The Role of Cybersecurity in Engineering Education

Charalambos Konstantinou
Florida State University
Tallahassee, FL 32310
ckonstantinou@fsu.edu

Ioannis Zografopoulos
Florida State University
Tallahassee, FL 32310
izografopoulos@fsu.edu

XiaoRui Liu
Florida State University
Tallahassee, FL 32310
xliu9@fsu.edu

Juan Ospina
Florida State University
Tallahassee, FL 32310
jjospina@fsu.edu

Abstract

Cybersecurity spending was expected to reach $123 billion in 2020 yet, by some estimates, the global shortage for cybersecurity professionals reached 4.07 million in 2019. It is evident that, due to the growing cyber threats, is essential to grow the cybersecurity educational and research capacity to better secure and protect our vital information systems. At the same, it is important to prepare the next generation of cybersecurity experts to meet the needs of the field while creating a dynamic and credentialed workforce.

1 Introduction

Computer systems have become a vital part of our everyday professional and personal life (e.g., online banking, social networking). These tasks can, however, expose the users to various security threats (e.g., credit card number theft, personal information leakage). Most importantly, critical infrastructure such as power grid, water treatment facilities, transportation networks, etc. are considered cyberphysical entities that rely on safe and secure information, computing, and communication platforms [37]. Therefore there is a need for designing secure systems. Education in the field needs to provide both theoretical and practical concepts of cybersecurity, including topics such as symmetric ciphers, basic number theory, public key cryptosystems, digital signatures, hashes, message authentication codes, key management and distribution, authentication protocols, vulnerabilities and malware, access control, network security, etc.

The learning objectives in regards to the desired outcomes for cybersecurity education need to provide students the skills to distinguish the broad set of technical, social, and political aspects of cybersecurity, describe the vulnerabilities and threats posed by criminals, terrorists, and nation states to critical infrastructure, and relate the nature of secure software development, operating systems (OS) and database design. In addition, students should be able to interpret security guarantees, assess the level of security provided by a cryptographic protocol, identify the role security management in cybersecurity defense, and explain common vulnerabilities in computer programs including buffer overflow vulnerabilities, time-of-check to time-of-use flaws, and incomplete mediation. Lastly, education in cybersecurity need to provide students the skills to identify the security management methods to maintain security protection, be able to discuss the legal and social issues at play in developing solutions, and get the skills to apply theoretical concepts in practice by using a programming language to implement attacks and defenses against computer systems.

In this paper, we provide a list of the important topics, also presented in Table 1, that need to covered in an introductory, yet comprehensive, course to cybersecurity in order for the role of cybersecurity

Transactions on Techniques in STEM Education (2021).

Table 1: A list of topics that can be covered in an ‘introduction to cybersecurity’ course.

A/A	Topic
1	Security mindset: threat modeling, design principles, and ethics
2	Authentication and access control
3	Cryptography: symmetric and asymmetric crypto-primitives
4	Program security (non-malicious programming oversights)
5	Program security (malicious code and countermeasures)
6	Web security
7	Operating systems (permissions, security in the design, rootkits)
8	Network security
9	Databases security
10	Cloud computing
11	Hardware security
12	Cyberphysical systems and Internet-of-Things security
13	Privacy
14	Management (security planning, handling incidents, risk analysis)

education to be enhanced within engineering curricula [13], and stand as the “backbone of building strong cybersecurity professionals and informed citizens” [19]. The depth in each topic can be determined by instructors according to the background level and computing systems knowledge of the students. This paper presents a list of essential cybersecurity topics for education while providing an overview of each topic and the important aspects of it.

2 Topics for Cybersecurity Engineering Development and Education

2.1 Security mindset: threat modeling, design principles, and ethics

Security is about making informed decisions with risk management in order for a collection of properties (such as confidentiality, integrity, availability (CIA)) to hold in a given system (where a system is anything from hardware, software, firmware, and information being processed, stored, and communicated) under a given set of constraints which define adversaries and their capabilities. Thus, it is important towards securing critical assets to: (1) first understand adversaries motives, capabilities, and degree of access, and then (2) identify the security policies and properties to be enforced on the assets that need to be protected. In other words, it is critical to perform threat modeling and risk assessment to understand the system, the potential attackers and the possible (known) attacks, determine what is the expected risk (quantitative or qualitative) because of an attack, and employ, if needed, security mechanisms (mitigation).

For any computing system, the desired functioning (specifications) need to be translated from the design stage to the implementation. Basic themes for design principles for secure systems are often characterized by: (1) simplicity (KISS[1] which makes design and interactions easy as well as easy to prove the safety of the system, and (2) restriction which aims to minimize the power of entities. Furthermore, design principles for secure systems typically follow selection and combination of the Saltzer and Schroeder list [26]: principles of least privilege, fail-safe defaults, economy of mechanism, complete mediation, open design, separation of privilege, least common mechanism, and psychological acceptability.

The security mindset of future engineers needs to also cultivate ethical considerations of systematizing, defending, and recommending concepts of right and wrong behavior within not only system design to formulate proper policies to guide actions but also personal deontological behavior. As Institute of Electrical and Electronics Engineers (IEEE) puts it, in its code of ethics [11], ethical behavior needs to be accompanied by following: (1) high standards of integrity, responsible behavior, and ethical conduct in professional activities, (2) treating all persons fairly and with respect, to not engage in harassment or discrimination, and to avoid injuring others, and (3) striving to ensure that ethical behavior is upheld by colleagues and co-workers.

[1] KISS is an acronym for “Keep it simple, stupid” as a design principle noted by the U.S. Navy in 1960.

2.2 Authentication and access control

Authentication binds an identity to a subject in a two step process: (1) identification – establish identity to system, and (2) verification – process verifies and binds entity and identity. From email and Internet banking accounts, to e-commerce and social media, one of the most pervasive method for authenticating users is using secret passwords, or *something you know*. The implicit assumption is that whoever proves knowledge of the correct password associated with an account, must be the legitimate owner. To prevent unauthorized users from predicting one's password, modern computing platforms impose strict policies on acceptable passwords, such as the use of special characters and numerals, which increases the memorization burden so that end users tend to reuse their passwords. According to a 2006 study by Florencio and Herley, users reuse the same password across 4.48 different websites on average [9]. At the same time, as the number of reported cybersecurity breaches to online services continues to grow, for example, the website `www.haveibeenpwned.com` catalogs more than 0.5 billion leaked passwords, users are asked to 'never reuse passwords' across different services to prevent an attacker that compromises the password database of one service from applying the stolen passwords to another, uncompromised service. Due to the contradiction between usability and security requirements, the effectiveness of passwords as an authentication mechanism has been challenged.

Through the years, several techniques have been proposed to address the limitations of password-based authentication. One popular solution is the use of a password manager that uses a database to encrypt and store all user's passwords; in this case, the user has to memorize a master password to unlock this database. Nevertheless, the impact of losing or forgetting the master password is very high, as access to all passwords in the database will be lost. At the same time, since master passwords are valuable targets for attackers, their increased complexity requirements make them harder to memorize.

Since passwords could be stolen, authentication mechanisms may incorporate additional factors, such as physical tokens (*something you have*) or biometric measurements (*something you are*). If multi-factor authentication is used, stealing one's password would not be enough to gain access to a service; however, the security benefits of biometrics and tokens are countered by additional usability concerns. For example, biometric measurements cannot be changed, so if attackers obtain a copy of such measurements (e.g., fingerprints), they will be able to reuse them forever (i.e., there is no way to "reset" one's biometrics). Likewise, a physical token should always be with the user, which increases the risk of loss or theft.

In addition to authentication requirements that put restrictions on who (or what) can access system, authorization enforces restrictions on actions of authenticated users. It is a form of access control, i.e., a collection of methods and components that supports security properties of confidentiality and integrity in order to allow only authorized subjects to access permitted objects, e.g., least privilege philosophy – a subject is granted permissions needed to accomplish required tasks and nothing more. Access control designs define rules for users accessing files or devices, and there are three common ways to establish them, via: mandatory access control, discretionary access control, and role-based access control.

2.3 Cryptography: symmetric and asymmetric crypto-primitives

Cryptography[2] is the study of mathematical techniques to achieve various goals in information security, such as confidentiality, authentication, integrity, non-repudiation, etc. and essentially achieve to communicate secretly through the use of cipher, i.e., an algorithm to perform encryption and/or decryption. In symmetric/private key cryptography, a sender and a receiver share a common (private) key and encryption and decryption are performed using that shared private key. On the other hand, in asymmetric/public key cryptography every user has a private key and a public key. Encryption is performed using the public key and decryption using the private key.

One of the methods to establish and manage public key encryption is public key infrastructure (PKI); a standardized method which by using the required policies can efficiently authenticate the identities of communication entities and systematically provide access to system resources while accounting for all the performed transactions. In this way, the principles for authentication, authorization

[2] The word traces back to the Greek roots "kryptos – ϰρυπτός" meaning "hidden," and "graphy – γραφή", meaning "to write."

and accountability are satisfied through the creation, distribution and management of digital and cryptographic public key certificates. Thus, PKI-aided systems can effectively exchange encrypted data over insecure media, e.g., internet, knowing that the identity of both the sender and the receiver have been verified by a trusted third party, the certification authority (CA). PKI is the cornerstone of most distributed systems involving big amounts of exchanged data over different organizations using widely adopted technologies such as digital signing and encryption. They serve as the root-of-trust between untrusted interacting endpoints and are widely adopted in business applications, banking, internet-of-things (IoT) environment and also critical infrastructures. PKI certificates issued by the CA encompass the public key required for the encryption of the data, cryptographic authentication signatures for the identity of the party that sent those data, as well as other ancillary ones, e.g., the certificate date of creation and/or its validity period, etc.). The exchange of sensitive data and system control information can be performed even without the use of PKI using encryption algorithms that ensure confidentiality. However, in those cases we can no longer be certain about the identity of the parties involved in that communication round; there exists no authentication stage. In essence, nowadays systems rely heavily on PKI for the exchange of information since it can provide both confidentiality – through public key cryptography – and authentication because the identities of the involved parties have exchange certificates securely using the CA.

2.4 Program security (non-malicious programming oversights)

Computers are a crucial part of our lives and indispensable tools for completing our everyday work-related assignments, thus being able to safeguard the security of private (e.g., bank account, social accounts, etc.) and enterprise data (e.g., business network credentials) is paramount. However, to enhance information security we first need to understand how computers operate. Programs are the epicenter of every computational system and define how a computer is expected to operate via a list of commands, also known as *instructions*. Programs are behind every task completed by computers, and are typically written by programmers using some sort of computer language (e.g., C, C++, Python, Java, etc.) since writing them using machine language – digital ones and zeros – is tedious. Typical computer program examples include applications like web browsers, text and document editors, video games, peripheral device drivers, as well as the OS, which realize the interface between hardware (e.g., CPU, memory, data storage, etc.) and software applications (i.e, other computer programs).

Given the variety of different computer programs that exist and their specific objectives, ideally we would not want any of them to operate unexpectedly. However, we should remember that programs are written by programmers, and as all people, programmers too can make mistakes. In most cases these errors are unintentional and non-malicious (i.e., they do not deliberately aim to jeopardize the system security). On the other hand, some of them can lead to security vulnerabilities if not properly addressed by corresponding updates, also known as *patches* in the information technology (IT) notation; one of the most common examples of such programming oversights is the buffer overflow. Buffers are areas of memory used for storing data which are transferred from one section of a program to another, or exchanged between programs. In information security and programming, a buffer overflow is an anomaly where a program, while writing data to a buffer, exceeds the buffer's boundary and overwrites adjacent memory locations. Buffer overflows are usually caused by mismatched inputs, a typical scenario is when somebody assumes a certain size for the data (used in a program) and allocates the buffer size accordingly. However, during an *abnormal* operation, more data can be generated, resulting in writing past the end of the buffer. Overwriting the adjacent data or executable code stored in those memory addresses can result in erratic program behavior, including memory access errors, incorrect results, and system crashes. By careful coordination of the data sent to a buffer, an attacker can overflow it, and overwrite areas holding executable code with malicious commands. Since buffers are widely used in OS implementation, a well organized buffer overflow attack can result in gaining unlimited control of the target device.

2.5 Program security (malicious code and countermeasures)

Apart from the non-malicious and unintentional programming errors existing in computer programs which we discussed in Section 2.4, another category of programs exists which deliberately aim to harm our computer systems. These types of malicious programs, also known as malicious software – malware, are written with harmful intent and upon execution can compromise our systems. Some of the most common ways for malware deployment and execution are via, (1) downloading and installing, (2) running executable files from untrusted sources (e.g., sent to our email), (3) inserting removable media on our system (e.g., via usb), or (4) by exploiting an unintentional system flaw which might have already been identified (e.g., buffer overflow) or can be completely knew (zero-day).

The most common malware categories include computer viruses, worms, spywares, ransomwares etc. Malware categories are named after the approach that they pursue to exploit a computer system or a target device in general. For instance viruses, upon reaching a system, can start propagating from files to directories and so forth, infecting as many of the system data as possible. Similarly, worms after they infect a system they start replicating themselves recursively attempting to infect as many computing resources as possible; this exponential growth resembling actual worms has given them their name. Spyware on the other hand, aim to collect information from their victims – by violating their data privacy – that can be later used to harm them. For example, keylogger spyware, as its name suggests monitors the user keyboard activity (i.e., keystrokes pressed) and can discover sensitive information such as user credentials and passwords, credit account numbers, websites visited, etc. In ransomware cases however, the malware would demand from their victims to pay the ransom, else their data would either be publicly disclosed or will be permanently inaccessible. In the latter case, the user data are copied and then encrypted, while the original data are permanently removed from the system making any attempt to restore them futile. Then, the ransomware will extort users until they pay the ransom so that a decryption key is provided and users are again able to decrypt and access their data. One of the most common ransomware attacks is the WannaCry incident in 2017 [10].

Although in most cases that malware infiltrate computer systems it is due to user oversights, there are cases where attackers exploit system faults to deploy their malware. For instance, two well documented attack examples that leverage the aforementioned buffer overflow vulnerability to compromise their targets, are the Morris worm [8] and the Structured Query Language (SQL) slammer [33] worm. More information about buffer overflow attack coordination can be found in [7, 22]. Nonetheless, we should preemptively protect our devices from malware by hindering their intrusion attempts, and mitigating them in the unfortunate cases that they already reside within our system. Antivirus programs offer a comprehensive security solution able to meet both of the two mentioned goals (i.e., deal with malware pre or post-compromise). Antivirus software, with their plethora of features such as automatic updates, removal tools, web browsing security extensions, suspicious file probing routines, etc. can potently identify and mitigate any type of malware before it infects our system and endangers our data security. Of course, keeping our systems current and updated, by installing any issued security patches, is crucial, while user caution is also advised when dealing with untrusted resources (e.g., files, links, web-pages, etc.) in order to safeguard data security and our privacy.

2.6 Web security

The rapid development of the World Wide Web allows user's remote access through the internet to publish information or complete transactions. The web becomes an attractive target for the attackers to disclose confidential information or cause catastrophic financial consequences by cyberattacks. Web security requires significant attention in order to protect web servers, web users, and their surrounding environment by detecting, preventing, and responding to cyber-threats. The common web security vulnerabilities are (1) web malware attacks, (2) Denial-of-Service (DoS)/Distributed DoS (DDoS) attacks, (3) brute force attacks, (4) SQL injections, and (5) cross-site scripting attacks. The web malware attacks refer to harmful software developed by attackers with the intention of gaining access, collecting information, disrupting availability, hijacking the website, launching attacks against other websites, and etc. The DoS attacks aim to make the targeted web unavailable or slow it down by overwhelming it with flooding internet traffic. The DDoS attacks refer to a DoS attack that comes from many different sources at the same time. The brute force attacks are performed by systematically searching the credentials from all potential combinations to gain access as an admin. SQL injections occur when the malicious code is inserted to exploit a software vulnerability which will allow attackers to collect or modify the existing information from the database. Cross-cite scripting attacks enable attackers to inject malicious scripts into web pages for the purpose of executing it by the other end users. The attackers could take over the user's accounts and steal credentials.

In order to protect the websites against cyberattacks, the Cybersecurity and Infrastructure Security Agency (CISA) provides seven procedures for organizations to follow for enhancing web security [5].

1. Secure domain ecosystems: *(i)* The registrar and domain name system (DNS) records of all domains need to be reviewed. *(ii)* The new credentials are required for avoiding such attacks that leverage default credentials. *(iii)* Multi-factor authentication (MFA) is enforced to enhance security by successfully presenting more than one factors.
2. Secure user accounts: Other than enforcing the MFA and requiring a change of the default credentials, the least privilege is achieved in this procedure that only the necessary ones are provided to the users while the unnecessary ones disabled.
3. Continuously scan for critical and high vulnerabilities The critical and high vulnerabilities of configuration and software need to be scanned within a certain period. Any unsupported system, applications should be removed.
4. Secure data in transit Hypertext Transfer Protocol Secure (HTTPS) and HTTP ((Hypertext Transfer Protocol)) Strict Transport Security (HSTS) are recommended to ensure encrypted communication between the web and users. The weak ciphers with an insufficient key length need to be disabled.
5. Backup data The critical data and system configurations should be stored in a safe and physically remote environment which can be used as a backup solution at any time. Disaster recovery scenarios need to be verified.
6. Secure web applications The top ten critical web applications should be identified and remediated. The MFA should be utilized for user logins as well as other underlying website infrastructure. The logging and regularly audit website logs need to be sent to the centralized log server which is necessary to detect security incidents.
7. Secure web servers The specific security checklists of each application can be utilized to improve the server's reliability. Additionally, the network segmentation and segregation can make it harder for attackers to compromise the whole system. Moreover, only the necessary data should be saved on the web server to avoid public access.

2.7 Operating systems (permissions, security in the design, rootkits)

As already discussed, OS are a critical part of every computer system since they orchestrate the communication between hardware (e.g., CPU, ram, data storage) and applications employed by users to perform specific tasks. As a consequence, with the term OS security we refer to the process responsible for safeguarding OS confidentiality (e.g., only authenticated users can access system data), integrity (e.g., only authorized users can modify the system according to predefined manners), and availability (e.g., system resources should be available to users at any given time). Furthermore, OS security incorporates all the preventative countermeasures to ensure that system data will not be disclosed, modified, malformed or destroyed, and protects systems against viruses, worms, spywares, ransomwares, or any other potential malicious attempt.

Computers are accessed by different entities or groups (e.g., users, groups with escalated privileges, system administrators, third parties, etc.), an it is the OS responsibility to identify these entities and provision them with the necessary access and privileges (i.e., permission to perform system tasks). In more detail, OS by design correlate each system user category with specific access to system resources, as well as permission to perform certain operations based on their clearance level. This practice is commonly referred to as the *principle of least privilege*, authorizing users with the minimum access required to carry out their jobs, in order to better protect system assets. For example, in the same manner we would not allow a university student to be able to modify their course grade records, and this functionality would be reserved for their professors, OS also allocate certain access permissions and privileges to every user group. These access control policies (role based access control – RBAC) ensure, that even in the case of a security breach, the malicious attacker will have limited authorization, and thus will not compromise the whole system.

Attackers in their endeavor to wreak havoc in computer systems however, and knowing the security considerations and enforced RBAC OS policies, have developed a new type of malware able to creep into these systems, achieve privilege escalation and gain access to restricted system resources with the aim to destroy them. This type of malware is called "Rootkit", and gets its name from the word "root" which is a common term used in Nix-based OS; root users have unrestricted access and privileges to all system resources, similar to administrator accounts in Windows-based architectures. While the second component of the word, "kit", denotes the necessary programs exploited by attackers to achieve the aforementioned unauthorized and unrestricted system access. Rootkits, pursue their malicious objectives secretly (without the user noting any suspicious behavior), and the situation is further exacerbated by the fact that recent Rootkit iterations are also able to reside stealthily inside OS, without triggering any security mechanism.

However the battle against malware is not yet lost and there are several defense strategies that we can opt for in order to defend against sophisticated attacks targeting OS security such as rootkits. Some common practises to enhance OS security include, (1) regularly checking and installing OS patches, (2) enforcing user permissions and RBAC policies, (3) preserving the OS current by removing unnecessary or dated applications and services since they could become potential attack entry points, (4) probing (i.e., monitor and log) incoming and outgoing system traffic (e.g., via internet) using firewalls, and (5) install and keep updated antivirus suites or other intrusion detection mechanisms which could prevent malware spread or mitigate them if malware have successfully breached the preliminary OS defenses.

2.8 Network security

The necessity for connectivity of our modern world has driven our economy, society, and our overall livelihoods to a point where the internet, i.e., large interconnected communication networks, is starting to be cataloged as a basic right or essential need for any human being. Since our lives rely heavily on communication networks, it is essential for us to ensure that every interaction that occurs through these networks is secure and private. The unauthorized access of our data by attackers (or adversaries) has the potential to cause severe adverse effects to businesses, governments, and regular people by affecting their social relationships, economics, health, and their lives in general. So, as seen, network security is an essential topic that needs to be learned, understood, and comprehended not only by security experts and network engineers but everyone that uses communication networks.

Communication networks are made up of four main elements: nodes, links, protocols, and messages (commonly known as packets). The nodes represent an abstraction of any computing device (e.g., computers, smartphones, micro-controllers, etc.) or networking component (e.g., switches, routers, hubs, etc.) that serves as an endpoint or backbone of the communication network infrastructure. An *endpoint* node represents a computing device that sends or receives messages to and from other devices while an *backbone* node represents the devices that allow endpoints to exchange messages (or packets) by routing them throughout the network. The links in a network represent the physical (wired or wireless) connection between the individual nodes in a network and they typically refer to the type of medium used in the network such wires, optical fiber, microwaves, wireless signals, or satellite. Each one of the mediums mentioned has its strengths and weaknesses in terms of costs, security, and signal degradation due to distances. Finally, the protocols in a communication network are defined as the set of rules used to standardize the transmission of messages between two or more computing nodes. It is essential for students, or network security stakeholders, to first have a good understanding of how communication networks work, how packets/messages travel through a network, the basic types of communication networks that exist (i.e., LANs, MANs, and WANs), and how communication protocols are used in real systems before diving into network security.

After grasping the concepts related to how communication networks work, network security needs to be presented in a way such that stakeholders can identify the possible threats, attacks, and vulnerabilities a network may be exposed to and how attackers may attempt to compromise devices by leveraging these weaknesses in the network. From a general point of view, a successful network attack can be defined as one that violates one or more security goals from the security triad, i.e., CIA. An attack that violates confidentiality, commonly known as wiretapping or interception, is an attack where the attacker is able to intercept unauthorized data/messages flowing through the network. Here, we can find man-in-the-middle (MiTM) attacks and software-based attacks such as malware, rootkits, or trojans that can compromise an endpoint node. The strongest and most commonly used countermeasure against this type of attack is *Encryption*. Similarly, an attack that violates integrity, via modification and/or fabrication, is characterized by an attacker that is able to perform unauthorized changes to data exchanged or fabricate false messages. Some example attacks in this category are spoofing, replay, and selective forwarding attacks. A common countermeasure against integrity failures is *Hashing*. Finally, an attack that violates availability, via interruption, can be defined as an attack where the attacker is capable of denying authorized users access to a part of the network. Usually, in this type of attack, the attacker is capable of generating enough network demand to overwhelm a critical part of the network. DoS, flooding, and time-delay-attacks are examples of availability attacks. Common countermeasures for this type of attacks are networking protections such as *Firewalls*, *Virtual Private Networks (VPNs)*, and *Intrusion Detection Systems (IDS)*.

In order to improve network security in our modern communication networks, security experts often recommend six best practices for securing networks. These best practices are:

1. Identify the systems that need to be protected.
2. Separate systems into functional groups.
3. Implement a layered defense strategy for each group.
4. Provide access control into and between each group (via identification, authentication, and authorization).
5. Monitor the activities occurring between groups.
6. Limit the actions that can be executed within and between groups.

One of the best design strategies for improving network security is to follow the principle of "*least route*", which states that in a purpose-built network, a node should only be given the connectivity necessary to perform its function; no more no less.

2.9 Databases security

The enormous quantity of data that our modern technological world produces daily is something normal users are not quite aware of when performing everyday technology-related tasks, such as being in a video conference, sending an email, or simply chatting with a colleague. According to the World Economic Forum (WEF), around 294 billion emails are sent, 4 petabytes of data are created on Facebook, and 5 billion searches are made daily. It is estimated that by 2025, 463 exabytes of data will be created globally every day [6]. So, what does this has to do with databases and their security? In our modern computing systems, databases are the basis for storing large amounts of data. These data can be anything. It can range from company emails to high-resolution data coming from sensors measuring the status of our electric grid. Databases are the go-to technology for storing, securing, and backing up very critical data for diverse types of applications.

In essence, a *database* can be defined as a collection of data with a set of rules that define how to organize these data based on relationships that the data may have. The set of rules is defined by a *database administrator* who is also in charge of securing and controlling access to the data inside the database. To accomplish these tasks, the database administrator makes use of a *database management system* (DBMS) which can be defined as the software or program that database users and the administrator use to interact with the data inside the database structure. Every database that exists is composed of: *records* and *schemas*. A *record* is the elemental entry databases have, and it is made up of different values given to the different *fields* or *elements*. Each field or element in a database can represent information such as *Names*, *Addresses*, *Phone Numbers*, etc. On the other hand, *schemas* are a way of representing the logical structure of the database. A *schema* can be seen as a 'blueprint' of how the data in the database is constructed and organized. It should be mentioned that '*subschemas*' also exist and they refer to subsections of the database that are requested by users of the database. For example, a user may only want to access specific fields of the database (or may only have authorized access to those) and may query a subschema of the database that contains only the necessary information. Finally, a *Query* is defined as the way a user interacts, through commands, with the DBMS. This command is what we call a *query* and the result of the query is a subschema. Each DBMS has its own syntax for these commands but one of the most common use syntax, or language, is the SQL. Microsoft SQL, MySQL, and SQLite are services that use SQL as their main language. It should be noted that small variations between them may exist. The main advantages of using databases are presented below:

- Shared Access: users can use one common database.
- Controlled Access: only authorized users can access the data.
- Minimal Redundancy: individual users do not have to maintain their own sets of data.
- Data Consistency: if a value is changed, it will be reflected for all users almost simultaneously.

- Data Integrity: data can be protected against accidental or malicious changes.

Now that the concepts that make up databases and their advantages have been explored, let us dive into database security, security requirements, and how to achieve these security requirements. The security requirements of databases are the following:

1. Physical Database Integrity: data must be protected from data corruption caused by physical problems such as power failures and fires. The data must be able to be reconstructed if destroyed. The best way to protect the physical integrity of a database is to perform periodic backups.
2. Logical Database Integrity: the logical structure of the database must be preserved at all times, i.e., the modification of a value must not negatively affect others in the database. The best way to protect the logical integrity of a database is to perform periodic backups.
3. Element Database Integrity: the data contained in each element must be accurate. The best ways to accomplish this is by performing access control, do field checks, and have a changelog.
4. Auditability: administrators and authorized users can track what has been accessed and modified in a database and who or what performed the change/access. The best way to achieve the auditability requirement is to generate audit records that contain all changes made to the databases (i.e., all the reads and writes).
5. Access Control: data can only be accessed by authorized users. The best way to achieve this security requirement is to logically separate the database by user access privileges.
6. User Authentication: every user must be positively identified for audit trails and permissions. The best way to achieve the user authentication requirement is to perform user, password, and time-of-day checks based on the specific application. DBMS must be able to perform their own authentication and not rely on other programs or the OS itself.
7. Availability: users must be able to access authorized data when needed. To achieve this security requirement, mechanisms must be put in place to allow authorized users access to the data they are requesting at all times.

Finally, let us discuss a very important dilemma that emerges when designing databases. This dilemma is called the *Security vs. Precision* problem. In this dilemma, we must decide the tradeoff between how secure the data must by restricting its accessibility depending on how sensitive it is while making sure the queries users make to the database are precise (i.e., users are able to get all the data they are authorized and are requesting). This dilemma can be visualized using concentric circles, where the outermost circle represents the most precise data a user can access (public data) and the innermost circle represents the most sensitive data that no user must be able to access (restricted data). As you descend from the outermost circle to the innermost circle, data starts being less secure but very precise until it reaches the point where the data is very secure but less precise.

2.10 Cloud computing

The word cloud is a metaphor for the internet, where cloud computing refers to the delivery of computing services over the internet. The services include servers, data analytic, data storage, intelligence, and etc. In order to satisfy different demands, cloud computing falls into three categories. (1) Public cloud: the computing resources (servers, storage) are delivered by commercial providers who own and manage all hardware, software, and the supporting infrastructures. The leading public cloud providers are Amazon Web Services, Microsoft Azure, and etc. (2) Private cloud: the cloud computing resources serve exclusively a single organization on a private network. The private cloud can be physically located on the organization's data center or hosted by a third party. (3) Hybrid cloud: it combines both public and private clouds, allowing the data and application interact between them. The infrastructure can be scaled up to the public cloud during the overflow event without providing access to the entire data for external users. The hybrid cloud improves overall flexibility while securing the data behind a company firewall.

According to different levels of responsibility of cloud provider, the cloud computing services can be categorized into:

1. Infrastructure as a service (IaaS) is a benchmark which offers computing services and IT infrastructure such as servers, networking, storage, visualization on a pay-per-use basis.
2. Platform as a service (PaaS) provides a computing platform which includes all the components in IaaS, together with the OS, middleware, and development tools for users to develop, test, or deploy their software applications efficiently.
3. Software as a service (SaaS) integrates all features of PaaS with two additional components, data, and application, supplying a complete package of software applications over the internet. The providers are responsible for managing and maintaining the applications and underlying infrastructures.

In order to protect the cloud resources from cyberattacks, the typical cloud security strategies include:

1. Identity and access management (IAM): IAM is designed for ensuring that appropriate access to cloud resources are deployed to authorized users. It mitigates the potential threats when unauthorized users gaining access to internal assets or authorized users exceeding their privileges. IAM may feature such capabilities: *(i)* authenticating user identity, *(ii)* strengthening user authentication by using MFA and *(iii)* allowing and restricting user access by utilizing access control.
2. Encryption: Advanced encryption algorithms effectively ensure data security by translating the plaintext (original data) into ciphertext (encrypted data). The ciphertext is only readable by someone having the encryption key. Cloud data should be kept encrypted when in use, at rest (when the data are stored), and in transit (while the data sent from one place to another) to prevent any data leakage and exposure to attackers.
3. Firewall: A cloud firewall can protect cloud assets by blocking web traffic from potential malicious sources. There are two types of firewalls: *(i)* SaaS Firewalls designed to secure the network of organizations and its users *(ii)* Next Generation Firewalls as cloud-based services to secure incoming and outgoing traffic between cloud-based applications.

In reality, different sets of cloud security strategies corresponding to the specific cloud computing services can be adopted by organizations to improve the overall security. More specifically, to enhance the security of IaaS applications, it is required to:

- Encrypt cloud data
- Avoid misconfiguration of cloud resources

For safeguarding the OS and physical infrastructure of PaaS applications, it is necessary to:

- Ensure the provider's security management
- Use threat modeling
- Implement role-based access controls
- Manage inactive accounts

Moreover, the security of SaaS can be enhanced by utilizing such strategies:

- Detect rogue services and compromised accounts
- Deploy IAM
- Encrypt cloud data
- Enforce data loss prevention
- Check provider's security plan

2.11 Hardware security

Device security is a key concern for every device with computation capabilities ranging from smart IoT modules to large data center infrastructures. In most cases system security is based on cryptographic schemes, or information and communication security functions, however in all of these cases a critical assumption is made. It is assumed that the hardware, responsible for materializing the aforementioned security schemes and defined in some form of high-level language description, can be trusted apriori [25, 14]. The same can be argued for critical infrastructure – such as the power grid, transportation systems, water treatment facilities, etc. – which operates on the assumption that hardware not only is inherently secure, but also resilient to attacks aiming to disrupt nominal operation. However, it has been proven that even hardware can be compromised and backdoors can be planted on integrated circuit (IC) blocks during their manufacturing process awaiting their weaponization once certain trigger commands are received. For instance, Quo Vadis labs in UK, have reported that hardware backdoors have been found in military-grade weapon control systems[28].

Although the globalization of IC market – with billions of semiconductor-based devices precipitously overflowing our everyday lives– has enabled more economic ways of fabricating chips, the distributed fashion in which electronics are manufactured nowadays raises security concerns. In more detail, electronic systems are typically transferred through different facilities during their manufacturing stages, instead of building the whole system from scratch on a single fab allowing for better security and quality control of the devices. As a consequence, hardware trojans can be ported to these systems endangering the overall system security and allowing adversaries to perform catastrophic attacks on-demand [29]. With the term hardware trojans, we refer to device modifications (on the circuit level of the chip), which can cause the system operation to deviate from its nominal and expected behavior; we can think of hardware trojans as remotely controlled "time bombs" waiting to be armed by attackers with disastrous consequences [27]. Such trojans can be ported to microprocessors, microntrollers, or other commercial electronics devices granting adversaries the ability to enable or disable, maliciously control, leak sensitive victim information, or destroy the devices. Researchers have provided multiple methodologies to detect hardware trojans (e.g., side channel analysis, path delay monitoring, etc.), some of them compare the behavior of an IC under-test with a known and trusted one (golden circuit), where any significant performance deviation would indicate a compromised IC [12]. Other methods investigate IC on an architectural or transistor-gate level, searching for maliciously inserted logic, while statistical methods and machine learning approaches to detect hardware trojans have become increasingly popular recently [15]. Although identifying hardware trojans might require sophisticated and laborious processes, we should not underestimate the adverse impact that such stealthy attacks can have especially in mission critical systems (e.g., military applications, critical infrastructure, etc.).

Apart from hardware-enabled attacks, like the hardware trojans, hardware can also be used as a means of protecting device security (e.g., personal computers, mobile phones, smart IoT devices, etc.). A plethora of hardware-based security mechanisms has been reported for both the detection as well as the prevention of malicious access. For instance, hardware performance counters (HPCs) are a prime example of a hardware mechanism serving a crucial role for computer systems, which can, however, be used for security purposes too. In more detail, HPCs are special-purpose registers integrated into the microprocessing units for the monitoring and storing of low-level hardware events (e.g., load/store/branch instruction counting). Although they were originally implemented for the profiling and tuning of hardware applications, HPCs can also be leveraged to detect suspicious code execution patterns, malicious firmware and malware[32, 31, 16]. Other hardware aided security mechanisms instead of supporting the detection of adversarial activities focus on specific security objectives. Trusted platform modules (TPMs), for example, are standalone IC utilized for storing encryption keys and supporting hardware authentications between host devices. Trusted execution environments, such as *Intel SGX* and *ARM Trustzone, enable the segmentation of computing hardware architectures, into a secure and a non-secure environment, allowing minimal interaction between these environments and securing critical processes from threat actors*[21]. Other approaches enable secure authentications, using physical manufacturing variations to create digital "fingerprint-like" signatures that are difficult to predict or replicate. Recent works harness the manufacturing variations of lithium-ion battery cells, which are ubiquitous in our everyday lives (from mobile devices to electric vehicles), as a secure and low-cost alternative for hardware authentications[35, 36]. Thus, although hardware can be vulnerable against some harmful attacks, such as hardware trojans, a multitude of hardware mechanisms exist to combat these potentially disastrous attacks.

2.12 Cyberphysical systems and Internet-of-Things security

The rapid technological transition our society is experimenting is being facilitated by the introduction of modern computing devices into our day-to-day lives. Progressively, people are starting to rely on and depend more and more on devices such as smartphones, personal computers, and general IoT devices to accomplish everyday tasks such as shopping, working, and even meeting with other people. This rapid technological transition is also being experienced in other areas such as smart-manufacturing [30], healthcare [34], robotics [4], transportation [20], and even in critical infrastructure systems such as gas networks, chemical plants, water networks, and the electric power grid. Due to this transition, many of these systems have become cyber physical systems (CPS), where physical devices (operational technology (OT)) are interfaced with digital components (information and communication technology (ICT)) with the objective of improving the resiliency, controllability, and operation of these systems. Nonetheless, the introduction of computing and IoT devices that is driving the transition to CPS can be also a double-edged sword since it opens venues for cybercriminals and attackers wanting to compromise these systems and cause harm to users and stakeholders while gaining political or economic gains [23]. This is why understanding the possible vulnerabilities, threats, attacks, and defenses that exist in IoT as well as CPS is a crucial task that needs to be encouraged in our modern society.

More specifically, the term Internet-of-Things refers to the connection devices have with the Internet. This is from where the term 'smart-devices' also comes from and is directly related to the capability these devices have when performing computing operations using embedded processors, and the ability to exchange information between them making use of wired/wireless connections. Some example IoT devices we currently can find in the market are [24]:

- Smart home/appliances: Many companies are offering smart appliances designed to automate household chores. Currently, customers are able to find in the market smart appliances ranging from smart refrigerators, which can sense when you are running out of a particular ingredient, up to smart vacuum cleaners that can clean the house with minimal supervision.
- Smart health: Modern exercise monitors and Bluetooth-connected heart monitors exist currently in the market. Athletes use these devices to monitor their health and heart rates while doing strenuous exercise. Patients that suffer from heart problems or diagnosed with diabetes also use smart health devices to monitor their health and glucose levels.
- Smart transportation: Current developments in smart transportation are transforming the way users move through big cities. Nowadays, many public transportation infrastructures, such as subways, buses, and airplanes, heavily depend on automation and communication systems. Passengers are able to monitor the current location of many of these vehicles using GPS and satellite technologies; thus facilitating ticketing and boarding processes, among other processes. In addition, many car manufacturing companies are focusing their efforts on the development of automated self-driving cars that could become the standard in the roads of the future.

But, as with all things, not everything is 'perfect'. Automation comes with a high price tag when security is not taken into consideration. All of these IoT devices have at least one of the following privacy or security problems [24]:

1. Loss of privacy: since devices are constantly communicating the user's location, habits, and behavior, there exists risk related to how companies or other people can use this information against the user. This is also related to the loss of anonymity and how data can be used to categorize vulnerable populations.
2. Loss of control: A company that you trust your data to could potentially use that data for their own interests. In addition, cybercriminals may compromise part of your systems and lock you out, thus diminishing the control the user has on their devices.
3. Potential for subversion: Cybercriminals with political intents may poison the data received in some of your devices and feed the population with false stories or news that could create problems in society and cause disturbances.
4. Mistaken identification: Cybercriminals may compromise your devices with the objective of using them as part of a major cyberattack; or just simply stole your credentials and IDs compromising your identity.

Also, users from these devices can suffer from cyberattacks targeted at regular computing systems such as phishing, malware, or ransomware attacks. Imagine you have a very secure home network and you introduce a new IoT device to your home network. If the vulnerabilities in this device are compromised by an attacker, all your data and information could be easily stolen or even hijacked.

Remember that networks are as secure as their weakest device. For critical infrastructure, the term IoT is not usually used since devices and/or controllers deployed in these CPS networks may or may not connect directly to the Internet. Most of the time, these systems use proprietary, isolated, or very-well protected communication network infrastructures that facilitate the interconnection between different devices such as human-machine interfaces (HMIs), remote terminal units (RTUs), or intelligent electronic devices (IEDs), among others.

Nowadays, researchers and security engineers are focusing on developing qualitative and quantitative ways to characterize vulnerabilities and estimate the cyberphysical security of systems. One example of this is the use of scores such as the Common Vulnerability Scoring System (CVSS) or the Industrial Vulnerability Scoring System (IVSS) [1]. Other approaches also focus on developing and improving threat modeling techniques while taking into consideration complex attack-behaviors that can be capture using graph theory and attack graphs [3, 17]. However, since IoT environments and CPS are essentially facilitated via ICTs, the best practices to secure databases and communication networks can also be applicable to these systems. These best practices involve the use of IDS, VPNs, firewalls, and other types of security measures such as cryptography.

2.13 Privacy

Privacy is an essential right for individuals to seclude or selectively share their personal information, such as finances information or real-time location data. Since large amount of personal data is collected, stored or re-purposed to the third party online through the social networking, targeted advertising, and etc, more attention should be paid on internet privacy. The internet private data can be primarily sorted as personally identifiable information (PII) or non-PII. The PII can identify the individual without disclosing their name by using two factors identification, such as both physical address and age. The non-PII refers to the information which cannot directly identify an individual, like the IP address. For example, the user can be identified by matching their IP address with other collected information which cause a potential risk of the online privacy. Meanwhile, the HTTP cookies which save all history data to provide automatic access online also can be used for user tracking. Other potential risks can be from malware, phishing, web bugs, social engineering where confidential information can be divulged by psychological manipulation.

There are several common privacy principles used globally to protect the user's privacy [2]. (1) *Notice:* The individuals should be aware of the collection, utilization, and processing of their personal information. (2) *Choice:* The individuals are able to decide how their personal information can be used or processed. (3) *Security:* The company is required to protect the collected personal information in compliance with industry standards and applicable laws and regulations. (4) *Access, Accuracy, Integrity, and Quality:* To ensure the personal information is accurate, complete, relevant, and timely, the individuals should have access to their collected personal information in order to avoid mistakes. (5) *Retention of Personal Information:* The collected personal information can only be retained for a period, which should be complied with privacy law.

Privacy policy determines the specific personal information that can be gathered from website visitors, the way to use this information, and how to keep it safe. It should be posted on the site for the web visitor to know. In general, the privacy policy includes the types of collected personal information, the purpose of data collection, the security and access of the collected data, the affiliated organizations that could have access with this collected data, and effective date. In order to protect the citizen's personal information online, most countries have their web privacy law for all domestic companies to follow. For example, as one of the strictest privacy laws in the US, the California Online Privacy Protection Act (CalOPPA) protects personal information of people residing in California. General Data Protection Regulation (GDPR) is followed by all European Union members; Canada's Personal Information Protection and Electronic Documents Act (PIPEDA) are utilized to protect the personal privacy of Canadian citizens.

It should be noticed that privacy is different from the authentication that aims to provide proper permission to the authorized users. In order to confirm the specific user, the authentication process may need some private information such as something the user has, something the user is, or something the user knows. On one hand, it protects the data and reduce the risk by limiting users to access it. On the other hand, the data gathering and storing of the individual authentication can potentially affect privacy. Since the centralized password, public-key, or biometric systems pose potential security threats and privacy hazards, decentralized authentication can be utilized to reduce risks [18].

2.14 Management (security planning, handling incidents, risk analysis)

Assuming that we have followed every precautionary measure and applied all the discussed security suggestions, will our system eventually be secure against attackers? The answer is sadly no, because computer systems, similar to every other application, will always have “weak links” which can decrease the overall system Security. Sometimes, these “weak links” might be known or at least easy to identify, but on many occasions we might not even know them, such is the case with zero-day vulnerabilities for instance. Even though we cannot secure every potential attack entry point, still making sure that we abide by current security standards and best practises is paramount because, although these measures might not be the *perfect defense* they will still deter most attackers.

Thus, when discussing computer security or cybersecurity in general, we should always do it with the mindset that every system can be hacked; knowing that, we can proceed in devising strategies and plans to defend against a potential compromise. Security planning and risk management (SPRM) are critical operations which should be regularly performed in every system, organization, business, etc., to ensure that proper mechanisms are in place in the unfortunate event that an attack succeeds. In more detail, SPRM include, (1) the identification of crucial assets for the system operation, (2) the enumeration possible attack entry points, (3) the assessment of potential risks as well as their impact on the system, and finally (4) the prioritization of these risks and design of mitigation strategies to eliminate the threats or minimize their impact on the system. Similar to the concepts of a car’s or house’s insurance, SPMR serves as a worst case scenario policy indicating which incident response plan should be followed in the uneventful case of a cyberattack. During the SPRM and leveraging steps (1) – (3), security analysts construct potential attack use-cases targeting system assets, before the risk assessment procedure is initialized. During the risk assessment every prescribed attack is assigned a score, quantitative or qualitative, which indicates how adversely the system operation would be affected in such scenario. Then, in step (4) all the prescribed attacks are prioritized based on their risk scores, and mitigation plans are devised for each one of them. The aim of these mitigation strategies is to keep the system fully operational regardless of the attack, and typical risk management plans will either mitigate, avoid, transfer or accept the corresponding risk. For serious attack incidents for example., immediate remediation procedures might commence (i.e., mitigate risk). On the other hand, for low-impact risks we have the option to avoid it, in which case we opt for an alternative solution to bypass the risk, to transfer the risk to another system, company, group, etc., or to accept the risk and the associated system impact.

In any case having a structured SPRM can expedite attack handling, and even if the system impact cannot be totally eliminated, at least it can be restricted without endangering the whole system. The significance of SPRM could not prove more current with the recent influx of cyberattacks. The penetration of computer systems in all aspects of our lives also makes them prominent target for attacks. Especially in the cyberwarfare context, where malicious attackers might be backed by governments of competing nations, these cyberattacks on computing systems can have detrimental effects on the safety of people, businesses, organizations or the economy. Thus, discovering vulnerabilities, assessing risks, and employing comprehensive mitigation strategies – to protect our systems and enhance their security – arise as our most valuable allies in this thrust.

3 Conclusions

For many, cyberspace is an abstract concept only accessible through a laptop or computer screen. Cyberspace, however, touches almost every aspect of business, government, and life in general. From banking, communications, travel, shopping, and healthcare to automated systems, social media, and online resources, cyberspace is an integral component of the modern world. In this context, cybersecurity is freedom from and resilience against potential harm or other unwanted coercive change caused by others in and from cyberspace. Cybersecurity is the confluence of technology (systems, software, hardware, and infrastructure), people, and policy that protects our systems, data, and people in cyberspace.

The dependence on information technologies will continue to increase, along with the vulnerabilities associated with it. We have already seen cyber-attacks at all levels – targeting everything from critical infrastructure to individual citizens – that try to exploit those vulnerabilities. In addition, cyber-threats will come from nation-states and the criminal organizations they sponsor as well as from cyber-criminals and hacktivists with a political agenda. The public and private sectors, including government, academia, and industry, will have to make continued investments in cybersecurity education to address those threats in order to increase the number of skilled cybersecurity professionals.

Acknowledgement

Partial support of this research was provided by the Woodrow W. Everett, Jr. SCEEE Development Fund in cooperation with the Southeastern Association of Electrical Engineering Department Heads.

References

[1] Common vulnerability scoring system version 3.1, specification document, revision 1. [Online]. Available: `https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf`.

[2] Global privacy principles. [Online]. Available: `https://www.djeholdings.com/global-privacy-principles`, year=2019,.

[3] M Ugur Aksu, M Hadi Dilek, E İslam Tatlı, Kemal Bicakci, H Ibrahim Dirik, M Umut Demirezen, and Tayfun Aykır. A quantitative cvss-based cyber security risk assessment methodology for it systems. In *2017 International Carnahan Conference on Security Technology (ICCST)*, pages 1–8. IEEE, 2017.

[4] Y. Chang. Architecture design for performing grasp-and-lift tasks in brain–machine-interface-based human-in-the-loop robotic system. *IET Cyber-Physical Systems: Theory Applications*, 4(3):198–203, 2019.

[5] Cybersecurity and Infrastructure Security Agency (CISA). Security tip(st18-006). [Online]. Available: `https://us-cert.cisa.gov/ncas/tips/ST18-006`, year=2020,.

[6] Jeff Desjardins. How much data is generated each day? [Online]: `https://www.weforum.org/agenda/2019/04/how-much-data-is-generated-each-day-cf4bddf29f/`.

[7] Wenliang. Du. Computer security: A hands-on approach. [Online]. Available: `https://books.google.com/books?id=sp0JxAEACAAJ` [Accessed 23 July 2019], 2019.

[8] FBI. The morris worm. [Online]. Available: `https://www.fbi.gov/news/stories/morris-worm-30-years-since-first-major-attack-on-internet-110218` [Accessed 23 July 2019], 1988.

[9] Dinei Florencio and Cormac Herley. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web*, pages 657–666, 2007.

[10] S. Hsiao and D. Kao. The static analysis of WannaCry ransomware. In *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pages 1–1, 2018.

[11] Institute of Electrical and Electronics Engineers (IEEE). IEEE Code of Ethics. [Online]: `https://www.ieee.org/about/corporate/governance/p7-8.html`.

[12] Yier Jin and Yiorgos Makris. Hardware trojan detection using path delay fingerprint. In *2008 IEEE International workshop on hardware-oriented security and trust*, pages 51–57. IEEE, 2008.

[13] C. Konstantinou. Cyber-physical systems security education through hands-on lab exercises. *IEEE Design Test*, 37(6):47–55, 2020.

[14] Charalambos Konstantinou and Michail Maniatakos. Hardware-layer intelligence collection for smart grid embedded systems. *Journal of Hardware and Systems Security*, 3(2):132–146, 2019.

[15] A. Kulkarni, Y. Pino, and T. Mohsenin. Svm-based real-time hardware trojan detection for many-core platform. In *2016 17th International Symposium on Quality Electronic Design (ISQED)*, pages 362–367, 2016.

[16] Abraham Peedikayil Kuruvila, Ioannis Zografopoulos, Kanad Basu, and Charalambos Konstantinou. Hardware-assisted detection of firmware attacks in inverter-based cyberphysical microgrids. *arXiv preprint arXiv:2009.07691*, 2020.

[17] X. Liu, J. Ospina, and C. Konstantinou. Deep reinforcement learning for cybersecurity assessment of wind integrated power systems. *IEEE Access*, pages 1–1, 2020.

[18] Lynette I Millett and Stephen H Holden. Authentication and its privacy effects. *IEEE Internet Computing*, 7(6):54–58, 2003.

[19] National Security Agency. Cybersecurity education. [Online]: `https://www.nsa.gov/What-We-Do/Cybersecurity/Cybersecurity-Education/`.

[20] J. K. Naufal, J. B. Camargo, L. F. Vismari, J. R. de Almeida, C. Molina, R. I. R. González, R. Inam, and E. Fersman. A2cps: A vehicle-centric safety conceptual framework for autonomous transport systems. *IEEE Transactions on Intelligent Transportation Systems*, 19(6):1925–1939, 2018.

[21] Bernard Ngabonziza, Daniel Martin, Anna Bailey, Haehyun Cho, and Sarah Martin. Trustzone explained: Architectural features and use cases. In *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*, pages 445–451. IEEE, 2016.

[22] Aleph One. Smashing the stack for fun and profit. [Online]. Available: `http://www-inst.eecs.berkeley.edu/~cs161/fa08/papers/stack_smashing.pdf` [Accessed 23 July 2019], 2019.

[23] Juan Ospina, Ioannis Zografopoulos, XiaoRui Liu, and Charalambos Konstantinou. Trustworthy cyberphysical energy systems: Time-delay attacks in a real-time co-simulation environment. In *Proceedings of the 2020 Joint Workshop on CPS&IoT Security and Privacy*, pages 69–69, 2020.

[24] Charles P Pfleeger, Shari L Pfleeger, and Jonathan Margulies. *Security in computing: Fifth Edition*. Pearson Education, 2015.

[25] Masoud Rostami, Farinaz Koushanfar, and Ramesh Karri. A primer on hardware security: Models, methods, and metrics. *Proceedings of the IEEE*, 102(8):1283–1295, 2014.

[26] Jerome H Saltzer and Michael D Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, 1975.

[27] Y. Shiyanovskii, F. Wolff, C. Papachristou, D. Weyer, and W. Clay. Hardware trojan by hot carrier injection, 2009.

[28] Sergei Skorobogatov and Christopher Woods. Breakthrough silicon scanning discovers backdoor in military chip. [Online]. Available: `https://www.iacr.org/archive/ches2012/74280019/74280019.pdf` [Accessed 5 Nov. 2020].

[29] M. Tehranipoor and F. Koushanfar. A survey of hardware trojan taxonomy and detection. *IEEE Design Test of Computers*, 27(1):10–25, 2010.

[30] J. Wan, S. Tang, D. Li, M. Imran, C. Zhang, C. Liu, and Z. Pang. Reconfigurable smart factory for drug packing in healthcare industry 4.0. *IEEE Transactions on Industrial Informatics*, 15(1):507–516, 2019.

[31] Xueyang Wang, Charalambos Konstantinou, Michail Maniatakos, and Ramesh Karri. Confirm: Detecting firmware modifications in embedded systems using hardware performance counters. In *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 544–551. IEEE, 2015.

[32] Xueyang Wang, Charalambos Konstantinou, Michail Maniatakos, Ramesh Karri, Serena Lee, Patricia Robison, Paul Stergiou, and Steve Kim. Malicious firmware detection with hardware performance counters. *IEEE Transactions on Multi-Scale Computing Systems*, 2(3):160–173, 2016.

[33] Sql slammer. [Online]. Available: `https://en.wikipedia.org/wiki/SQL_Slammer` [Accessed 23 July 2019], 2003.

[34] Y. Zhang, M. Qiu, C. Tsai, M. M. Hassan, and A. Alamri. Health-cps: Healthcare cyber-physical system assisted by cloud and big data. *IEEE Systems Journal*, 11(1):88–95, 2017.

[35] I. Zografopoulos and C. Konstantinou. Derauth: A battery-based authentication scheme for distributed energy resources. In *2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 560–567, 2020.

[36] Ioannis Zografopoulos, Juan Ospina, and Charalambos Konstantinou. Harness the power of ders for secure communications in electric energy systems. *arXiv preprint arXiv:2009.06975*, 2020.

[37] Ioannis Zografopoulos, Juan Ospina, XiaoRui Liu, and Charalambos Konstantinou. Cyber-physical energy systems security: Threat modeling, risk assessment, resources, metrics, and case studies. 2021.