ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

UDC 004.312.466.056.5:004.415.53
doi: 10.32620/reks.2024.2.13

Artem TETSKYI[1], Artem PEREPELITSYN[1], Oleg ILLIASHENKO[1,2], Olga MOROZOVA[1], Dmytro UZUN[1]

[1] ***National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine***
[2] ***Institute of Informatics and Telematics of the National Research Council (IIT CNR), Pisa, Italy***

ENSURING CYBERSECURITY OF FPGA AS A SERVICE WITH THE USE OF PENETRATION TESTING OF COMPONENTS

The subject *of study in this article is modern penetration testing technologies, in which the test object is the implementation of a service based on a platform using Field Programmable Gate Array (FPGA) resources. The* ***goal*** *of this study is to improve modern methods of penetration testing of services provided by FPGA as a Service (FaaS) to find vulnerabilities for further fixing and increasing the level of services security and trust.* ***Task****: to analyze the technological capabilities for the development of FPGA as a Service; to analyze possible threats for FPGA as a Service platform; to analyze the structure of the FPGA as a Service platform and the peculiarities of attacks on it; to analyze options for using the penetration testing standard; to propose the classification of possible use of FPGA as a Service platform for solving of cybersecurity tasks; and to propose the sequence of critical components of ensuring of the cybersecurity of FPGA as a Service platform. The following* ***results*** *were obtained based on the tasks. The analysis of the capabilities of existing chips, FPGA accelerator cards, programming technologies, and the integrated environments of a leading company for creation of FPGA as a Service is performed. A study on the cybersecurity problems of FPGA as a Service platforms is conducted, and a set of components to ensure the cybersecurity of FPGA as a Service Platform is proposed. Modern cybersecurity threats of FPGA as a Service platforms are analyzed. A threat structure for FPGA as a Service is proposed. The possibility of applying a penetration testing standard to FPGA services is considered. Regular audits and penetration testing are crucial elements of a cybersecurity strategy and help maintain customer and user trust in FPGA services. Based on the analysis of the possible use of FPGA as a Service to solve cybersecurity tasks, a classification of five variants considering FPGA as an object and tool is proposed. The sequence of critical components of ensuring of the cybersecurity of FPGA as a Service platform is proposed to correspond to modern known threats. Complex activities, including the software updates, security monitoring, auditing, and penetration testing, based on security standards.* ***Conclusions.*** *The primary contribution and scientific novelty of the obtained results is the research into the possibilities of penetration testing for services, where the test object is a platform with access to FPGA. As in many other areas, ensuring the cybersecurity of FPGA as a Service platform is complex, and ignoring any component can lead to critical consequences. Applying only penetration testing is not enough; therefore, a comprehensive list of cybersecurity measures for FPGA as a Service platforms is provided, underlining the urgency and necessity of their implementation.*

Keywords: *FPGA; FPGA as a Service; penetration testing; cybersecurity ensuring; protection measures.*

Introduction

It is challenging to imagine a modern world without calculations. The amount of transferred, stored, and processed data is constantly growing, and the technologies are continually improving. FPGA is an energy-efficient and relatively fast solution to computing tasks. Ensuring cybersecurity is a crucial task for such a solution. The investigation of vulnerabilities can be performed using penetration testing of such systems [1].

FPGA as a Service is not just another cloud-based model, but a game-changer that revolutionizes how users can access FPGA resources [2]. This innovative platform allows customers to exploit the power of flexible and potent FPGA resources for specialized computing tasks such as data processing, machine learning, and cryptography. This can be achieved without physical installation and hardware maintenance. The production cybersecurity and use of such systems during the lifecycle are critical issues for critical systems [3].

FPGA services have opened new possibilities and transformed the operations of space agencies and private companies. Cloud platforms enable the design, testing, and optimize hardware systems, significantly reducing development time and costs [2, 4].

The real time adaptability of FPGAs with implementation of any functions is particularly valuable for dynamic tasks. In addition, cybersecurity must be addressed carefully for safety-critical FPGA-based industrial systems and critical IT infrastructures [5-7].

As with any resource on the World Wide Web, platforms that provide FPGA Services are not just potential targets. They are prime targets of attackers. The urgency of researching the cybersecurity problems of FaaS and identifying possible attacks and ways to protect them cannot be overstated [8]. This understanding is not just important but crucial to maintaining the integrity and security of FaaS services.

The goal of this study is to improve modern methods of penetration testing of services provided by FPGA as a Service to find vulnerabilities, fix them, and increase the level of security and trust in such services.

To achieve this goal, **the following tasks** were considered and solved:

1) to analyze the technological capabilities of FPGAs for the development of services;

2) to analyze possible threats using FPGA as a Service platform;

3) to analyze the structure of FPGA as a Service Platform and the peculiarities of attacks on it;

4) to analyze options for using the penetration testing standard [9];

5) to propose the classification of possible use of FPGA as a Service for solving of cybersecurity tasks;

6) to propose a sequence of critical components to ensure the cybersecurity of FPGA as a Service.

1. Analysis of technological capabilities for creation of FPGA as a Service

Considering the described features, implementing computational services and systems more effectively requires high-performance and energy-efficient solutions. Processor manufacturers add special registers and hardware blocks to their CPUs to support the basic calculations required to implement computationally intensive tasks and neural networks [10]. Supporting this type of computation at the hardware level improves performance. This is important because computational tasks are becoming a significant part of server tasks.

The use of FPGA technology for the hardware implementation of services and systems with PCI express access opens up the possibility of significant performance improvement [11]. This makes it possible to solve tasks efficiently using a service. However, it is only efficient with hardware acceleration due to the long computation duration of one dataset of computational tasks at the CPU. Most computational services are now based on CPUs or GPUs.

Despite this, unlike FPGAs, CPUs, GPUs, ASSPs, and ASICs do not provide the dynamics required for increasing computation performance. The advantage of FPGA implementation is the possibility of parallel execution of the same type of calculation. In addition, it is possible to use the pipelining of operations based on the instruction set of a specific project. This possibility of producing a dedicated accelerator for a particular task by the service allows FPGA technology to be considered as an efficient target hardware for implementing computationally intensive services [12].

The domain-specific architecture is required to build an efficient FPGA Service. It also requires the ability to exchange data quickly with the software and a set of software solutions that are directly responsible for interacting with the service user. This is valid for all computational services and solutions in the entire range of projects, ranging from Edge to Cloud.

In all FPGA implementations of computational services, a dedicated architecture is used considering the specific task. In addition, dynamic memory is required to store data and exchange results with a host application. To improve performance, all operations are represented as hardware-based logic.

Fast communication with FPGAs is possible over a PCI express interface. Alveo FPGA accelerator cards are powerful boards with a fast interface and sufficient dynamic memory to perform computations inside the chip. Grouping such FPGA accelerator cards in data centers allows the organization of FPGA service [13]. Providing access to such chips via reprogramming helps implement FPGA as a Service to the task [14].

For the programmer of such a service running on the host computer, each FPGA accelerator is accessible from the host application in C++ as the call of the function. The computational task can be divided into iterations for sequential computation using one FPGA accelerator or parallel execution of a few threads of FPGA accelerator cards connected to the host computer. Each enquired kernel execution in FPGA operation takes no longer than a few seconds. This is limitation of the communication framework. The Xilinx Runtime (XRT) framework simplifies communication with kernels over PCI Express [15]. At the same time, one host machine contains up to 8 slots with FPGA accelerator cards. All modules are connected to the host machine and can perform the same user task. They become super-logic machines that can be programmed to perform a single task in the entire data center. This type of acceleration can be used in most areas where data processing is required. The FPGA-based data processing efficiency is much higher than that of GPUs or general-purpose processors.

The performance advantage of an FPGA for AI applications is based on the ability to build its architecture inside FPGA chip. It helps to solve typical tasks much faster with the same or less power consumption. FPGA-based accelerator cards are specialized accelerators that significantly increase performance due to the parallel implementation of operations. This choice allows data centers to reduce their power consumption.

Creating hardware accelerators while building AI services is integral part for prototyping such systems.

The complexity of this process determines the duration and final cost of project.

Automated development environments, such as Vitis combined frameworks, for building AI systems can significantly reduce labor costs when designing and maintaining AI projects. The ability to use programming languages such as C++, OpenCL, and Python reduces the complexity of the design process. This involves developers with AI skills who require FPGA experience to create AI services. It is also possible to describe AI solutions directly at the hardware level using the TensorFlow and Caffe frameworks [16].

The Vitis development environment uses the AI optimizer, AI quantizer, and AI compiler in the development flow. An AI FPGA as a Service can be simplified using IP cores with customizable models frequently used in AI solutions [17].

Quantizing is the first step of AI project optimization. The most crucial process is quantizing by moving from a float-point representation of the system to a low-precision implementation based on the requirements of a particular subtask.

This step reduces the number of bits required to represent the neural network node. Development tools can automate this process. This optimization process makes one of the most significant contributions to increasing the density of parallel nodes in the model in FPGA implementation.

The following optimization step is to prune low-priority branches, which involves removing branches that have a low impact on the final result. It allows optimization of the descriptions of models for transfer to FPGA hardware implementations. This development promises a speedup of 5 to 50 times for FPGA implementation, which is a significant optimization in the amount of hardware resources per operation.

In the next optimization step, using a fixed-point representation of binary numbers reduces hardware complexity. It is essential to use data types that can be easily implemented using FPGA. In classical cases, the state represents a real value ranging from 0 to 1. The float data type, a single-precision floating-point data type consisting of 4 bytes, is used to represent this range.

To transfer the description of the neural network model to the hardware, a fixed-point representation rather than a floating-point representation is required.

It can be represented based on ranged integer data types with different numbers of bits and positions at the point. The exact number of bits depends on the allowed range of numbers of recognized levels. This transfer can be performed using level quantization based on the required accuracy. It can be performed manually during prototyping or with the use of quantization.

The last step of optimization is obtaining the final architecture. In this step, hardware solutions are optimized. The optimization of involved blocks and constructions with the previous steps allows us to obtain the final optimized architecture.

FPGA prototyping, language programming, and AI frameworks are employed. The proposed project can be implemented using languages that are not applicable to classical FPGA solutions. These supported languages and technologies for AI development using Vitis are OpenCL, C, C++, Python, TensorFlow, and Caffe. These three technologies were added directly to the Vitis development environment [16].

Most of these languages and technologies are suitable for implementing AI systems because most libraries for solving computational tasks are written in the same language. The languages must be used for FPGA projects based on hardware prototyping requirements.

If the project is described in one of these languages by the given formats and requirements, the output is an HLS description. This is the intermediate level that allows the project to be represented in SystemVerilog code. This output document is sufficiently hid. This is an internal project representation of the development environment itself.

This file was generated by Vivado during the project's compilation in Vitis. After generation, the generator will be compiled as usual for RTL projects.

Projects implemented using the mentioned languages usually achieve 50-60 percent of the resource utilization of the FPGA chip. This is of significant value because it does not require any specific knowledge for creation of projects in these languages.

The Deep Learning Processing Unit (DPU) is the basic block for processing frequently used algorithms in AI solutions. It is the basic equivalent of a dedicated processor unit. The DPU blocks support many popular neural networks, including VGG, ResNet, GoodLeNet, YOLO, SSD, MobileNet, and FNP.

This block processes graphic data, including real-time processing. Each DPU instance selects the necessary parameters during the development flow of each unit in the FPGA project.

After compilation, the FPGA project includes a set of kernels dedicated to accelerating the task. This must be considered because all algorithms run on an FPGA chip. The structure of all layers of the implementation of FPGA as a Service includes accelerator cards with FPGA chips connected to the host computer via a PCI express interface. The software component of the service runs on the host computer. The controller communicates with accelerator cards to line the task to FPGA and receive the results from kernels (see Figure 1). All software used to accelerate service runs at this layer.

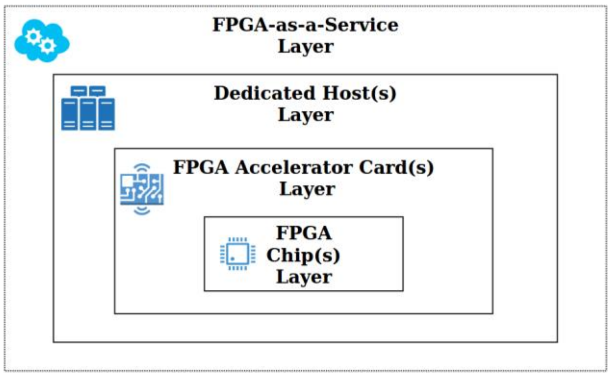

Fig. 1. Layers of implementation of FPGA as a Service

2. Possible threats analysis for FPGA as a Service platform

Due to the need for efficient, high-performance computing, modern computer architectures often combine Central Processing Units (CPUs), Graphics Processing Units (GPUs), and FPGAs.

However, each of these components is exposed to security and safety risks at the electrical level. The transition to heterogeneous systems, including the possibility of multi-user operation, requires an understanding and study of how the security vulnerabilities of individual components can affect a system as a whole [18].

The security of FPGA-type programmable logic integrated circuits is a crucial issue, as any vulnerability in the hardware can have severe consequences if used in secure designs. Since FPGA designs are encoded as bit streams, ensuring the security of the bit stream is critical. Attackers can have many motives to restore and manipulate the bitstream, including design cloning, intellectual property theft, design manipulation, and compromise (design subversions), e.g., through hardware Trojans. Given that FPGAs are often used in cyber-physical systems, such as in aerospace, medical, or industrial engineering, this can lead to negative physical consequences. As a result, manufacturers introduced the bit stream's encryption, ensuring its authenticity and confidentiality. Attacks against bitstream encryption have been proposed previously; however, such attacks require sophisticated hardware and considerable technical expertise to execute. A new low-cost example of attacks against Xilinx 7-Series (and Virtex-6) bitstream encryption, which resulted in complete loss of authenticity and privacy, was published [19]. This vulnerability is referred to Starbleed.

Another known vulnerability is Thangrycat, which was recently discovered in Cisco routers and allows attackers to compromise the router's trusted computing module. This leads to the possibility of the malware running undetected, which makes it virtually impossible for it to be removed once installed. Thangrycat exploits execution ability with system administrator rights. Red Balloon, a company that disclosed this vulnerability, also discovered a flaw that allows attackers to run code using administrator rights [20].

Research of configuration-based solutions is relevant when there is another severe threat to FPGA security called bitstream forgery. Bitstream tampering allows malicious modification of FPGA designs, which allows attackers to compromise the hardware root of trust and bypass any software-level security mechanisms. To address this issue, many vendors include various hardware blocks to ensure the trust and authenticity of bitstream files. However, over the past decade, many studies have demonstrated how to bypass these protections and exploit modified bitstreams to enable FPGA operations [21]. In a recent example of tampering with a real-world bitstream, a Cisco enterprise router was turned on by malicious bitstream modifications. This FPGA-based Cisco router attack affected an entire generation of Cisco products and could not be fixed without hardware update. It has been demonstrated that this attack can be carried out remotely without physical access to the target device. Attacks like Thangrycat are significant in FPGA security because they enable remote exploits. As FPGAs are increasingly deployed in networks such as Amazon EC2 F1 instances for FPGA cloud computing, FPGA security practices must move from a “trusted environment” model to one that remains vulnerable after the system is designed and deployed.

Modern FPGAs employ various passive protection techniques to protect the configuration. Modern FPGAs, such as the Xilinx 7-series and later device families, implement security features such as Hashed-Message-Authentication-Code (HMAC) tags and Advanced Encryption Standard (AES) bitstream encryption. These static defenses have been proven to be susceptible to reverse engineering and key extraction [22].

Security vulnerabilities, such as FPGA Trojans or malicious configuration logic embedded in the FPGA design, have been the focus of significant FPGA security research in Academia and Industry. Popular mitigation for hardware Trojans often involve identifying such malicious schemes. However, traditional Trojan detection approaches typically assume that an attacker cannot make targeted changes to the underlying configuration logic that can be used to detect potentially malicious activity. Thus, malicious changes to bitstream files can go undetected, leading to aggressive FPGA behavior.

In [23] presents the results of a comparison of commercially available solutions for building services with FPGA accelerators. This section discusses the advantages of the Xilinx platform and the tools used to create an FPGA service. The proposed model has stages for developing solutions based on FaaS. Some challenges related to FaaS are listed, and development trends are discussed. The SDAccel and Vitis platforms of Xilinx are considered, as is the possible role of these tools in creating an FPGA computing-as-service platform.

These results confirm the relevance of researching the possibilities of applying established penetration testing approaches, where the test object is a platform with access to FPGA resources. The identified features allow us to determine the most likely ways of violating confidentiality, integrity, or availability. Information about known vulnerabilities and threats specific to FPGAs should also be used. Appropriate keywords can be entered when searching the National Vulnerability Database [24]. Large organizations such as America's Cyber Defense Agency also publish helpful information about discovered vulnerabilities [25]. Some FPGA-specific attacks can be found in the Common Attack Pattern Enumerations and Classifications [26]. However, all the information from the above sources must be systematized for practical applications.

3. Analysis of structure and peculiarities of attacks on FPGA as a Service

The FPGA as a Service architecture considered in this study [27] consists of the layers shown in Figure 1. Based on this structure, it is possible to trace the possible threats to such layers of the system:

– FPGA as a Service – an online resource with a web interface where the project is uploaded;

– Dedicated Host – a dedicated machine (or several) with Ubuntu installed;

– FPGA Accelerator Card – a module connected to the motherboard (PCIe);

– FPGA Chip – one of the chips on the module.

The general structure of the interaction among the elements of the IoT system is presented in Figure 2.

The higher the system level, the easier it is to attack it. In the considered system, the upper level is the Web service; thus, the most significant attention will be paid to attacks that can be implemented at this level.

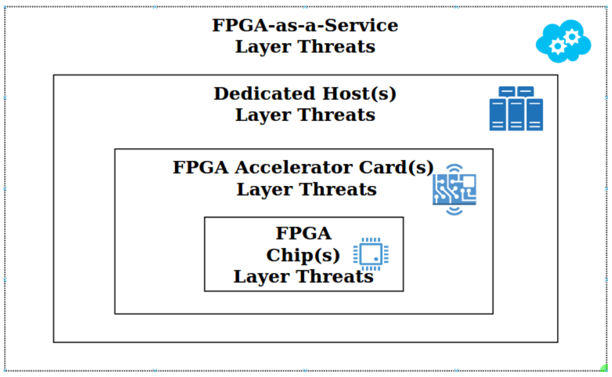

Fig. 2. Layers of threats for FPGA as a Service implementation

Attacks on FPGA service can be diverse due to the unique characteristics of FPGA and the general vulnerabilities of cloud and network services. It is worth highlighting several attacks and methods of their implementation:

1. Attacks on privacy. Because FPGAs in the commutation system process customer data, sensitive information is likely leaked, especially if the data are not encrypted or the system is not properly isolated. An attacker can attempt to intercept sensitive data on the FPGA.

2. Attacks on integrity. Attackers can attempt to change the data or algorithms running on the FPGA, leading to incorrect results or harmful effects of the computations. They can also gain access to the physical device's control functions. The FPGA configuration can be manipulated to achieve unauthorized or malicious purposes.

3. Attacks on availability. Distributed Denial-Of-Service (DDoS) attacks can be directed at the FPGA service, leading to its load and unavailability to clients. Like many other systems, FPGA services include two types of software and hardware vulnerabilities. Software Vulnerabilities – Vulnerabilities in the software used to configure and control the FPGA, such as outdated software versions, improper input data handling, and vulnerabilities in data transfer protocols.

Hardware-level vulnerabilities can include physical device security issues and vulnerabilities in the FPGA manufacturing process, such as flaws in circuitry or manufacturing processes that can be exploited to gain unauthorized access or damage to the device.

The search for vulnerabilities is a mandatory stage in critical application testing. With the development of technologies, the quality of the developed products also increases. In this way, the possibility of detecting a critical vulnerability in a developed product decreases. Then, in the “man-system” pair, the weakest link can be a person.

If exploiting known vulnerabilities in software or hardware related to the FPGA is impossible, attackers may employ social engineering techniques. Such methods include deceiving or manipulating personnel to gain access to confidential information. Accounts can be accessed through phishing, traffic interception, and other means.

It should also be mentioned that a dedicated machine with an installed FPGA module is part of the network infrastructure. Network infrastructure attacks can be carried out to redirect, modify, or block data, which can lead to disruption of the FPGA service. It is also possible to exploit vulnerabilities in software that is not directly related to the FPGA but is used on a dedicated machine.

4. Analysis of options for using the penetration testing standard

Penetration testing of FPGA service identifies and exploits vulnerabilities in systems using FPGAs to assess security. This includes analyzing both hardware and software aspects of the system. The main stages of penetration testing are as follows [28]:

1. Previous interaction. Determination of testing objectives, planning, and scope of testing. Establish test boundaries by defining test environments and system components. Customers may want to conduct testing during non-working hours when the load on the system is minimal, and possible failures will not interfere with the work of the customer service resource. Possible problems during testing were also recorded at this stage. This issue is essential because conducting automated testing increases the resource load, which can cause a denial of service. Options to get back up and run as soon as possible to minimize potential losses during downtime must also be developed.

2. Information gathering. In the case of black or gray box testing, any information related to the test object can be helpful. Port scanning is performed, software and its versions are determined, and information about known vulnerabilities in detected software versions is collected. If penetration testing involves social engineering techniques, information about individuals with administrative access rights is also collected. In future, this information can be used to create password dictionaries. An understanding of how an FPGA system is configured and the vulnerabilities associated with its architecture is advantageous. The network of the FPGA resource is also analyzed, and network vulnerabilities could be exploited.

3. Vulnerabilities identification. At this stage, automated tools and manual methods must be used to identify potential vulnerabilities. Here, FPGA settings and configurations must be checked for vulnerabilities. Maximum attention should be paid to the vulnerability at the upper level of the system – the web service.

4. Exploitation of vulnerabilities. Attempts are made to exploit the identified vulnerabilities to check for possible unauthorized access or malicious actions.

5. Post-exploitation. The primary goal of the post-exploitation stage is to develop an attack and search for new opportunities in the system. New data about the system can be obtained, and access to the system and network activity can be gained for further analysis.

6. Reporting. The last crucial step is to document the results. The report contains a description of each stage's results. The information is provided to the stakeholders and includes details of identified vulnerabilities, methods of exploiting, and remediation recommendations.

A possible next step is to retest the identified issues and verify that the vulnerabilities have been adequately addressed. One of the options for using publicly available knowledge bases on software and hardware weaknesses is the analysis of the CWE database [29]. It is possible to adapt known weaknesses in the context of the security analysis of platforms that provide FPGA as a Service. The result of such adaptation can be the construction of a penetration testing plan that can cover vulnerabilities in both the service itself and specific FPGA projects uploaded to the service and executed by relevant infrastructure nodes.

At the same time, penetration testing tools can be selected with the help of appropriate recommendation services, particularly those that use artificial intelligence [30].

A detailed penetration testing plan for FPGA as a Service should include testing of aspects specific to the technology being used. When dealing with hardware problems, it makes sense to rely on existing software and hardware vulnerability databases.

MITRE's CWE is an example of such a base, where specific hardware vulnerabilities can be taken from the CWE-1194 representation. This is a list of possible hardware vulnerabilities such as privilege separation and access control issues, memory and storage issues, and cryptographic issues. This knowledge should be used during the creation of a penetration testing plan for FPGA as a Service platforms.

Due to the specific hardware characteristics, FPGAs have unique characteristics that may require specialized knowledge and approaches. The examination of such features may require physical access to the device.

5. Proposed classification of the possible use of FPGA as a Service for ensuring of cybersecurity

An analysis of the cybersecurity issues of FPGA as a Service solutions reveals that they can be considered from different perspectives.

Computing nodes to solve computational tasks is a possible form of an FPGA-based service that can be used in cybersecurity analysis; this requires significant computing power. Such services provide the possibility of commuting explicitly for user tasks [17].

In this case, FPGA-based service is a tool for solving a specialized problem. A task can be the differential analysis of a dataset containing the specific information about the instance of the observing system under consideration. It allows significantly faster processing using computing resources and performs better than other CPU- or GPU-based solutions and systems [2, 23].

Implementing complex crypto primitives is another possible example of using FPGA**.** It provides the ability to produce calculation results for each clock cycle using pipelining at the hardware level [2]. Such calculations can check the hash function's cryptographic strength and search for specific hash values using given parameters. The ability to implement the architecture for a specific task allows parallelization if the datasets do not have data dependency. Pipelining also ensures continuous operation of each hardware component, generating one computational result in each clock cycle. At the same time, the comparison conditions and computation operations can be straightforward, which ensures a compact implementation and a performance increase of ten times compared to CPU and GPU implementations [14, 17].

The physical implementation of true random number generators and physically unclonable functions (PUFs) is another important and unique direction in which FPGA can be used as a service for cybersecurity and related tasks [27, 31]. Such solutions can be used to identify specific instances of a project and ensure the possibility of organizing the copyright protection of such a project [32].

The use of these physical implementation features prevents the emulation or simulation of instances of such FPGA projects. Thus makes it possible to ensure cybersecurity components at the implementation level. It can be used not only as an independent project but also as an integral part of FPGA as a service to provide the ability to control the execution of individual instances [33].

The use of physical implementations and features of true random generation [34, 35] and unclonability significantly simplify the digital rights management task for FPGA services [31, 33]. Such cybersecurity components allow the physical implementation of FPGA systems to be considered a tool and research object.

The production of new FPGA accelerator cards, both for prototyping and regular use, is also where the possibility of considering FPGA as an object of investigation for penetration is significant and valuable. Such FPGA accelerator cards include a built-in set of components to simplify interactions with the host computer, especially for XRT-managed kernels [23, 36] for AMD-based FPGAs.

Such shell components contain a firmware program from the manufacturer for communicating with FPGA over PCI Express [37]. For communication, the firmware version must be compatible with the exact version of the XRT framework. The code is not available to FPGA programmers. Only the vendor knows what is contained in this shell firmware version. Therefore, investigating and analyzing the work of such firmware components is required to create FPGA as a service while creating critical systems.

Such firmware also includes a firewall to track failures and protect the FPGA accelerator card. There is an interface to read the card status; however, the complete list of events affecting the firmware status is unavailable in the public specification. Investigating the FPGA as an object allows testing the probable behavior of such firmware for each new version of the shell associated with the XRT framework version.

To find defects and vulnerabilities in the firmware programs from the manufacturer for each specific FPGA accelerator card and each particular version of XRT, specialized projects can be created and used for investigation using the proposed penetration testing steps. These solutions are exactly FPGA as a service because they directly use XRT-managed kernels. Such solutions consider FPGAs as an object for investigation and attempt to apply penetration testing methods for service components during the manufacturer's preparation of firmware to identify and eliminate vulnerabilities [38] during the production stage.

Implementing FPGA-based AI services is another essential and continuously developing area of security and vulnerability research, which can be more correctly referred to as imitation of intelligence.

Such services can also be considered as tools for performing cryptanalysis tasks to reduce complexity and required efforts and to investigate the reliability of other systems [39].

However, they can also be considered as objects of investigation. Combined with implementing specialized AI tasks using FPGAs, such services can significantly increase data processing speed for specific investigations, such as data processing during differential analysis.

Creating a service for a specific task and using the capabilities of development environments allows prototyping using specialized tools, languages, and frameworks (Python, Caffe, and TensorFlow), which reduces labor costs while analyzing such data [2, 17].

When implementation is the object of the research, the proposed penetration tests on individual components of an FPGA as a Service can be applied to investigate and eliminate potential vulnerabilities in the next step.

The comparison of the proposed categories of suggested steps of investigation with penetration tests allows us to conclude how to apply the proposed sequence to the target object. The comparison performed from the point of view of the possibility of using FPGA as a service, like the object of investigation (see Table 1), allows us to conclude that the first two variants consider FPGA only as a tool to increase computation performance.

Table 1

Comparison of the proposed classification categories from the point of view of the possibility of using an FPGA as a Service similar to the object of investigation

The purpose of FPGA in FPGA as a Service for solving of cybersecurity tasks	FPGA as an object	FPGA as a tool
Implementation of computing node for acceleration of computational intensive tasks	No	Yes
Implementation of crypto primitives and hash functions with pipelining	No	Yes
Ensuring tasks of DRM with the use of physical implementation associated with FPGA instance	Yes	Yes
The search for vulnerabilities in the FPGA firmware of the shell	Yes	Yes
Implementation of Artificial Intelligence services based on FPGA	Yes	Yes

The difference is that in the first scenario, the end user prepares the entire project for FPGA as a service for computation. Still, in the case of implementing crypto primitives, the end user can use the service with already prepared and well-tested FPGA projects to accelerate the set of prepared functions for computation with the user's datasets.

The last three options consider FPGA as a Service both the object of investigation and tool for implementing practically essential tasks. The ensuring digital rights management (DRM) variant uses the specifics of the physical implementation instance.

Here, the features of FPGA as the object, like TRNG and PUF, allow the identification of the instance and the exact instance of the service. They also allow control of the execution of licensed IP-core instances that use such security components.

The investigation of the FPGA shell is performed when the project is loaded into the FPGA, which is the tool used to investigate the FPGA itself as an object. This analysis direction is critically important for enabling the use of new firmware in FPGA services for critical systems.

Implementing AI with FPGA as a Service allows both the implementation of a powerful project of the service with hardware acceleration for AI computations and the use of such a solution during the investigation of the vulnerabilities of FPGA as an object.

The third and last variants are the most interesting directions for research and practical use because they use hardware features of FPGA implementations that are not available for CPU-based implementations of such services.

6. Proposed sequence for ensuring the FPGA as a Service cybersecurity

As in many other areas, ensuring the cybersecurity of FaaS platforms is complex, and ignoring any component can lead to critical consequences. It is worth highlighting the following elements of cybersecurity:

1. It is essential to regularly update FPGA software and firmware to address known vulnerabilities. Regular updates of FPGA software and firmware are critical to maintaining the cybersecurity and performance of these devices because they help address known vulnerabilities, improve functionality, and prevent potential cyberattacks [40]. In FPGAs, where devices are often used to process sensitive data and perform critical tasks, outdated software or firmware can become weak links, exposing the system to the risk of unauthorized access or tampering. Updates often include patches to address security vulnerabilities discovered after the release of previous versions and may also include performance improvements that improve device reliability and efficiency [41]. In addition, with the ever-evolving cybersecurity threat landscape, regular updates ensure adaptation to new attack methods, which prevent potential threats before they cause harm. In addition to being out-of-date systems prone to vulnerabilities, they may need to meet security standards and regulatory requirements, leading to legal and reputational risks for organizations. Thus, regularly updating FPGA software and firmware is an integral part of the strategy to ensure the cybersecurity and reliability of critical systems.

2. Continuous monitoring of network traffic and analysis of system behavior can help identify attempts to exploit vulnerabilities. Network traffic monitoring in FPGA services is a critical security system component that continuously monitors and analyzes data transmitted over the network. It enables the rapid detection of anomalies, suspicious activity, or unauthorized access attempts, which is key to preventing and minimizing potential cyber-attacks. Effective network traffic monitoring involves using specialized tools and software capable of analyzing large volumes of data in real-time and identifying unusual patterns of data transmission that may indicate system compromise. This approach detects active attacks and predicts potential threats based on analyzing traffic behavior patterns. The importance of network traffic monitoring is increasing significantly in the context of FPGA services, where attackers can exploit systems with high performance and flexibility to spread malicious attacks or obtain valuable data quickly. Therefore, the continuous monitoring and analysis of network traffic is an integral part of an overall cybersecurity strategy that is intended to protect valuable information and maintain the stable operation of FPGA services.

3. Regular security audits and penetration testing can identify potential vulnerabilities before exploiting them. Security auditing and penetration testing play an essential role in ensuring the security of FPGA services because they enable the identification [42] and remediation of potential vulnerabilities before attackers exploit them. These procedures include a comprehensive evaluation of the hardware and software aspects of FPGA, including configuration analysis, code verification, network security assessment, and device security [43]. Experienced professionals typically perform security audits to identify weaknesses in a system, whereas penetration testing focuses on actively trying to break into a system to identify vulnerabilities. These techniques help protect against external threats and prevent internal threats such as software bugs and insecure configurations. Regular auditing and penetration testing is crucial elements of a cybersecurity strategy, and they help maintain the trust of customers and users in FPGA services. In addition, these procedures allow organizations to meet regulatory and legislative requirements, which is important in industries with high cybersecurity and safety standards.

4. Following security standards and best practices can help prevent many attacks. Adherence to security standards when creating FPGA services is critical to guaranteeing reliability and security, especially given their widespread use in sensitive areas, such as telecommunications, defense, and medicine. Security standards such as ISO 27001 [44], NIST SP 800-53 [45], and the 15408 Common Criteria series [46], provide frameworks and guidelines for development that help identify, manage, and minimize risks associated with cybersecurity. Protecting against external and internal threats such as unauthorized access, data manipulation, and hardware attacks. Applying these standards to developing and operating FPGAs helps ensure data confidentiality, integrity, and availability and maintains user confidence in the system. In addition, compliance with international standards increases the product's competitiveness in the market and ensures high safety and quality. In addition, adherence to standards helps meet legal and regulatory requirements, which is especially important in industries with stringent safety requirements.

5. Raising staff awareness of current attack patterns is a key protection element. Staff training on attack methods against FPGA services plays a vital role in cybersecurity because employees who are informed about potential threats and attack methods can effectively prevent, detect, and respond to incidents. FPGAs, which are used in industries ranging from telecommunications to the military, are complex and flexible systems that are subject to unique threats, including hardware-level attacks and the exploitation of software vulnerabilities. Carefully trained personnel who understand how such attacks work and what security measures are necessary to prevent them are an essential element of the defense system. This helps minimize the risk of successful cyberattacks and strengthens an organization’s overall cybersecurity strategy, increasing its resilience against potential threats. Training promotes a culture of security among employees, which is critical for maintaining high levels of protection against evolving cyber threats.

7. Discussion

A study on the cybersecurity problems of FaaS platforms was conducted, and a set of components was proposed to ensure their cybersecurity. An analysis of the current cybersecurity threats of FaaS platforms was also conducted. The penetration testing standard for FPGA services was considered. Regular auditing and penetration testing is crucial elements of a cybersecurity strategy and help maintain customer and user trust in FPGA services.

A set of components corresponding to modern threats is proposed to ensure the cybersecurity of FaaS platforms. The proposed complex includes regular software updates, security monitoring and analysis, audit and penetration testing, compliance with security standards, and staff training. The quantitative indicators of the evaluation of such measures can be used to determine the number of identified vulnerabilities and their criticality, which are determined using the Common Vulnerability Scoring System.

The scientific novelty of the obtained results lies in the fact that a study of the possibilities of penetration testing was conducted, where the object of testing was a platform with access to FPGA. As in many other areas, ensuring the cybersecurity of FaaS is a complex task, where ignoring any component can lead to critical consequences. Application of penetration testing is not enough, so a comprehensive list of measures to ensure the cybersecurity of FaaS platforms is provided.

Conclusions

The practical significance of this study is that it proposes a comprehensive cybersecurity strategy, including regular audits, penetration testing, and other measures, to identify and mitigate vulnerabilities in FPGA as a Service platforms, thereby enhancing their security and trustworthiness.

A study on the cybersecurity problems of FPGA asService platforms was conducted, and a set of components was proposed to ensure their cybersecurity. An analysis of the current cybersecurity threats of FPGA as a Service platforms was performed. The penetration testing standard for FPGA services was considered.

Regular auditing and penetration testing are key elements of a cybersecurity strategy and help maintain customer and user trust in FPGA services.

A set of components corresponding to modern threats is proposed to ensure the cybersecurity of FPGA-based Service platforms. The proposed complex includes regular software updates, security monitoring and analysis, audit and penetration testing, compliance with security standards, and staff training.

The scientific novelty of the obtained results is that a study of the possibilities of penetration testing was conducted with a platform with access to FPGA resources as the object of testing. This allows to identify an instance of FPGA chip in the service to ensure DRM.

The cybersecurity of FPGA as a Service platforms is complex, and ignoring any component can lead to critical consequences. Applying only penetration testing is insufficient; therefore, a comprehensive list of measures for FPGA as a Service platforms is provided.

Further research will be dedicated to the following directions: (i) investigation and development of advanced penetration testing techniques explicitly tailored for FPGA as a Service platforms, including automated tools and frameworks that can identify complex and hidden vulnerabilities; (ii) exploring the application of AI and ML for real-time cyber threat intelligence in FPGA as a Service platforms, focusing on anomaly detection, predictive analytics, and automated response systems; (iii) development of interoperability and compliance frameworks that ensure FPGA as a Service platforms can seamlessly integrate with existing cybersecurity infrastructures and adhere to evolving regulatory requirements, thereby enhancing overall security posture and compliance.

Contribution of authors: goal and tasks formulation, analysis of publications, analysis of penetration testing and proposing of a sequence of steps for FPGA, draft writing, preparing of references – **Artem Tetskyi**; analysis of publications, tasks formulation, analysis of modern FPGA technology, proposing of classification, draft writing, references preparing, translation, overall review and editing, validation – **Artem Perepelitsyn**; concept formulation, publications analysis, translation, formulation of conclusions, references preparing, overall review and editing, validation – **Oleg Illiashenko**; concept formulation, validation – **Olga Morozova**; search of references with database of vulnerabilities, review and editing – **Dmytro Uzun**.

Conflict of interest

The authors declare that they have no conflict of interest about this research, whether financial, personal, authorship, or otherwise, that could affect the research and its results presented in this paper.

Acknowledgments

This article was prepared with the support of the project “Methods and case technologies of evidence-based cybersecurity assessment of programmable systems to ensure the protection of critical IT infrastructure”, financed by the Ministry of Education and Science of Ukraine, No. 0123U102106. This publication reflects the views of the authors only, and the Ministry of Education and Science of Ukraine cannot be held responsible for any use of the information contained therein.

The authors express their gratitude to colleagues from the Department of Computer Systems, Networks and Cybersecurity of National Aerospace University “Kharkiv Aviation Institute”.

Data availability

The manuscript contains no relevant data.

Use of Artificial Intelligence

The authors confirm that they did not use artificial intelligence methods in their work.

All authors have read and agreed to the publication of the finale version of this manuscript.

References

1. Tetskyi, A. Testuvannia na pronyknennia komponentiv FPGA yak servisu dlia zabezpechennia kiberbezpeky [Penetration testing of FPGA as a Service components for ensuring cybersecurity]. *Aviacijno-kosmicna tehnika i tehnologia – Aerospace technic and technology*, 2023, no. 6, pp. 95-101. DOI: 10.32620/aktt.2023.6.11. (In Ukrainian).
2. Perepelitsyn, A. Method of creation of FPGA based implementation of Artificial Intelligence as a Service. *Radioelectronic and Computer Systems*, 2023, no. 3, pp. 27-26. DOI: 10.32620/reks.2023.3.03.
3. Illiashenko, O., Kharchenko, V., & Kovalenko, A. Cyber security lifecycle and assessment technique for FPGA-based I&C systems. *Proceedings of IEEE East-West Design & Test Symposium (EWDTS 2012)*, Kharkiv, Ukraine, 2012, pp. 432-436. DOI: 10.1109/EWDTS.2013.6673155.
4. Tsai, W. C. Field-Programmable Gate Array-Based Implementation of Zero-Trust Stream Data Encryption for Enabling 6G-Narrowband Internet of Things Massive Device Access. *Sensors*, 2024, vol. 24, no. 3, article no. 853, pp. 1-22. DOI: 10.3390/s24030853.
5. Illiashenko, O., Kharchenko, V., & Odarushchenko, O. Towards Evidence-Based Cybersecurity Assessment of Programmable Systems to Ensure the Protection of Critical IT Infrastructure. *Proceedings of the 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Dortmund, Germany, 2023, pp. 1178-1183, DOI: 10.1109/IDAACS58523.2023.10348834.

6. Kharchenko, V., Illiashenko, O., Brezhnev, E., Boyarchuk, A., & Golovanevskiy, V. Security informed safety assessment of industrial FPGA-based systems. *Proceedings of the Probabilistic Safety Assessment and Management Conference (PSAM 12),* Honolulu, Hawaii, 2014, pp. 1-11.

7. Kharchenko, V., Illiashenko, O. Sklyar, V. Invariant-Based Safety Assessment of FPGA Projects: Conception and Technique. *Computers*, 2021, 10, no. 10: 125, pp. 1-10. DOI: 10.3390/computers10100125.

8. Tsantikidou, K., & Sklavos, N. Threats, Attacks, and Cryptography Frameworks of Cybersecurity in Critical Infrastructures. *Cryptography*, 2024, vol. 8, no. 1, article no. 7, pp. 1-24. DOI: 10.3390/cryptography8010007.

9. *The Penetration Testing Execution Standard*. Available at: http://www.pentest-standard.org/ (accessed August 22, 2023).

10. Vincy Davis. *Intel's 10th gen 10nm 'Ice Lake' processor offers AI apps, new graphics and best connectivity*. Available at: https://hub.packtpub.com/intels-10th-gen-10nm-ice-lake-processor-offers-ai-apps-new-graphics-and-best-connectivity/ (accessed May 22, 2024).

11. Yoon, Y. H., Hwang, D. H., Yang, J. H., & Lee, S. E. Intellino: Processor for embedded artificial intelligence. *Electronics*, 2020, vol. 9, no. 7, article no. 1169, pp. 1-12. DOI: 10.3390/electronics9071169.

12. *UltraFast Design Methodology Guide for Xilinx FPGAs and SoCs, Xilinx, UG949 (v2021.2).* Available at: https://docs.xilinx.com/r/2021.2-English/ug949-vivado-design-methodology/SLR-Utilization-Considerations. (accessed February 28, 2023).

13. Yanovskaya, O., Yanovsky, M.., & Kharchenko, V. The concept of green Cloud infrastructure based on distributed computing and hardware accelerator within FPGA as a Service. *Proceedings of IEEE East-West Design & Test Symposium (EWDTS 2014), Kiev, Ukraine*, 2014, pp. 1-4, DOI: 10.1109/EWDTS.2014.7027089.

14. Perepelitsyn, A., Zarizenko, I., & Kulanov, V. FPGA as a Service Solutions Development Strategy. *Proceedings 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies DESSERT 2020*, 2020, pp. 376-380, DOI: 10.1109/DESSERT50317.2020.9125017.

15. *Vitis Unified Software Platform Documentation: Application Acceleration Development, Xilinx, UG1393 (v2022.2).* Available at: https://docs.xilinx.com /r/en-US/ug1393-vitis-application-acceleration/Getting-Started-with-Vitis. (accessed December 07, 2022).

16. *AI Engine Kernel and Graph Programming Guide, Xilinx, UG1079 (v2022.2).* Available at: https://docs.xilinx.com/r/en-US/ug1079-ai-engine-kernel-coding/Overview. (accessed October 19, 2022).

17. Perepelitsyn, A., Fesenko, H., Kasapien, Y., & Kharchenko, V. Technological Stack for Implementation of AI as a Service based on Hardware Accelerators. *Proceedings 2022 IEEE 12th International Conference on Dependable Systems, Services and Technologies, DESSERT 2022,* 2022. 5 p. DOI: 10.1109/DESSERT58054.2022.10018615.

18. Mahmoud, D. G., Lenders, V., & Stojilović, M. Electrical-Level Attacks on CPUs, FPGAs, and GPUs: Survey and Implications in the Heterogeneous Era. *ACM Computing Surveys*, 2022, vol. 55, no. 3, article no. 58, pp. 1-40. DOI: 10.1145/3498337.

19. Ender, M., Moradi, A., & Paar, C. The unpatchable silicon: a full break of the bitstream encryption of xilinx 7-series FPGAs. *Proceedings of 29th USENIX Conference on Security Symposium (SEC'20)*, 2020, article no. 102, pp. 1803-1819. DOI: 10.5555/3489212.3489314.

20. Red Balloon Security. *100 Seconds of Solitude: Defeating Cisco Trust Anchor With FPGA Bitstream Shenanigans*. Available at: https://redballoonsecurity. com/files/CycIhULVL5FS6VNM/100_seconds_of_solitude.pdf (accessed August 22, 2023).

21. Chakraborty, R. S., Saha, I., Palchaudhuri, A., & Naik, G. K. Hardware Trojan Insertion by Direct Modification of FPGA Configuration Bitstream. *IEEE Design & Test*, 2013, vol. 30, no. 2, pp. 45-54. DOI: 10.1109/MDT.2013.2247460.

22. Lohrke, H., Tajik, S., Krachenfels, T., Boit, C., & Seifert, J.-P. Key Extraction Using Thermal Laser Stimulation: A Case Study on Xilinx Ultrascale FPGAs. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018, vol. 2018, no. 3, pp. 573-595. DOI: 10.13154/tches.v2018.i3.573-595.

23. Perepelitsyn, A. & Kulanov, V. Technologies of FPGA-based projects Development Under Ever-changing Conditions, Platform Constraints, and Time-to-Market Pressure. *Proceedings 2022 IEEE 12th International Conference on Dependable Systems, Services and Technologies, DESSERT 2022*, 2022, pp. 1-5, DOI: 10.1109/DESSERT58054.2022.10018828.

24. National Vulnerability Database. *Search results for keyword FPGA*. Available at: https://nvd.nist.gov/vuln/search/results?form_type=Basic&results_type=overview&query=FPGA&search_type=all&isCpeNameSearch=false (accessed March 22, 2024).

25. America's Cyber Defense Agency. *Sielco Poly-Eco FM Transmitter.* Available at: https://www.cisa.gov/news-events/ics-advisories/icsa-23-299-07 (accessed March 22, 2024).

26. Common Attack Pattern Enumeration and Classification. *CAPEC-674: Design for FPGA Maliciously Altered.* Available at: https://capec.mitre.org/ data/definitions/674.html (accessed March 22, 2024).

27. Perepelitsyn, A., & Kulanov, V. Analysis of Ways of Digital Rights Management for FPGA-as-a-Service for AI-Based Solutions. *Proceedings 2023 IEEE 13th International Conference on Dependable Systems, Services and Technologies, DESSERT 2023*, 2023. 5 p. Accepted.

28. *Penetration Testing Methodologies.* Available at: https://owasp.org/www-project-web-security-testing-guide/latest/3-The_OWASP_Testing_Framework/1-Penetration_Testing_Methodologies (accessed March 22, 2024).

29. *Common Weakness Enumeration*. Available at: https://cwe.mitre.org/ (accessed March 22, 2024).

30. Tetskyi, A., Kharchenko, V., Uzun, D., & Nechausov, A. Architecture and Model of Neural Network Based Service for Choice of the Penetration Testing Tools. *International Journal of Computing*, 2021, vol. 20, no. 4, pp. 513-518. DOI: 10.47839/ijc.20.4.2438.

31. Tan, T. H., Ooi, C. Y., & Marsono, M. N. drDRM: A PUF-Based Dynamically Reconfigurable DRM Mechanism for FPGA-Based Platform. *2018 Sixth International Symposium on Computing and Networking, CANDAR 2018,* 2018. pp. 194-200. DOI: 10.1109/CANDAR.2018.00034.

32. *FPGA-Centric Software Acceleration Made Easy*. Available at: https://www.accelize.com/blog-fpga-centric-software-acceleration-made-easy (accessed March 23, 2022)

33. Perepelitsyn, A. Zabezpechennya upravlinnya tsyfrovymy pravamy dlya stvorennya shtuchnoho intelektu yak servisu na osnovi FPGA realizatsiyi [Ensuring of Digital Rights Management of FPGA based implementation of Artificial Intelligence as a Service]. *Aviacijno-kosmicna tehnika i tehnologia – Aerospace technic and technology*, 2023, no. 6, pp. 102–110. DOI: 10.32620/aktt.2023.6.12. (In Ukrainian).

34. Ahmed, M. K., Saha, S. K. & Bobda, C. Trusted IP Solution in Multi-tenant Cloud FPGA Platform. *Proceedings of 2022 IEEE 8th World Forum on Internet of Things (WF-IoT),* 2022, pp. 1-6. DOI: 10.1109/WF-IoT54382.2022.10152167.

35. Dridi, F., El Assad, S., El Hadj Youssef, W., Machhout, M., & Lozi, R. The design and FPGA-based implementation of a stream cipher based on a secure chaotic generator. *Applied Sciences,* 2021, vol. 11, no. 2, article no. 625, pp. 1-19. DOI: 10.3390/app11020625.

36. *XRT Controlled Kernel Execution Models*. Available: https://xilinx.github.io/XRT/master/html/xrt_kernel_executions.html. (accessed October 7, 2022).

37. *Alveo U280 Data Center Accelerator Card User Guide, Xilinx, UG1314 (v1.3).* Available at: https://www.sandycast.com/support/documentation/boards_and_kits/accelerator-cards/ug1314-u280-reconfig-accel.pdf. (accessed February 27, 2020).

38. Aljuffri, A., Huang, R., Muntenaar, L., Gaydadjiev, G., Ma, K., Hamdioui, S., & Taouil, M. The Security Evaluation of an Efficient Lightweight AES Accelerator. *Cryptography,* 2024, vol. 8, no. 2, article no. 24, pp. 1-20. DOI: 10.3390/cryptography8020024.

39. Azar, K. Z., Hossain, M. M., Vafaei, A., Al Shaikh, H., Mondol, N. N., Rahman, F., Tehranipoor, M., & Farahmandi, F. Fuzz, penetration, and AI testing for SoC security verification: Challenges and solutions. *Cryptology ePrint Archive,* 2022, article no. 394, pp. 1-22. Available at: https://eprint.iacr.org/2022/394.pdf (accessed March 22, 2024).

40. Potestad-Ordóñez, F. E., Casado-Galán, A., & Tena-Sánchez, E. Protecting FPGA-Based Cryptohardware Implementations from Fault Attacks Using ADCs. *Sensors,* 2024, vol. 24, no. 5, article no. 1598, pp. 1-15. DOI: 10.3390/s24051598.

41. Proulx, A., Chouinard, J. Y., Fortier, P., & Miled, A. A survey on FPGA cybersecurity design strategies. *ACM Transactions on Reconfigurable Technology and Systems,* 2023, vol. 16, no. 2, article no. 20, pp. 1-33. DOI: 10.1145/3561515.

42. Al-Shaikh, H., Vafaei, A., Rahman, M. M. M., Azar, K. Z., Rahman, F., Farahmandi, F., & Tehranipoor, M. SHarPen: SoC Security Verification by Hardware Penetration Test. *Proceedings of the 28th Asia and South Pacific Design Automation Conference,* 2023, pp. 579-584. DOI: 10.1145/3566097.3567918.

43. Kharchenko, V., & Ivasiuk, O. Vykorystannia metodu veryfikatsii FMEDA/FIT dlia otsiniuvannia kiberbezpeky prohramovnoho lohichnoho kontrolera [Using the FMEDA/FIT verification method to assess the cybersecurity of a programmatic logic controller]. *Systemy upravlinnia, navihatsii ta zviazku – Control, Navigation and Communication Systems,* 2023, no. 4, pp. 114–119. DOI: 10.26906/SUNZ.2023.4.114. (In Ukrainian).

44. *ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection – Information security management systems – Requirements.* Available at: https://www.iso.org/standard/27001 (accessed March 22, 2024).

45. *NIST SP 800-53 Rev. 5 Security and Privacy Controls for Information Systems and Organizations.* Available at: https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final (accessed March 22, 2024).

46. *ISO/IEC 15408-1:2022 Information security, cybersecurity and privacy protection — Evaluation criteria for IT security.* Available at: https://www.iso.org/standard/72891.html (accessed March 22, 2024).

Received 22.03.2024, Accepted 15.04.2024

ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ FPGA ЯК СЕРВІСУ З ВИКОРИСТАННЯМ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ ЙОГО КОМПОНЕНТІВ

А. Г. Тецький, А. Є. Перепелицин, О. О. Ілляшенко, О. І. Морозова, Д. Д. Узун

Предметом дослідження в даній статті є сучасні технології тестування на проникнення, в яких об’єктом тестування є реалізований сервіс на основі платформи, що використовує ресурси FPGA. **Метою** даної роботи є вдосконалення сучасних методів тестування на проникнення сервісів, що надають послугу FPGA as a Service, для пошуку вразливостей для подальшого їх виправлення та підвищення рівня безпеки та довіри до сервісів. **Завдання:** проаналізувати технологічні можливості для розробки FPGA as a Service; проаналізувати можливі загрози для FPGA як сервісної платформи; проаналізувати структуру платформи

FPGA as a Service та особливості атак на неї; проаналізувати варіанти використання стандарту тестування на проникнення; запропонувати класифікацію можливого використання FPGA як сервісної платформи для вирішення завдань кібербезпеки; запропонувати послідовність критичних компонентів для забезпечення кібербезпеки FPGA як сервісної платформи. За результатами поставлених завдань отримано наступні **результати**. Проведено аналіз можливостей існуючих чіпів, карт прискорювачів FPGA, технологій програмування та інтегрованих середовищ провідної компанії, що надає FPGA як послугу. Проведено дослідження проблем кібербезпеки платформ FPGA as a Service та запропоновано набір компонентів для забезпечення кібербезпеки платформ FPGA as a Service. Проаналізовано сучасні загрози кібербезпеці платформ FPGA as a Service. Запропоновано структуру загроз для FPGA as a Service. Розглядається можливість застосування стандарту тестування на проникнення до сервісів FPGA. Регулярні аудити та тестування на проникнення є ключовими елементами стратегії кібербезпеки та допомагають зберегти довіру клієнтів і користувачів до послуг FPGA. На основі проведеного аналізу можливого використання FPGA як сервісу для вирішення завдань кібербезпеки запропоновано класифікацію п’яти варіантів, що розглядають FPGA як об’єкт та інструмент. Запропоновано послідовність критичних компонентів для забезпечення кібербезпеки платформи FPGA as a Service, яка відповідає сучасним відомим загрозам. Пропонуються комплексні дії, включаючи оновлення програмного забезпечення, моніторинг безпеки, аудит і тестування на проникнення на основі стандартів безпеки. **Висновки.** Основний внесок і наукова новизна отриманих результатів полягає в дослідженні можливостей тестування на проникнення для сервісів, де об’єктом тестування є платформа з доступом до FPGA. Як і в багатьох інших сферах, забезпечення кібербезпеки платформ FPGA as a Service є комплексним, де ігнорування будь-якого компонента може призвести до критичних наслідків. Застосування лише тестування на проникнення недостатньо; тому надається вичерпний перелік заходів кібербезпеки для платформ FPGA as a Service, підкреслюючи актуальність і необхідність їх впровадження.

Ключові слова: FPGA; FPGA як сервіс; тестування на проникнення; забезпечення кібербезпеки; заходи захисту.

Тецький Артем Григорович – канд. техн. наук, доц. каф. комп’ютерних систем, мереж і кібербезпеки, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Перепелицин Артем Євгенович – канд. техн. наук, доц., доц. каф. комп’ютерних систем, мереж і кібербезпеки, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Ілляшенко Олег Олександрович – канд. техн. наук, доц., докторант каф. комп’ютерних систем, мереж і кібербезпеки, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна; Дослідник відділу довіри, інформаційної безпеки та приватності Інституту інформатики та телематики Національної дослідницької ради, Піза, Італія.

Морозова Ольга Ігорівна – д-р техн. наук, проф., проф. каф. комп’ютерних систем, мереж і кібербезпеки, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Узун Дмитро Дмитрович – канд. техн. наук, доц., доц. каф. комп’ютерних систем, мереж і кібербезпеки, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Artem Tetskyi – PhD, Associate Professor at the Department of Computer Systems, Networks and Cybersecurity, National Aerospace University «Kharkiv Aviation Institute», Kharkiv, Ukraine,
e-mail: a.tetskiy@csn.khai.edu, ORCID: 0000-0003-1745-2452, Scopus Author ID: 57202894656.

Artem Perepelitsyn – PhD, Associate Professor, Associate Professor at the Department of Computer Systems, Networks and Cybersecurity, National Aerospace University «Kharkiv Aviation Institute», Kharkiv, Ukraine,
e-mail: a.perepelitsyn@csn.khai.edu, ORCID: 0000-0002-5463-7889, Scopus Author ID: 56332607800.

Oleg Illiashenko – PhD, Associate Professor, Associate Professor at the Department of Computer Systems, Networks and Cybersecurity, National Aerospace University «Kharkiv Aviation Institute», Kharkiv, Ukraine,
e-mail: o.illiashenko@khai.edu.
Researcher at the Trust, Security and Privacy Research Unit of the Institute of Informatics and Telematics of the National Research Council (IIT CNR), Pisa, Italy,
e-mail: oleg.illiashenko@iit.cnr.it, ORCID: 0000-0002-4672-6400, Scopus Author ID: 55842633400.

Olga Morozova – Doctor of Technical Science, Professor, Professor at the Department of Computer Systems, Networks and Cybersecurity, National Aerospace University «Kharkiv Aviation Institute», Kharkiv, Ukraine,
e-mail: o.morozova@csn.khai.edu, ORCID: 0000-0001-7706-3155, Scopus Author ID: 57194517520.

Dmytro Uzun – PhD, Associate Professor, Associate Professor at the Department of Computer Systems, Networks and Cybersecurity, National Aerospace University «Kharkiv Aviation Institute», Kharkiv, Ukraine,
e-mail: d.uzun@csn.khai.edu, ORCID: 0000-0001-5574-550X, Scopus Author ID: 57194773530.