electronics

MDPI

Article

Security Information Sharing in Smart Grids: Persisting Security Audits to the Blockchain

Andrés Marín-López [1,*,†], Sergio Chica-Manjarrez [1], David Arroyo [2], Florina Almenares-Mendoza [1] and Daniel Díaz-Sánchez [1]

[1] Telematics Engineering Department, Politechnical Engineering School, University Carlos III de Madrid, Avda. de la Universidad, 20, 28911 Leganés, Madrid, Spain; sergio.chica@uc3m.es (S.C.-M.); florina.almenares@uc3m.es (F.A.-M.); daniel.diaz@uc3m.es (D.D.-S.)

[2] Institute of Physical and Information Technologies (ITEFI), Spanish National Research Council (CSIC), C/Serrano, 144, 28006 Madrid, Spain; david.arroyo@csic.es

* Correspondence: andres.marin@uc3m.es

† Current address: KIT Campus South, Building 50.34, Am Fasanengarten 5, 76131 Karlsruhe, Germany.

Received: 15 September 2020; Accepted: 4 November 2020; Published: 6 November 2020

Abstract: With the transformation in smart grids, power grid companies are becoming increasingly dependent on data networks. Data networks are used to transport information and commands for optimizing power grid operations: Planning, generation, transportation, and distribution. Performing periodic security audits is one of the required tasks for securing networks, and we proposed in a previous work AUTOAUDITOR, a system to achieve automatic auditing. It was designed according to the specific requirements of power grid companies, such as scaling with the huge number of heterogeneous equipment in power grid companies. Though pentesting and security audits are required for continuous monitoring, collaboration is of utmost importance to fight cyber threats. In this paper we work on the accountability of audit results and explore how the list of audit result records can be included in a blockchain, since blockchains are by design resistant to data modification. Moreover, blockchains endowed with smart contracts functionality boost the automation of both digital evidence gathering, audit, and controlled information exchange. To our knowledge, no such system exists. We perform throughput evaluation to assess the feasibility of the system and show that the system is viable for adaptation to the inventory systems of electrical companies.

Keywords: security auditing; permissioned blockchain; scalability; smart grid security

1. Introduction

A smart grid is an energy delivery system that moves from a centrally controlled system, like the ones we currently have, to a consumer-driven approach, i.e., an iterative system relying on bi-directional communication to adapt and tune the delivery of energy in the real-time market. A smart grid includes a broad range of sophisticated sensors that constantly assess the state of the grid and the electrical power demand and availability, with the aim of optimizing the energy supply. The power grid is evolving into a cyber physical system where smart devices allow advanced monitoring and control. This area is developing fast, as one can guess by seeing the 68 active working groups of the IEEE SA around smart grids. More than half of the EU Member States have reached a 10% installation rate for electricity smart meters, showing the first important step in their large-scale rollout programs. Seven have already reached 80% like Denmark, or even finished their large-scale electricity smart metering roll-out like Estonia (>98% in 2017), Finland (100% by 2013), Italy (95% by 2011), Malta (80–85% by 2014), Spain (100% end of 2018), and Sweden (100% by 2009). Some of them are already proceeding with the second generation rollout, like Italy, or are planning to [1].

Other devices are also expected to be incorporated into the so-called smart grid, such as domestic power micro-generators, smart protection and storage devices, and even electric vehicles. Electrical data provided by such devices will be collected not only by customer applications but also by other stakeholders, most notably power distribution and power transportation companies. Besides enhanced control, such data will allow new intelligent features such as self-healing, resilience, sustainability, and will improve the efficiency of energy critical infrastructure. Near-real time communication between the power grid subsystems (from the concentrators to the legacy systems) with smart sensors and devices will be required to achieve optimization, automation and control of the smart grid. These new features will bring obvious benefits for electrical companies, but also customers are expected to benefit from smart grids. Specifically, from smart meters they can have three important benefits: (1) A better customer service comes from (a) fewer and shorter duration of outages as smart meters report instantaneously and (b) from a faster service (remote operations); (2) viewing energy usage in near real-time and comparing with current tariffs, which can lead to adjusted monthly bills; and (3) manual periodic metering no longer being required.

Besides the mentioned benefits of smart grids, the risks are obvious if the system fails. In this work we address the protection of smart grids as critical infrastructures against cyber attacks, and we specifically address the security auditing of devices and networks that comprise smart networks. The security auditing we are referring to in this article is also known as pentesting or vulnerability testing.

Critical infrastructures require automated security control assessment, as described in NISTIR-8011 Volume 3 and 4 [2,3]. This can be achieved by the integration of auditing with the inventory system, so as to have a realistic view of the company assets together with their associated risks in near real time. We foresee this integration as automatic or semi-automatic auditing procedures fired from the inventory system. The auditing processes incorporate security metadata to the inventoried assets of the company: Releases and patch status history, known vulnerabilities and their respective severity information. The risk metrics incorporate the severity of exposures, and facilitate the selection of the vulnerabilities that have to be mitigated, according to the risk appetite of the company.

Cybersecurity information sharing and collaboration between organizations can decrease the time of threat detection and increase the accuracy of detection. Several researches have pursued privacy in cybersecurity information [4,5]. We aim to contribute in the accountability of this information sharing.

In [6] we described AUTOAUDITOR, a system for automatic or semi-automatic security auditing to be integrated with power grid companies inventory systems. In this article we address the accountability of the system via the integration of auditing results records in a permissioned blockchain. Certainly, in the energy sector there exist a vast set of requirements and needs than can be fulfilled by an adequate blockchain architecture [7]. In our case, the decentralized nature of the blockchain is very appealing as the backbone of a least privilege strategy along the chain of custody of digital evidences. Furthermore, the tamper-resistant nature of blockchain makes it a very good candidate for protecting the integrity of audit trails. Finally, a permissioned blockchain enables the design of an adequate access control for the overall process of auditing information systems. Indeed, without a proper separation of duties this task cannot be satisfied.

This paper is organized as follows. Section 2 examines related automatic auditing systems, some of them using machine learning, and related works with a different usage of distributed ledgers in energy systems. Section 3 presents the previous version of AUTOAUDITOR [6], and discusses the requirements of an automatic auditing system in the field of smart grids. Section 4 outlines the benefits of introducing blockchains in the system, analyzes and justifies our selection of distributed ledger technology. Section 5 gives an overview of the implementation details, explaining the execution flow, identified roles, and the smart contract implemented. Finally, Section 6 describes performance results and Section 7 formulates our conclusions.

2. Related Work

2.1. Security Assessment. Works in Automatic Auditing

Security assessment is defined as "a circular process of assessing assets for their security requirements, based on probable risks of attack, liability related to successful attacks, and costs for ameliorating the risks and liabilities", according to the standard developed for handling the security of power systems and associated information exchange, IEC TS 62351 [8].

The U.S. National Institute of Standards and Technology (NIST) and the Department of Homeland Security (DHS) are working in defining capabilities of automation support for ongoing security control assessments such as software asset management (SWAM) [2] and software vulnerability management (VULN) [3]. Security control assessment is a crucial part to manage information security risks across a company. Risk management is a complex and multifaceted activity, which starts establishing a realistic "risk frame" where assumptions about threats, vulnerabilities, consequences/impact, and likelihoods are identified, followed by information security assessment and monitoring [9]:

- *Software Asset Management* capability is defined as part of Continuous Diagnostics and Mitigation (CDM) process. Its main purpose is to control risk created by unmanaged or unauthorized software installed on a supervised network. The authorized software installed on every device is inventoried and can be as small as a line of source code or as large as a software suite made up of multiple products, thousands of individual executables, and countless lines of code, e.g., firmware, BIOS, operating systems, applications, services, and malware. Thus, a software asset is usable and automated, being described in terms of Common Platform Enumeration (CPE) names.
  CPE is a standardized method of describing and identifying classes of applications, operating systems, and hardware devices present among an enterprise's computing assets. For that, Software Identification (SWID) tags for identifying software installed is included. A CPE name includes at least four unique attributes: Part, vendor, product, and version, and four additional attributes: Language, sw_edition, target_hw, and update. These second group of attributes are used to identify where software vulnerabilities may be found.
  SWAM supports vulnerability management and configuration settings management. Likewise, it directly supports Hardware Asset Management (HWAM) because checking software asset requires knowing where it was or should be installed.
- *Software Vulnerability Management* addresses defects present in software on the network. Thus, once software and hardware are part of the inventory, VULN capability provides visibility into the vulnerabilities in software authorized to operate or access to the company's network(s), in order to manage and patch them in an appropriate manner. Vulnerable software is software in use on a system that has a vulnerability, but has not yet been patched or otherwise mitigated, being a key target of attackers in order to initiate an attack [3]. VULN manages and assesses directly two kinds of software flaws: Common Vulnerabilities and Exposures (CVEs), whose program works with software providers, vulnerability coordinators, bug bounty programs, and vulnerability researchers to provide a list of publicly disclosed vulnerabilities, and Common Weakness Enumeration (CWE), which provides identifiers for weaknesses that result from poor coding practices and have the potential result in software vulnerabilities.

Researchers, developers, and industry have developed tools for automating these capabilities. The project Software Assurance Marketplace (SWAMP) offers a service to provide continuous software assurance capabilities to researchers and developers. They also offer a self-contained, standalone version of SWAMP [10]. Besides this tool, there are many other available, some of them as part of the Black Hat Arsenal or/and Kali, which try to automate mainly vulnerability assessment such as:

- *DeepExploit* [11] is a fully automated penetration test tool linked with Metasploit. It identifies the status of all opened ports on the target server and executes an automated attack. Among the execution possibilities, it may be launched in a self-learning mode (using reinforcement learning);

- *VAPT framework* [12] is an automated Vulnerability Assessment and Penetration Testing tool that identifies vulnerabilities, retrieves exploits from open databases, e.g., ExploitDB, and performs penetration tests. The results are stored in graph-based database, Neo4j, at each stage;
- *APT2* [13] is a console-based Automated Penetration Testing Toolkit, whose results are stored locally and used to launch exploits and enumeration modules. This performs a nmap scan or import the results of other scanners;
- *Archery* [14] uses open source web and network vulnerability scanners (e.g., zap, nmap, openvas, selenium, etc.) to create a vulnerability assessment and management tool;
- *Lynis* [15] is a shell-script that runs on *NIX-based operating systems. It is an extensible security audit and vulnerability analysis tool, which performs security tests to check configuration errors, software vulnerabilities, or weaknesses, in order to perform vulnerability assessments and penetration tests;
- *CROZONO framework* [16,17] allows gathering information about possible attack vectors and performing automated penetration tests from autonomous devices (e.g., drones, robots, etc.) that could ease the access to the logical infrastructure of an industrial facility [16]. This framework has a key feature because it generates reports about gathered information identifying weak points and exposure levels;
- *Faraday platform* [18] reuses the available tools in the community to perform penetration-tests. This introduces the concept of Integrated Penetration-Test Environment (IPE), which automates distribution, indexation, and analysis of the data generated during a security audit;
- *Intrigue Core* [19] discovers assets (i.e., applications and infrastructure) and vulnerabilities utilizing APIs and OSINT techniques to discover an attack surface. This can be used from a docker image or web interface;
- *Leviathan framework* [20] is a python-based audit toolkit which includes service discovery (using Shodan and Censys), brute force, SQL injection detection, and running custom exploit capabilities. This tool allows to do massive scans (using masscan) on several systems at once;
- *Trommel* [21] is a python tool that sifts embedded device files to identify potential vulnerabilities, such as, protocol key files, email addresses, shell scripts, etc. It integrates vFeed for in-depth vulnerability analysis.

A comparison can be found in Table 1, where `X` indicates that functionality is satisfied and—means the opposite. We can see that none of the tools support all the analyzed characteristics. In particular, none are designed for exchanging or sharing logs with security information. Likewise, there is no a ledger that allows tracking risks over time. As mentioned before, these tools have been designed to assess vulnerabilities, which is the first step in the life cycle of vulnerabilities management, but these do not implement or facilitate the remaining steps. Our approach supports the smart grid companies to design, perform, and integrate their tests in their continuous auditing processes, and the use of secure procedures to grant access to security logs for the sake of advancing security countermeasures. This is the base to report, remedy, and verify such threats. Besides we aim at using containers, library objects, and well known components in pentesting, and to use common network infrastructure to provide autonomy to the auditing companies.

A practical risk assessment method applied to Austrian smart grid is presented [22]. The method follows a twofold approach: A conceptual and implementation-based assessment. For the conceptual assessment it uses a reference architecture based on the Smart Grid Architecture Model (SGAM) [23], in order to analyze the deployed architectures or, to be deployed in the near future, for mapping them to SGAM. The outcome of this first phase is a risk matrix and mitigation strategies. Then, the implementation-based assessment consists in evaluating details of systems, such as poor configurations and potential software implementation vulnerabilities. This second phase deals with existing systems that allow a security audit to assess the security with respect to the potential attack vectors and vulnerabilities, resulting in a set of possible exploits.

Table 1. Comparison of tools to automate software asset management (SWAM) and/or vulnerability management (VULN).

	Asset Mgmnt	VULN	Pen. Tests	Shared Logs	Full Automation
DeepExploit	-	-	X	-	-
VAPT	X	X	X	-	-
APT2	-	-	X	-	-
Archery	-	X	X	-	-
Lynis	X	X	X	-	X (docker)
CROZONO	-	-	X	-	X (dron & robots)
Faraday	-	-	X	-	-
Intrigue Core	X	X	-	-	X
Leviathan	X	-	X	-	X
Trommel	-	X	-	-	-

Though in this paper we restrict to network and computer security audits, recent works related to a broader concept of auditing, and the role of Supreme Audit Institutions, explore the dependency between auditing and trust [24]. Lack of trust in public organizations can contribute to the necessity of frequent auditing. Audits can enable auditors to correct errors and irregularities, strengthening the audited entities, and thus enhancing public trust. In [24], conclusions include that enforcing the informative function of audit institutions can strengthen SAI's trustworthiness for their customers. Blockchain-based architectures can be conceived not as a replacement of auditors but to achieve a less expensive and more effective auditing of information systems [25].

2.2. Distributed Ledger Technologies in Smart Grids

Researchers have proposed some solutions focused on smart grids and IoT (Internet of Things). In [26], an ISO/IEC 15408-2 compliant security auditing system based on a blockchain network as the underlying communication architecture is proposed for IoT. Certainly, the inner characteristics of blockchains and, in general, Distributed Ledger Technologies (DLT) paves the way to construct transparent and traceable procedures of major interest for the energy sector [7]. The heterogeneity of this ecosystem makes cumbersome to deploy management solutions and governance schemes to ponder security, reliability, but also regulatory compliance. Although blockchain was initially interpreted as channel to guide arising functional needs in the energy sector (e.g., P2P energy trading), there exists an increasing trend to handle cybersecurity and cyber safety goals in this ecosystem by means of blockchain procedures. Continuous cybersecurity management can be perfected with the guidance of blockchain logics [27], governance, and interoperability matters can be articulated in a more nuanced way with the assistance of on-chain and off-chain blockchain protocols. In this sense, it is worth noting current efforts in organizations as the International Association of Trusted Blockchain [28] or the European Telecommunications Standards Institute (ETSI) [29] to align efforts in pursuing standard data models and procedures for a broad set of application contexts, including the energy sector. Regarding incident management, standardization is crucial to ease the sharing, traceability, and trust evaluation of cyber threat intelligence sources [30]. Provenance evaluation is also critical for the identification of threats associated with inadequate software and firmware updates in the ecosystems of smart and micro grids. Blockchain could be used to leverage the root of trust of physical devices and get genuine information about critical security updates across the endpoints of the ecosystem [31].

There are other researches in the interconnection strategy of federated smart grids (power networks, control systems, market, customer premises). Ref. [32] proposes a three layer interconnection architecture, being L3 the distributed ledger layer which handles transactions of type `<object><resource><action>`. L3 acts thus as a distributed database across all partners of the federation. The paper argues that: "the own nature of blockchain in L3 already addresses itself"

(the need to manage provenance measures for traceability). As for auditors, denoted as SECAUD in IEC-62351-8, they are in charge of verifying the performance of the infrastructure, even ensuring the correct application of the authorization policies with the verification of access registers (stored as transactions in L3).

This work is a step towards the deployment of an auditing system aligned with [32]. We address to improve the interoperability and collaboration between the agents involved in forensic research of attacks and failures in smart grids, potentially federated.

3. AUTOAUDITOR System Description

Power grid companies hire third parties for performing security analysis. Those third-party companies have to perform security audits of the critical infrastructures deployed. Our proposal was to design a system that can be controlled by the power grid companies automatically, using a tailored and preconfigured system provided by the security company. This system has two main benefits: (1) The security company is not required to do the bulk work of auditing the whole installation, and (2) the power grid company does not have to grant access to the critical infrastructures for periodic auditing. AUTOAUDITOR [6] is designed to test elements of different types: Smart meters, smart meter concentrators, other smart grid equipment (power chargers, etc.), networking elements (switches, routers, proxies, gateways), and servers in the core of the company's network(s).

In AUTOAUDITOR, the security company role is to do an initial fingerprinting of a sample device or network equipment to test followed by the identification of potential vulnerabilities to test and the selection of the most suited auditing modules. The fingerprinting process output can be used by the company to update the company inventory. This will bring two additional benefits: Increase in the accuracy of the fingerprinting process and improve the automatic inventory with an additional online source. Inaccuracies in the automatic fingerprinting may be detected manually or in the reconciliation with the inventory system. This procedure can help the company to confirm the inventory with respect to production elements at different points of their network. For instance, imagine we test a smart meter X and end with a fingerprinting test procedure. The company can run this test to confirm that a given population of smart meters of type X, according to the inventory are indeed of type X, and have not been replaced by another element type.

Zero-day vulnerabilities and exploits are out of the scope of AUTOAUDITOR. The proposed approach towards zero-day(s) is to have the devices subject to continuous monitoring processes and further inspected by behavior analysis. Such a behavior analysis may trigger alerts and subsequent manual inspection, or even subject to preventive block or quarantine. We expect also to improve the defenses through the security information sharing and collaboration with other companies.

At this step AUTOAUDITOR offers the possibility of packing the collected information in an attack plan persisted to a JSON file. The attack plan is encapsulated in a container. The reason why we use containers is to better scale with the number of potential devices which will be delivered to the power company. It is up to the power company: (1) To configure the addresses of the devices to test, and (2) to decide upon the resources devoted for the testing. The result is the test plan which can include parameters, either manually defined or template based. Those parameters are useful for defining configurable connections including different network parameters for running the different tests. AUTOAUDITOR presently uses VPN connections, though other network connection configurations can be added. The main reason for this is that the equipment of a power grid operator is scattered along the geography of the region or country supplied by the operator. The operator internal routers and firewalls ensure the separation of the interconnecting networks. The system must include the capability of defining configurable connections including different network parameters for running different tests.

Finally, the encapsulated test plan can be manually or automatically executed, according to the inventory system needs to update the data regarding the included assets and their related security information (vulnerability meta-information). The test will fire the instantiation and execution

of the tests. That may require setting up connection elements according to the defined network configuration and the concrete parameters of the test. The execution of the system will output evidences of the vulnerabilities tested. AUTOAUDITOR outputs the tested CVEs, whether successful or not, and other information obtained in the testing. A link to a supporting video is included as Supplementary Material.

This information is part of the security assessment to better evaluate the vulnerabilities and incorporate the success likelihood for the tested attacks. This is the information we propose to improve its accountability, and to persist and share with other actors involved in the smart grid. The next section justifies the selection of distributed ledger technologies for such purposes.

4. Evolving AUTOAUDITOR with Blockchains

AUTOAUDITOR was designed with the requirements identified in the previous section for the protection of critical infrastructures such as power grids. In a first iteration, AUTOAUDITOR was conceived to automate security audits in power grids [6]. In this second iteration, AUTOAUDITOR is extended by integrating a blockchain protocol to enable collaboration in cyber threat intelligence. By forcing a strict AAAA (Access, Authorization, Audit, Accountability) policy on the basis of a blockchain, a set of functions will be implemented using smart contracts. These functions are targeted at persisting security trails on a blockchain and to provide access to such evidences according to a concrete policy. As an append-only log, and taking into account the tamper-proof nature of blockchain, AUTOAUDITOR is thus enhanced to foster accountability along the life cycle of the continuous security auditing of power grids.

Among the different types of blockchain, the AAAA goal demands functionality to restrict the set of users/entities with permissions to write in the blockchain, and to establish the information to share with other users/entities according to specific agreements and objectives in the construction of collaborative consortia in the sphere of cyber threat intelligence. In other words, audit information is not going to be publicly accessible and access is granted on the basis of explicit agreements among the concerned parties. Therefore, we are going to adopt a permissioned blockchain to extend AUTOAUDITOR in order to boost the automation of audits in power grids, and to establish accountable channels for sharing outcomes and insights in concrete investigations about security and safety incidents in this domain.

HyperLedger Fabric (HLF) provides one of the most adopted permissioned blockchains. As part of the Hyperledger initiative, HLF entails functionality to create and manage a Public Key Infrastructure by means of a so-called Membership Service Provider (MSP). The architecture of HLF is deployed through a meaningful set of nodes, which can have different roles according to their implication in the underlying execute-order-validate protocol [33]. This separation of duties is very relevant in a context of collaborative frameworks where information disclosure must be conducted according to data minimization criteria. Moreover, the throughput, latency, and scalability of HLF is adequate for the application context of AUTOAUDITOR (https://www.hyperledger.org/learn/publications/blockchain-performance-metrics [last accessed on 13 September 2020]). Finally, HLF comes with a mature Software Development Kit that paves the way for prototyping. All in all, HLF has been adopted in AUTOAUDITOR to extend the first version of the tool and to deploy a platform to favor information exchange among the different agents and entities involved in the investigation of security threats and incidents in power grids.

5. Implementation Details

5.1. Scenario and General Workflow

The scenario we address is a failure or an attack, where the affected company can help the forensic investigation with the details of audit results prior to the incident, i.e. which security audits have been performed in the organization including all the details about devices and vulnerabilities tested.

This can narrow the investigation and reveal sooner the details of the attack, so that the systems can be easily protected from similar attacks. However the most important gain of this accountability is obtained from sharing some of the audit results with other power grid companies and relevant players, therefore, learning from the attacks on other companies so as to be better protected.

AUTOAUDITOR workflow is illustrated in Figure 1. It starts with an auxiliary tool CVEScanner [34] and the list of metadata about potential vulnerabilities in the database, including available metasploit modules. The list is saved in the JSON format and the expert supervises and configures the attack plan, which is then encapsulated for execution. The configuration also comprises the VPN connection including addresses and credentials. AUTOAUDITOR outputs the composition of the containers including the connection and audit application. The composition is then manually instantiated the first time so that the expert can verify its correctness. After potential corrections are made, the company launches the automatic deployment and execution of as many instances as required for the equipment under test (EUT). Finally, the results are collected, processed, and stored in the blockchain.

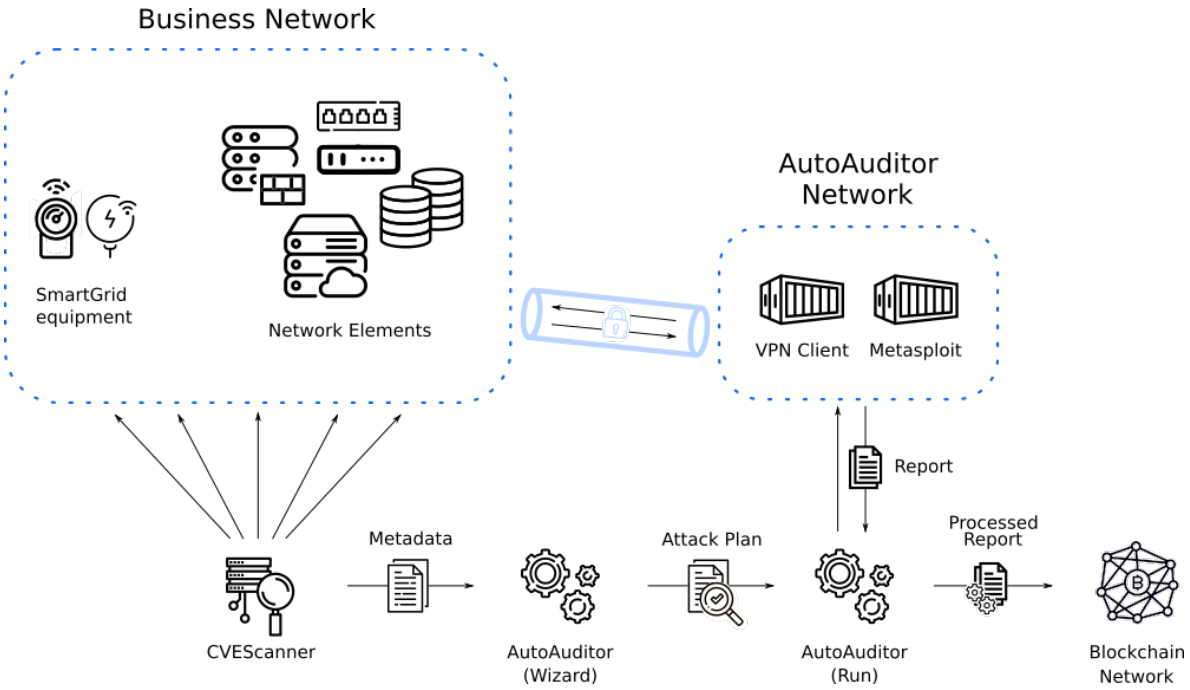

Figure 1. AUTOAUDITOR workflow.

5.2. Auditing Objects and Roles

As a first step in the implementation of the blockchain-based AAAA system upon AUTOAUDITOR, sensitive information has to be protected by defining and implementing a suitable access control policy [29]. We have identified the following information objects that are subject to access control:

- Network topology information including concrete network addresses;
- Fingerprint of the equipment under test (EUT):
  - Hardware architecture;
  - Firmware manufacturer and version;
  - Operating system version;
  - Network address;
  - Identified subcomponents.
- Details on the auditing tests execution results:

- Vulnerabilities tested and their severity;
- Test timestamps;
- Used modules;
- Successful modules;
- Hosts proved vulnerable;
- Information retrieved from the attack (possibly including credentials).

On the ground of the previous classification, we have identified three different roles for access control to AUTOAUDITOR audit results:

- Role A: Total access to the whole records. This access corresponds to people in charge of security as the CSO or CISO, and similar staff of the power grid company. It will also be granted to forensic investigators in case of attack;
- Role B: Partial access to records. This access is envisaged for Critical Infrastructure regional or national managers. They require some details of the audits to have a clear picture of the level of risk accepted by the companies, typically including last audit timestamp, periodicity of audits, vulnerabilities tested, and general information on the equipment (number of devices). Network topology, network addresses and some details of the devices are hidden;
- Role C: Limited access to records. Power grid companies should share information with this level of access. Some of the quantitative information granted for the B level like the number of devices will be hidden, but qualitative information used and successful modules will be available.

5.3. Smart Contract and Distributed Ledger Workflow

AUTOAUDITOR implements storage of reports in a HLF network. The version of HLF of this prototype is 2.1.1. We have developed a smart contract (chaincode as coined in HLF) compatible with AUTOAUDITOR, allowing audits reports storage and query by some given identities. This different access is programmed by defining two different collections, which makes possible to protect private data and to keep the secrecy of specific types of audit reports. In the blockchain-based AAAA system of AUTOAUDITOR, these two collections have been defined:

- Collection A: Stores very basic information, i.e. year and month of report and number of machines affected per vulnerability;
- Collection B: Stores more detailed information, i.e. accurate timestamp of report, metasploit modules tested, and network address of affected machines.

In addition, each collection stores common information, namely, number of vulnerabilities, tested vulnerabilities, and vulnerability score.

Developed smart contract exposes multiple functions for storage, querying, and deletion:

- **NewReport():** Main functionality of the smart contract. Enables AUTOAUDITOR to store reports in the blockchain. Makes use of transient map in order to not track input data in a transaction record;
- **GetReportById():** To query a report by a given unique identifier. Returns a report in JSON format;
- **GetReportsByOrganization():** To query all reports from a given organization. Returns a list of reports in JSON.
- **GetReportsByDate():** to query all reports from a given year and month. Returns a list of reports in JSON;
- **GetReportHash():** Allows data integrity check. Returns data hash;
- **DeleteReport():** Sets the status of the register as deleted so that it does not appear in queries. No evidence is ever deleted from the blockchain.

A general overview of the HLF workflow is presented in Figure 2.

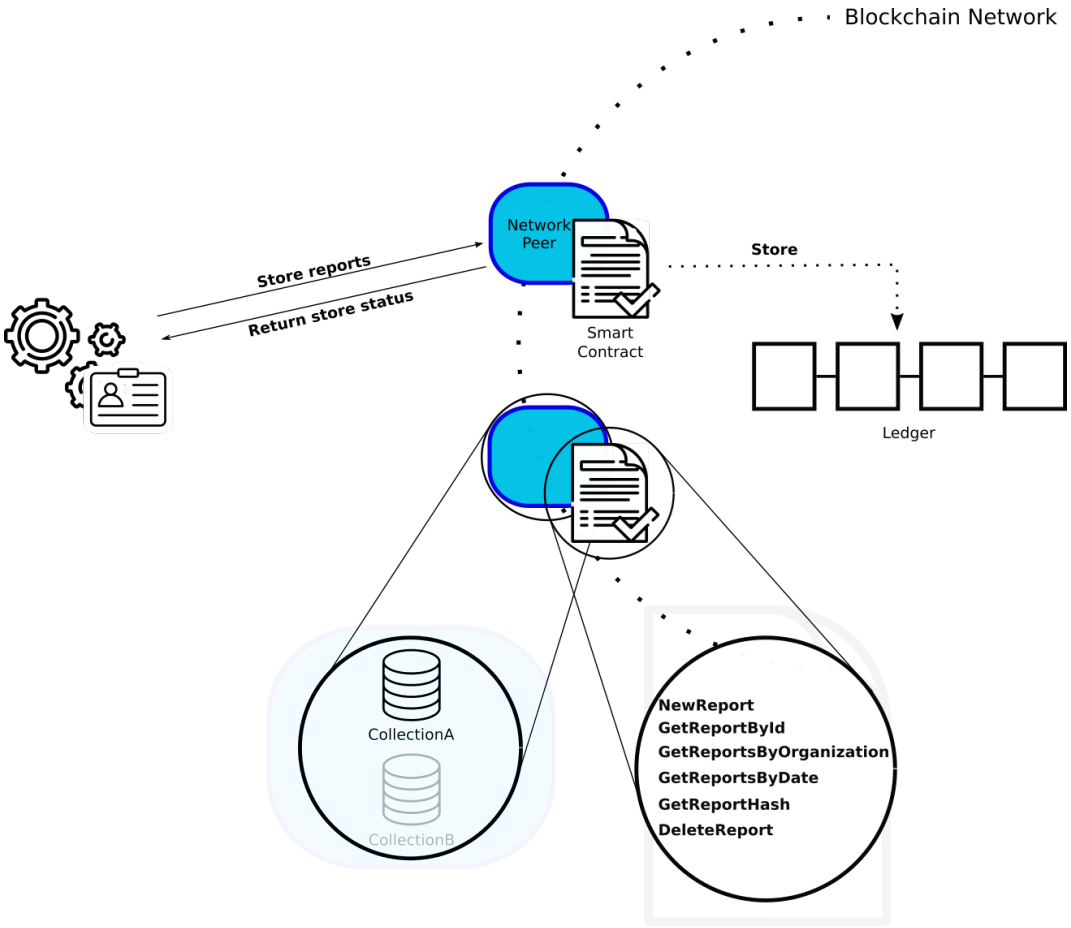

Figure 2. Hyperledger Fabric workflow.

6. Results

To test the correct behavior of the system, we have developed a closed environment including the EUTs, VPN server, the application performing the audit, and the HLF network. It is orchestrated as a python application that sets up the configuration and composition, instantiates and connects the components, and launches the auditing. In addition, it collects the results, and after the analysis, uploads the relevant information to the blockchain. Such results should be similar to the ones obtained in a company, without the time required to instantiate the EUTs, which have been modeled as a set of vulnerable containers.

The auditing process analyzes a list of 10 vulnerabilities. The list contains different vulnerabilities of software products that can be used in clientes, servers, IoT devices, and other devices in the smart grid. They are just examples of vulnerabilities with publicly available exploitation modules we have selected: CVE-2010-2961 correspond to a privilege escalation in Unix filesystems; CVE-2012-2122 is an authentication bypass in mySQL; CVE-2014-0160 is an OpenSSL implementation bug which leaks data, even the server private key; CVE-2014-6271 is a bash bug that allows attackers to execute arbitrary commands; CVE-2017-5638 allows attackers to execute remote commands in Apache Struts; CVE-2017-11610 can be used to issue remote commands by XML-RPC; CVE-2017-12635 is a Apache couchDB vulnerability which allows attackers to execute commands without being authenticated; CVE-2018-10933 exploits may lead to unauthorized OpenSSH server access; CVE-2018-15473 allows attackers to enumerate the users of a OpenSSH server; and CVE-2019-5418 may disclose files in RubyOnRails.

The test environment comprises a HLF network with two organizations and each organization has its own Certification Authority (CA) and both are connected to the same channel. In addition, there is an *orderer* with its own CA.

The study was executed in a processor Intel i7-8565U (8) @ 4.600GHz and was divided in three experiments. The first one studied the performance with 100 reports stored in the blockchain, second and third ones had 500 and 1000 reports stored, respectively.

Figure 3 presents the time needed for storing the reports in 500 executions of the audit. The majority of the measurements are depicted in blue, except for the first execution that took much longer since it required AUTOAUDITOR to gather vulnerabilities information from external sources, i.e., vulnerability score and related metasploit modules. As soon as AUTOAUDITOR collects vulnerabilities metadata, a SQLite database is generated and populated with that information, acting as a cache in subsequent executions. Due to extremely different times between first and following executions, we opted to add another y-axis on the right side for the first measurement.

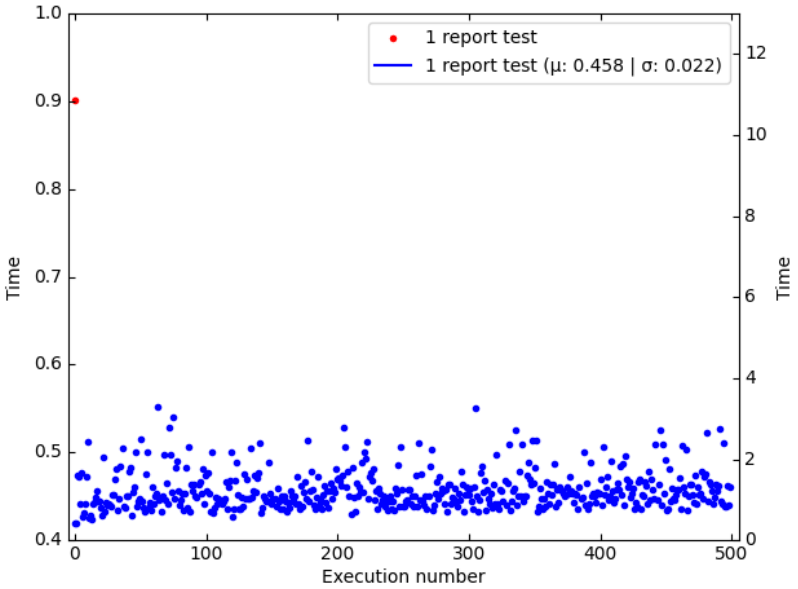

Figure 3. Time required to store a report.

Storing a report spent 0.458 s in average with a standard deviation of 0.022. Similar results were obtained when experimenting with executions storing 100 and 1000 reports, with average times of 0.442 and 0.466 s and standard deviations of 0.05 and 0.028 respectively. That slight penalty during upload shows that the system is usable for storing the audit reports of even large organizations. This performance can cope with changes in the number of devices, networks, maintenance operations, and applicable vulnerabilities published by CERTs, even if continuous monitoring is required.

Figure 4 shows the time spent querying a single report with `GetReportById()` in multiple executions. Querying a single report spent 0.665 s in average with a standard deviation of 0.069.

Figure 5 shows the time spent doing a bulk query, calling `GetReportsByOrganization()` in multiple executions. We observe a significant improvement in time compared to single report queries, taking almost the same time for 100 reports.

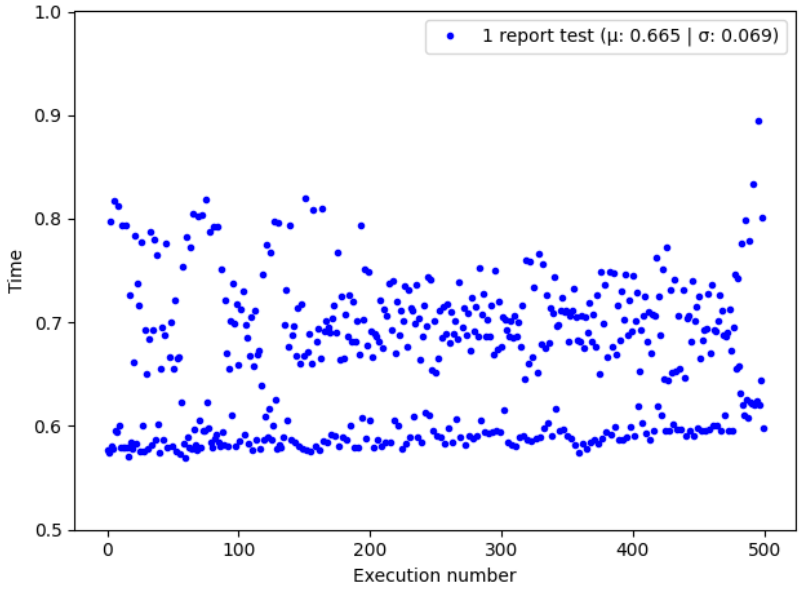

Figure 4. Time required to query a single report.

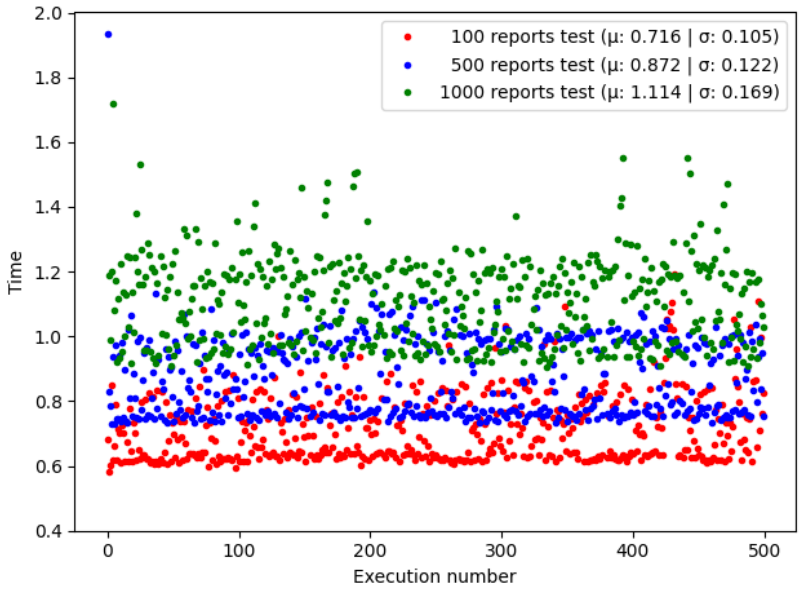

Figure 5. Comparison of time required to query multiple reports at once: Organization.

Figure 6 shows the time spent doing a bulk query, calling `GetReportsByDate()` in multiple executions. It can be seen that batch queries by date take slightly less time than queries by organization.

Table 2 compares the mean and standard deviation between the time required to complete a query in the experiments. As was expected, doing a batch query took significantly less time than querying reports individually due to the extra time added processing the transactions. We can see that HLF allows the querying of many reports at the same time without being notoriously affected, keeping the times low.

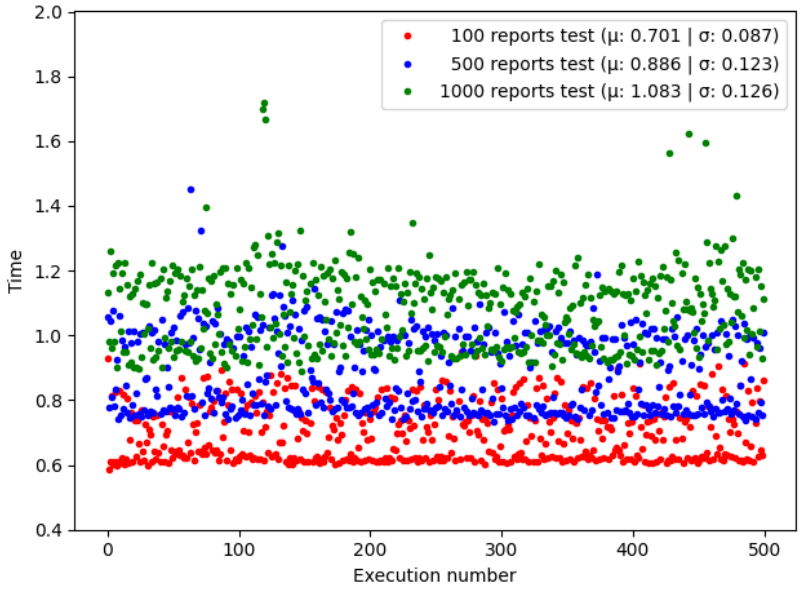

Figure 6. Comparison of time required to query multiple reports at once: Date.

Table 2. Comparison of mean and standard deviation in every experiment.

Query	Value	Reports			
		1	**100**	**500**	**1000**
GetReportById	μ	0.665	-	-	-
	σ	0.069	-	-	-
GetReportsByOrganization	μ	-	0.716	0.872	1.114
	σ	-	0.105	0.122	0.169
GetReportsByDate	μ	-	0.701	0.886	1.083
	σ	-	0.087	0.123	0.126

The apparent improvement in time of date-query can be attributed to the use of integer in `GetReportsByDate()` composite keys, allowing faster query indexing, unlike its counterpart `GetReportsByOrganization()` that uses string in composite keys.

7. Conclusions

Similar to other countries, the policies defined by the Spanish National Commission for the Protection of Infrastructures and Cybersecurity (abbreviated in Spanish as CNPIC) thrust two driving forces:Information sharing and public-private partnership. Sharing information of security audits results can bring benefits to the overall system. Our aim is not to substitute other tools that are already being used for security information exchange. In the case of Spain, we do not want to substitute neither the PI3 (Platform of Infrastructures-based Information Exchange), nor HERMES. The main contribution of this paper is enhancing AUTOAUDITOR with a blockchain-based AAAA scheme to gather audit information and to share in an accountable way cyberthreat intelligence. We provide AUTOAUDITOR as a new tool that can be integrated as a new source of security information and as platform to foster collaboration within the community of cyberthreat intelligence.

The tool helps to achieve continuous monitoring by integrating the audit system in a semi-automated way in the inventory control system of electrical grid companies. The audit result records are persisted in a permissioned blockchain, since blockchains are by design resistant to data modification. The results of the performance tests show that the system can be adapted to the inventory systems of electrical companies.

Future work will be intended to perfect the tokenization of audit trails and to define more fine-grained access control policies by leveraging HLF channels and defining more precise access control lists in the chaincode associated with AUTOAUDITOR. In this way, the set of sharing policies would be improved, which eventually paves the way for a more fluid habit of collaborative work in investigations of security incidents.

Supplementary Materials: A supporting video is available at https://www.youtube.com/watch?v=iAJVCirZFCg.

Author Contributions: Conceptualization, D.D.-S. and F.A.-M.; methodology, A.M.-L. and S.C.-M.; software, S.C.-M.; validation, S.C.-M. and D.A.; investigation, A.M.-L., S.C.-M., D.A., D.D.-S. and F.A.-M.; writing–original draft preparation, A.M.-L., S.C.-M. and D.A.; project administration, D.D.-S. and D.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by National R&D Projects TEC2017-84197-C4-1-R, TIN2017-84844-C2-1-R, by the Comunidad de Madrid project CYNAMON P2018/TCS-4566 and co-financed by European Structural Funds (ESF and FEDER), and by the Consejo Superior de Investigaciones Científicas (CSIC) under the project LINKA20216 ("Advancing in cybersecurity technologies", i-LINK+ program).

Acknowledgments: We want to thank the anonymous reviewers for their helpful comments in improving this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alaton, C.; Tounquet, F. *Benchmarking Smart Metering Deployment in the EU-28*; Final Report, Technical Report, Directorate-General for Energy (European Commission); Tractebel Impact: Brussels, Belgium, 2020. [CrossRef]
2. Dempsey, K.; Goren, N.; Eavy, P.; Moore, G. *Software Asset Management*; Technical Report NISTIR 8011; NIST: Gaithersburg, MD, USA, 2018; Volume 3.
3. Dempsey, K.; Takamura, E.; Eavy, P.; Moore, G. *Software Vulknerability Management*; Technical Report NISTIR 8011 (Draft); NIST: Gaithersburg, MD, USA, 2019; Volume 4.
4. Vakilinia, I.; Tosh, D.K.; Sengupta, S. Privacy-preserving cybersecurity information exchange mechanism. In Proceedings of the International Symposium on Performance Evaluation of Computer & Telecommunication Systems SPECTS, Seattle, WA, USA, 9–12 July 2017; pp. 1–7.
5. de Fuentes, J.M.; González-Manzano, L.; Tapiador, J.; Peris-Lopez, P. PRACIS: Privacy-preserving and aggregatable cybersecurity information sharing. *Comput. Secur.* **2017**, *69*, 127–141. [CrossRef]
6. Chica-Manjarrez, S.; Marín-López, A.; Díaz-Sánchez, D.; Almenares-Mendoza, F. *On the Automation of Auditing in Power Grid Companies*; Ambient Intelligence and Smart Environments; IOS Press EBooks: Amsterdam, The Netherlands, 2020; Volume 28, pp. 331–340. [CrossRef]
7. Andoni, M.; Robu, V.; Flynn, D.; Abram, S.; Geach, D.; Jenkins, D.; McCallum, P.; Peacock, A. Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renew. Sust. Energy Rev.* **2019**, *100*, 143–174. [CrossRef]
8. Power Systems Management and Associated Information Exchange—Data and Communications Security—Part 1: Communication Network and System Security—Introduction to Security Issues. Available online: https://webstore.iec.ch/publication/6903 (accessed on 1 April 2020).
9. Initiative, J.T.F.T. *Managing Information Security Risk: Organization, Mission, and Information System View*; Technical Report; NIST: Gaithersburg, MD, USA, 2011.
10. Aydemir, B.; Stienen, C. SWAMP-in-a-Box v1.34.5. Available online: https://github.com/mirswamp/deployment/ (accessed on 1 April 2020).
11. Takaesu, I. DeepExploit: Fully Automatic Penetration Test Tool Using Machine Learning. Available online: https://github.com/13o-bbr-bbq/machine_learning_security/tree/master/DeepExploit (accessed on 1 April 2020).

12. Veracode Vulnerability Assessment Software. Available online: https://www.veracode.com/security/vulnerability-assessment-software (accessed on 1 April 2020).
13. Compton, A.; Lane, A. APT2: An Automated Penetration Testing Toolkit. Available online: https://tools.kali.org/information-gathering/apt2 (accessed on 1 April 2020).
14. Tiwari, A. ArcherySec: Centralize Vulnerability Assessment and Management for DevSecOps Team. Available online: https://archerysec.github.io/archerysec/ (accessed on 1 April 2020).
15. Boelen, M. Auditing, System Hardening, Compliance Testing. Available online: https://cisofy.com/lynis/ (accessed on 1 April 2020).
16. Berta, S.A.; Villanueva, N.S.; Romanos, P.; Benítez, D.; Pepe, M. Crozono: Leveraging Autonomous Devices as an Attack Vector on Industrial Networks. Available online: https://www.blackhat.com/eu-16/arsenal.html (accessed on 1 April 2020).
17. Romanos, P.; Berta, S. A Framework to Test Your Security Perimeter with Drones & Robots. Available online: https://github.com/johnjohnsp1/CROZONO (accessed on 1 April 2020).
18. Riera, G.; Medina, M.A.R. Python-Faraday: A Multiuser Penetration Test IDE. Available online: https://tools.kali.org/information-gathering/faraday (accessed on 1 April 2020).
19. Cran, J.; Kaiser, T.; Bensalah, A. Intrigue Core: Discover Your Attack Surface. Available online: https://core.intrigue.io/ (accessed on 1 April 2020).
20. Jopling, B. Leviathan: Wide Range Mass Audit Toolkit. Available online: https://github.com/utkusen/leviathan (accessed on 1 April 2020).
21. Sift Through Embedded Device Files to Identify Potential Vulnerable Indicators. Available online: https://github.com/CERTCC/trommel (accessed on 1 April 2020).
22. Langer, L.; Skopik, F.; Smith, P.; Kammerstetter, M. From old to new: Assessing cybersecurity risks for an evolving smart grid. *Comput. Secur.* **2016**, *62*, 165–176. [CrossRef]
23. Smart Grid Coordination Group. Smart Grid Reference Architecture (SGAM). Available online: https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group1_reference_architecture.pdf (accessed on 1 April 2020).
24. Dobrowolski, Z.; Sulkowski, L. Supreme Audit Institutions and importance of their trustworthiness. In Proceedings of the 35th International Business Information Management Association Conference (35th IBIMA Conference), Seville, Spain, 1–2 April 2020.
25. Wang, K.; Zhang, Y.; Chang, E. A Conceptual Model for Blockchain-Based Auditing Information System. In Proceedings of the 2020 2nd International Electronics Communication Conference, Singapore, 8–10 July 2020; pp. 101–107. [CrossRef]
26. Cha, S.; Yeh, K. An ISO/IEC 15408-2 Compliant Security Auditing System with Blockchain Technology. In Proceedings of The 6th IEEE Conference on Communications and Network Security (CNS 2018), Beijing, China, 30 May–1 June 2018. [CrossRef]
27. White, J.; Daniels, C. Continuous Cybersecurity Management Through Blockchain Technology. In Proceedings of 2019 IEEE Technology Engineering Management Conference (TEMSCON), Atlanta, GA, USA, 12–14 June 2019; pp. 1–5.
28. International Association for Trusted Blockchain Applications. Available online: https://inatba.org/wp-content/uploads/2020/06/Co-Chairs_presentations_GA_10June.pdf (accessed on 1 September 2020).
29. ETSI GR PDL 001: Permissioned Distributed Ledger (PDL); Landscape of Standards and Technologies. Available online: https://standards.iteh.ai/catalog/standards/etsi/1dea1899-1b85-4e6c-974f-78a6546f037d/etsi-gr-pdl-001-v1.1.1-2020-03 (accessed on 1 April 2020).
30. Cha, J.; Singh, S.K.; Pan, Y.; Park, J.H. Blockchain-Based Cyber Threat Intelligence System Architecture for Sustainable Computing. *Sustainability* **2020**, *12*, 6401. [CrossRef]
31. Ángel Prada-Delgado, M.; Baturone, I.; Dittmann, G.; Jelitto, J.; Kind, A. PUF-derived IoT identities in a zero-knowledge protocol for blockchain. *Internet Things* **2020**, *9*, 100057. [CrossRef]
32. Alcaraz, C.; Rubio, J.E.; Lopez, J. Blockchain-assisted access for federated Smart Grid domains: Coupling and features. *J. Parallel. Distr. Com.* **2020**. [CrossRef]

33. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; Caro, A.D.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In Proceedings of the Thirteenth EuroSys Conference, EuroSys 2018, Porto, Portugal, 23–26 April 2018. [CrossRef]
34. Nmap Security Tool Used to Discover Potentially CVEs that Affects Services in Detected Open Ports. Available online: https://github.com/alegr3/CVEscanner (accessed on 1 April 2020).