, S. K., & Kawitkar, R. S. (2007, December). Data security using data hiding. In *Conference on computational intelligence and multimedia applications, 2007. International conference on*(Vol. 4, pp. 247-251). IEEE.

[Moore, 2009] Moore, T., Clayton, R., & Anderson, R. (2009). The economics of online crime. *Journal of Economic Perspectives*, *23*(3), 3-20.

[Nagle, 2014] Nagle, F. (2014). Crowdsourced digital goods and firm productivity: evidence from free and open source software. Harvard Business School.

[Nakhila, 2015] Nakhila, O., Attiah, A., Jinz, Y., & Zoux, C. (2015, October). Parallel active dictionary attack on wpa2-psk wi-fi networks. In *Military Communications Conference, MILCOM 2015-2015 IEEE*(pp. 665-670). IEEE.

[Namratha, 2016] Namratha, P. (2016). A Novel Method for Provenance Forgery and Packet Drop Attacks in Wireless Sensor Networks. *International Journal of Research*, *3*(14), 1694-1698.

[Natoli, 2017] Natoli, C., & Gramoli, V. (2017, June). The balance attack or why forkable blockchains are ill-suited for consortium. In *Dependable Systems and Networks (DSN), 2017 47th Annual IEEE/IFIP International Conference on* (pp. 579-590). IEEE.

[Nicholson, 2012] Nicholson, A., Webber, S., Dyer, S., Patel, T., & Janicke, H. (2012). SCADA security in the light of Cyber-Warfare. *Computers & Security*, *31*(4), 418-436.

[Ogburn, 2013] Ogburn, M., Turner, C., & Dahal, P. (2013). Homomorphic encryption. *Procedia Computer Science*, *20*, 502-509.

[Okman, 2011] Okman, L., Gal-Oz, N., Gonen, Y., Gudes, E., & Abramov, J. (2011, November). Security issues in nosql databases. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on* (pp. 541-547). IEEE.

[Oktaviana, 2016] Oktaviana, B., & Siahaan, A. P. U. (2016). Three-Pass Protocol Implementation in Caesar Cipher Classic Cryptography. *IOSR Journal of Computer Engineering (IOSR-JCE)*, *18*(4), 26-29.

[Ou, 2005] Ou, X., Govindavajhala, S., & Appel, A. W. (2005, July). MulVAL: A Logic-based Network Security Analyzer. In *USENIX Security Symposium* (Vol. 8).

[Panda, 1999] Panda, B., & Giordano, J. (1999). Reconstructing the database after electronic attacks. In *Database Security XII* (pp. 143-156). Springer, Boston, MA.

[Pandey, 2015] Pandey, R. M., & Verma, V. K. (2015). Data Security using Various Cryptography Techniques: A recent Survey.

[Pandya, 2015] Pandya, D., Narayan, K. R., Thakkar, S., Madhekar, T., & Thakare, B. S. (2015). Brief History of Encryption. *International Journal of Computer Applications*, *131*(9), 28-31.

[Park, 2017] PARK, Jin Ho; PARK, Jong Hyuk. Blockchain security in cloud computing: Use cases, challenges, and solutions. *Symmetry*, 2017, 9.8: 164.

[Parker, 2012] Parker, D. B. (2012). Toward a New Framework for Information Security?. *Computer security handbook*, 3-1.

[Paryasto, 2014] Paryasto, M., Alamsyah, A., & Rahardjo, B. (2014, May). Big-data security management issues. In *Information and Communication Technology (ICoICT), 2014 2nd International Conference on* (pp. 59-63). IEEE.

[Pawar, 2015] Pawar, R. G. (2015). SQL Injection Attacks. *KHOJ: Journal of Indian Management Research and Practices*, 125-129.

[Perlner, 2009] Perlner, R. A., & Cooper, D. A. (2009, April). Quantum resistant public key cryptography: a survey. In *Proceedings of the 8th Symposium on Identity and Trust on the Internet* (pp. 85-93). ACM.

[Petsas, 2014] Petsas, T., Voyatzis, G., Athanasopoulos, E., Polychronakis, M., & Ioannidis, S. (2014, April). Rage against the virtual machine: hindering dynamic analysis of android malware. In *Proceedings of the Seventh European Workshop on System Security* (p. 5). ACM.

[Pries, 2008] Pries, R., Yu, W., Fu, X., & Zhao, W. (2008, May). A new replay attack against anonymous communication networks. In *Communications, 2008. ICC'08. IEEE International Conference on* (pp. 1578-1582). IEEE.

[Prout, 2018] Prout, A., Arcand, W., Bestor, D., Bergeron, B., Byun, C., Gadepally, V., Michaleas, P. et al. (2018). Measuring the Impact of Spectre and Meltdown. *arXiv preprint arXiv:1807.08703*.

[Rao, 2015] Rao, R. V., & Selvamani, K. (2015). Data security challenges and its solutions in cloud computing. *Procedia Computer Science*, *48*, 204-209.

[Ron, 2015] Ron, A., Shulman-Peleg, A., & Bronshtein, E. (2015). No SQL, no Injection?. In *IEEE S&P workshop on Web* (Vol. 2).

[Ryba, 2015] Ryba, F. J., Orlinski, M., Wählisch, M., Rossow, C., & Schmidt, T. C. (2015). Amplification and DRDoS Attack Defense--A Survey and New Perspectives. *arXiv preprint arXiv:1505.07892*.

[Sadeghi, 2015] Sadeghi, A., Bagheri, H., & Malek, S. (2015, May). Analysis of android inter-app security vulnerabilities using COVERT. In *Proceedings of the 37th International Conference on Software Engineering-Volume 2* (pp. 725-728). IEEE Press.

[Scott, 2002] Scott, D., & Sharp, R. (2002, May). Abstracting application-level web security. In *Proceedings of the 11th international conference on World Wide Web* (pp. 396-407). ACM.

[Scott, 2003] Scott, D., & Sharp, R. (2003). Specifying and enforcing application-level web security policies. *IEEE Transactions on Knowledge & Data Engineering*, (4), 771-783.

[Shan, 2012] Shan, Z., Wang, X., & Chiueh, T. C. (2012). Enforcing mandatory access control in commodity OS to disable malware. *IEEE Transactions on Dependable and Secure Computing*, *9*(4), 541-555.

[Sheridan, 2015] Sheridan, S., & Keane, A. (2015). Detection of DNS-Based Covert Channel Beacon Signals. *Journal of Information Warfare*, *14*(4), 98-IV.

[Singh, 2013] Singh, G. (2013). A study of encryption algorithms (RSA, DES, 3DES and AES) for information security. International Journal of Computer Applications, 67(19).

[Singh, 2014] Singh, M. M., Siang, S. S., San, O. Y., Hashimah, N., Malim, A. H., & Shariff, A. R. M. (2014). Security Attacks Taxonomy on Bring Your Own Devices (BYOD) Model. *International Journal of Mobile Network Communications & Telematics (IJMNCT) Vol*, *4*.

[Singh, 2015] Singh, P., & Kaur, K. (2015, February). Database security using encryption. In Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015 International Conference on (pp. 353-358). IEEE.

[Smart, 2010] Smart, N. P., & Vercauteren, F. (2010, May). Fully homomorphic encryption with relatively small key and ciphertext sizes. In *International Workshop on Public Key Cryptography* (pp. 420-443). Springer, Berlin, Heidelberg.

[Spreitzenbarth, 2015] Spreitzenbarth, M., Schreck, T., Echtler, F., Arp, D., & Hoffmann, J. (2015). Mobile-Sandbox: combining static and dynamic analysis with machine-learning techniques. *International Journal of Information Security*, *14*(2), 141-153.

[Spyridopoulos, 2013] Spyridopoulos, T., Tryfonas, T., & May, J. (2013). Incident analysis & digital forensics in SCADA and industrial control systems.

[Srinivasan, 2017] Srinivasan, S. M., & Sangwan, R. S. (2017). Web App Security: A Comparison and Categorization of Testing Frameworks. *IEEE Software*, (1), 99-102.

[Stahlberg, 2007] Stahlberg, P., Miklau, G., & Levine, B. N. (2007, June). Threats to privacy in the forensic analysis of database systems. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data* (pp. 91-102). ACM.

[Stohl, 2006] Stohl, M. (2006). Cyber terrorism: a clear and present danger, the sum of all fears, breaking point or patriot games?. *Crime, law and social change*, *46*(4-5), 223-238.

[Thambiraja, 2012] Thambiraja, E., Ramesh, G., & Umarani, D. R. (2012). A survey on various most common encryption techniques. International journal of advanced research in computer science and software engineering, 2(7).

[Thanh, 2015] Thanh, T. Q., Covaci, S., Ertl, B., & Zampognano, P. (2015, June). An Integrated Access Control Service Enabler for Cloud Applications. In *International Conference on Future Network Systems and Security* (pp. 101-112). Springer, Cham.

[Thota, 2018] Thota, C., Sundarasekar, R., Manogaran, G., Varatharajan, R., & Priyan, M. K. (2018). Centralized fog computing security platform for IoT and cloud in healthcare system. In *Exploring the convergence of big data and the internet of things* (pp. 141-154). IGI Global.

[Tian, 2014] Tian, X., Huang, B., & Wu, M. (2014, May). A transparent middleware for encrypting data in MongoDB. In *Electronics, Computer and Applications, 2014 IEEE Workshop on* (pp. 906-909). IEEE.

[Tidwell, 2001] Tidwell, T., Larson, R., Fitch, K., & Hale, J. (2001, June). Modeling internet attacks. In *Proceedings of the 2001 IEEE Workshop on Information Assurance and security* (Vol. 59). United States Military Academy West Point, NY.

[Tiwari, 2016] Tiwari, P. K., & Joshi, S. (2016). Data security for software as a service. In *Web-Based Services: Concepts, Methodologies, Tools, and Applications* (pp. 864-880). IGI Global.

[Tofan, 2011] Tofan, D. C. (2011). Information security standards. *Journal of Mobile, Embedded and Distributed Systems*, *3*(3), 128-135.

[Tripathi, 2014] Tripathi, R., & Agrawal, S. (2014). Comparative study of symmetric and asymmetric cryptography techniques. International Journal of Advance Foundation and Research in Computer (IJAFRC), 1(6), 68-76.

[Valiron, 2015] Valiron, B., Ross, N. J., Selinger, P., Alexander, D. S., & Smith, J. M. (2015). Programming the quantum future. *Communications of the ACM*, *58*(8), 52-61.

[Verma, 2015] Verma, N., & Kaur, A. (2015). A Detailed Study on Prevention of SQLI attacks for Web Security. *International Journal of Computer Applications Technology and Research*, *4*(4), 308-311.

[Visas, 2014] Vidas, T., & Christin, N. (2014, June). Evading android runtime analysis via sandbox detection. In *Proceedings of the 9th ACM symposium on Information, computer and communications security* (pp. 447-458). ACM.

[Vogt, 2007] Vogt, P., Nentwich, F., Jovanovic, N., Kirda, E., Kruegel, C., & Vigna, G. (2007, February). Cross Site Scripting Prevention with Dynamic Data Tainting and Static Analysis. In *NDSS* (Vol. 2007, p. 12).

[Vorobiev, 2010] Vorobiev, V. I., Fedorchenko, L. N., Zabolotsky, V. P., & Lyubimov, A. V. (2010, September). Ontology-based analysis of information security standards and capabilities for their harmonization. In *Proceedings of the 3rd international conference on Security of information and networks* (pp. 137-141). ACM.

[Weaver, 2003] Weaver, N., Paxson, V., Staniford, S., & Cunningham, R. (2003, October). A taxonomy of computer worms. In *Proceedings of the 2003 ACM workshop on Rapid malcode* (pp. 11-18). ACM.

[Welch, 2003] Welch, D., & Lathrop, S. (2003, June). Wireless security threat taxonomy. In *Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society* (pp. 76-83). IEEE.

[Wood, 2016] Wood, S. (2016). *U.S. Patent No. 9,379,912*. Washington, DC: U.S. Patent and Trademark Office.

[Wu, 2006] Wu, M., Miller, R. C., & Garfinkel, S. L. (2006, April). Do security toolbars actually prevent phishing attacks?. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*(pp. 601-610). ACM.

[Wu, 2006b] Wu, M., Miller, R. C., & Little, G. (2006, July). Web wallet: preventing phishing attacks by revealing user intentions. In *Proceedings of the second symposium on Usable privacy and security* (pp. 102-113). ACM.

[You, 2014] You, Y. I., & Lee, K. H. (2014). Study on accurate calculating the risk of the SCADA. *International Journal of Security and its Applications*, *8*(1), 225-230.

[Zhang, 2006] Zhang, Y., Yang, B., & Niu, X. M. (2006). Reversible watermarking for relational database authentication. *Journal of Computers*, *17*(2), 59-66.

[Zhu, 2012] Zhu, Q., Yang, X., & Ren, J. (2012). Modeling and analysis of the spread of computer virus. *Communications in Nonlinear Science and Numerical Simulation*, *17*(12), 5117-5124.

[Zuo, 2004] Zuo, Y., & Panda, B. (2004). Damage discovery in distributed database systems. In *Research directions in data and applications security XVIII* (pp. 111-123). Springer, Boston, MA.