National Security Agency
Cybersecurity Technical Report

DoD Microelectronics:

Field Programmable Gate Array

Best Practices – Threat Catalog

December 2022

U/OO/230113-22
PP-22-1865
Version 1.0

For additional information, guidance or assistance with this document, please contact the Joint Federated Assurance Center (JFAC) at https://jfac.navy.mil.

Notices and history

Document change history

Date	Version	Description
December 2022	1.0	Initial Publication

Disclaimer of warranties and endorsement

The information and opinions contained in this document are provided "as is" and without any warranties or guarantees. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the United States Government, and this guidance shall not be used for advertising or product endorsement purposes.

Publication information

Author(s)

National Security Agency
Cybersecurity Directorate
Joint Federated Assurance Center

Contact information

Joint Federated Assurance Center: https://jfac.navy.mil
Cybersecurity Report Feedback / General Cybersecurity Inquiries: CybersecurityReports@nsa.gov
Defense Industrial Base Inquiries and Cybersecurity Services: DIB_Defense@cyber.nsa.gov
Media inquiries / Press Desk: Media Relations, 443-634-0721, MediaRelations@nsa.gov

Purpose

This document was developed in furtherance of NSA's cybersecurity missions. This includes its responsibilities to identify and disseminate threats to National Security Systems, Department of Defense information systems, and the Defense Industrial Base, and to develop and issue cybersecurity specifications and mitigations. This information may be shared broadly to reach all appropriate stakeholders.

Executive summary

This report categorizes and catalogs hardware assurance threats that apply to field programmable gate arrays (FPGA) as described in the Joint Federated Assurance Center (JFAC) FPGA Levels of Assurance (LoA) document. This report does not list all technical methods an attacker might employ, but rather it identifies categories where common mitigation strategies and approaches will be necessary. These threats originate from an adversary, are malicious, and compromise the operation of an FPGA-based system by:

- Modifying intended behavior
- Adding extraneous new behaviors
- Impeding or preventing operation
- Degrading operation or reliability
- Making use of known vulnerabilities in specific FPGA devices

This report can be used to better understand how JFAC selected its FPGA best practices and to understand the threats that require mitigation using alternative approaches.

Contents

Tables

1 Threat category process

The process for cataloging threats is characterized by researching threats, summarizing the data points, scoring the results, and categorizing the threats into the corresponding level. Each activity is further defined below.

The characteristics of these threats were examined for commonalities and then summarized in the following 11 broad categories:

- Adversary utilizes a known FPGA platform vulnerability
- Adversary inserts malicious counterfeit
- Adversary compromises application design cycle
- Adversary compromises system assembly, keying, or provisioning
- Adversary compromises third-party soft intellectual property (IP)
- Adversary swaps configuration file on target
- Adversary substitutes modified FPGA software design suite
- Adversary modifies FPGA platform family at design
- Adversary compromises single-board computing system (SBCS)
- Adversary inserts a compromise during the FPGA fabrication process
- Adversary modifies FPGA software design suite

Next, it was necessary to determine which threats were relevant to each LoA. The LoAs are defined in *Levels of Assurance Definitions and Applications*. They measure attacks by an adversary's investment and execution requirements and by the consequence and targetability of the outcome. Next, the threats were binned into groups that correspond to the appropriate level of assurance:

Table 1: Level of Assurance threats

Level	Threats
	Threats that have a low cost to implement and with high utility to the adversary.
	Threats with moderate costs to implement and that achieve moderate levels of utility to the adversary. Additionally, this is inclusive of all LoA1 threats.
	Threats with a high cost to implement and/or low utility to the adversary, in addition to all LoA1 and LoA2 threats.

The assurance team then scored these broad attack categories on the criteria specified in *Levels of Assurance Definitions and Applications*. As described in that document, the criteria include access, technology, investment, value of effect, and targetability.

The following table outlines the parameters for evaluating threats and identifying the corresponding level of assurance.

Table 2: Attack cost and attack value

	Criteria	LoA1	LoA2	LoA3
Attack cost	**Access**			
	A single available point of access	●	●	●
	A difficult point of access *or* multiple available points of access		●	●
	Multiple points of difficult access			●
	Technology			
	Existing public technology	●	●	●
	Low implementation risk technology		●	●
	Technologically feasible			●
	Investment			
	Minimal investment of resources	●	●	●
	A large multidisciplinary team		●	●
	A nation scale directed priority			●
Attack utility	**Value of effect**			
	Disable or subvert a system	●	●	●
	Establish vulnerabilities (for future exploitation)		●	●
	Degrade system performance			●
	Targetability			
	Inherently targetable and controllable	●	●	●
	Affects only a subset of systems		●	●
	Blind attacks (*Difficult to precisely target or control the outcome)*[1]			●

The scoring resulted in assigning the 11 threat categories to the level of assurance at which they first become relevant. However, a particular threat category could have one attack that falls into LoA1 and another that falls into LoA2.

1 As will be discussed later in this document, LoA3 systems are best approached with a full risk analysis to identify which blind attacks are of concern to a given system. Realistic risks in this space are often idiosyncratic, and the most concerning blind attacks are typically not the most expensive. LoA3 is the least "one size fits all" of all the categories specifically because many such systems are judged to need to concern themselves with such unpredictable effects.

Further, a threat mitigation appropriate at LoA1 might itself have vulnerabilities that make it unsuitable at LoA2. As a result, different or additional mitigations might be required for LoA2 than for LoA1. Similarly, mitigations for LoA3 will differ from those for LoA2. The specifics of this are defined on a threat-by-threat basis and are discussed in the JFAC FPGA assurance best practices documents. In some cases, a cost-effective mitigation might be sufficient for all LoAs. In others, a mitigation that is appropriate at LoA1 may be superseded by a more complex mitigation at a higher LoA. Because of this, the JFAC best practices documentation detailing valid mitigations is targeted at a specific level of assurance.

The threat categories addressed in this report are listed in the following table with the corresponding LoA(s) indicated.

Table 3: Threat description and LoA matrix

#	Threat description (TD)	LoA1	LoA2	LoA3
TD 1	Adversary utilizes a known FPGA platform vulnerability	●	●	●
TD 2	Adversary inserts malicious counterfeit	●	●	●
TD 3	Adversary compromises application design cycle	●	●	●
TD 4	Adversary compromises system assembly, keying, or provisioning	●	●	●
TD 5	Adversary compromises third-party soft IP	●	●	●
TD 6	Adversary swaps configuration file on target	●	●	●
TD 7	Adversary substitutes modified FPGA software design suite	●	●	●
TD 8	Adversary modifies FPGA platform family at design	●	●	●
TD 9	Adversary compromises single-board computing system (SBCS)	●	●	●
TD 10	Adversary inserts a compromise during the FPGA fabrication process		●	●
TD 11	Adversary modifies FPGA software design suite			●

2 Threat descriptions

2.1 TD 1 LoA1: Adversary uses a known FPGA platform vulnerability

In this threat, a foreign adversary uses a known vulnerability in an FPGA platform to attack a specific program. A known vulnerability is an unclassified published weakness in the design of an FPGA platform that allows an adversary to use it for malicious purposes. This threat does not focus on a particular vulnerability, but could be any weakness in the FPGA device. These vulnerabilities are published in public databases such as “Common Vulnerabilities and Exposures (CVE)” and the “National Vulnerabilities Database (NVD)”. Such vulnerabilities could allow for leakage of sensitive information or keys; compromise of security or tamper detection functions; or unauthorized reconfiguration of the product.

Criteria:

- **Access – *A single available point of access*** – the vulnerability can be exploited with a single point of access in the supply chain.
- **Technology – *Existing public technology*** – the vulnerability must already be known and able to be exploited utilizing commercially available tools and engineering methodology.
- **Investment – *Minimal investment of resources*** – Because the vulnerability is published and known, the attacker merely has to utilize the existing weakness.
- **Value of effect – *Disable or subvert a system*** – An insider or access to a fielded product could allow the attacker to use a variety of vulnerabilities to threaten a system.
- **Targetability – *Inherently targetable and controllable*** – This is dependent upon the type of vulnerability and access possessed by the attacker. For LoA1, the threat assumes the ability to exploit the weakness to attack a specific target.

While the possible vulnerabilities are too numerous to address individually, they could include methods to bypass the device’s security functions, extract sensitive data, disable the tamper detection features, or modify the configuration data. Examples include:

- A vulnerability in the authentication of a partial reconfiguration file could allow unauthorized reprogramming of the part.

- A vulnerability in the configuration clock could allow an attacker to turn off security features using “clock glitching” during power-up.

2.2 TD 2 LoA1: Adversary inserts malicious counterfeit

In this threat, a foreign adversary with access to an existing counterfeit design and proven fabrication process inserts additional logic for unauthorized access or malicious attacks. The adversary then arranges for that counterfeit to be used in a target system.

In general, modifications to an FPGA platform at the fabrication stage are considered to require a high level of effort. However, in cases where an adversary country has already successfully designed a compatible device, albeit a counterfeit, that work has already been performed. In this case, the adversary’s overall investment in the effort is reduced. Additionally, modifications to the package of an FPGA may require a lower level of effort, and are included within this threat. For it to affect assurance, this threat is only of a counterfeit that contains functional differences that are security relevant. This is a critical distinction.

Additionally, in this threat, there are new distinct threats introduced at each level of assurance that can be summarized by the following:

LoA1 - Counterfeits created in an unauthorized fabrication facility as previously described.

LoA2 - Counterfeits created in an authorized fabrication facility including package modifications.

LoA3 - Reliability concerns and all remaining threats.

While re-marked components may represent a reliability risk, they are not included in this threat. The threat from re-marked components is commercial, economic in nature, and neither controllable nor targetable.

This threat includes reliability degradation of an FPGA at manufacture. Programs with specific reliability requirements should plan for the appropriate level of testing to verify that their design and components meet those goals.

Criteria:

- **Access – *A single available point of access*** – This attack requires a single point of access in the distribution chain.
- **Technology – *Existing public technology*** – Given the existing design and a team that understands it, this is a straightforward design task using the same publicly available tools used in the design process.
- **Investment – *Minimal investment of resources*** – Given an existing design, this is a straightforward design task. While the investment may vary depending on the complexity of the attack, the level of effort is akin to developing an IP block according to a specification.
- **Value of effect – *Disable or subvert a system*** – In this case, the adversary has a wide degree of latitude in terms of available behaviors. While they may not know the precise configuration, they design the underlying platform and can work around that challenge.
- **Targetability – *Inherently targetable and controllable*** – In this scenario, the adversary can control the system containing the device. A malicious counterfeit could contain command and control functions triggered from multiple places, including hardware with known protocols.

Examples:

- An adversary has counterfeited an FPGA platform from a mainline vendor and inserted additional logic into it that compromises its security settings when sensing a specific input/output (IO) pattern. These platforms are then introduced into the supply chain of a DoD program in place of unaltered devices.
- An adversary has counterfeited an FPGA platform from a mainline vendor and inserted additional logic into it that acts as a kill switch when sensing a specific IO pattern. These platforms are then introduced into the supply chain of a DoD program in place of unaltered devices.

2.3 TD 3 LoA1: Adversary compromises application design cycle

In this threat, an adversary has access to the design process and data related to the application design effort that incorporates an FPGA. In short, this is the classic insider threat. The bad actor takes advantage of access to modify design code, change FPGA configuration settings, or substitute a modified configuration file that was authenticated and built with the same tools and keys the design team used. Such an attacker is in a

particularly advantageous position because they can monitor the design process and modify the product during any phase of the design.

Criteria:

- **Access – *A single available point of access*** – This attack can be conducted by an insider or through a computer network penetration. By default, JFAC assumes the design flow does not sufficiently control people, facilities, and networks to be considered a "difficult point of access." For many USG design efforts, typical application development uses this type of Internet connected commercial setting.
- **Technology – *Existing public technology*** – This attack requires the same commercial synthesis and design tools that are used in FPGA design efforts.
- **Investment – *Minimal investment of resources*** – Because an attacker has access to the hardware description language (HDL) and commercial design software, the process to carry out this attack is straightforward and is similar to developing any other IP block to a known specification.
- **Value of effect – *Disable or subvert a system*** – An insider can modify the design in arbitrary ways, leading to a wide range of possible arbitrary effects.
- **Targetability – *Inherently targetable and controllable*** – An insider understands the communications of the device and the various scenarios in which it will be used. They know the system they are developing. This leaves them many opportunities to target and control their effects.

Examples:

- A compromised insider builds a malicious function or backdoor into the register transfer language (RTL) of a design and modifies the test procedures to overlook or to "pass" the function as original behavior.
- An outsider uses a network penetration capability to insert a malicious function or to compromise a security setting on the system after design verification.
- A compromised insider swaps out a validated final configuration file with a compromised version prior to deployment.

2.4 TD 4 LoA1: Adversary compromises system assembly, keying, or provisioning

In this threat, an adversary carries out an attack on the product during printed circuit board assembly, key injection, or device configuration. This attack could include the assembly house replacing authentic FPGA parts with counterfeit ones, compromising configuration files, or stealing or modifying encryption keys.

Criteria:

- **Access – *A single available point of access*** – This attack can be carried out by a single insider and in some cases by remote access.
- **Technology – *Existing public technology*** – These attacks only require using existing technology at the assembly site.
- **Investment – *Minimal investment of resources*** – Since an insider would already have access to customer design data or the assembly floor, all of the attacks require minimal investment.
- **Value of effect – *Disable or subvert a system*** – The usefulness of this depends on the specific effect. The assembly house can change the device to a counterfeit or swap a configuration file depending on the specifics of what is performed at the assembly house.
- **Targetability – *Inherently targetable and controllable*** – This attack targets specific programs or end users and is not inherently random in nature. However, the usefulness depends on the kind of attack. A counterfeit part or configuration file swap at assembly can be used to insert a controllable Trojan.

Examples:

- An adversary steals a system's keys and configuration files from a third-party assembly house, enabling them to place new designs once the device is fielded.
- An adversary swaps the configuration file on targeted devices in a system during keying and provisioning.
- An adversary substitutes compromised counterfeit parts for inclusion in the system.

2.5 TD 5 LoA1: Adversary compromises third-party soft IP

In this threat, an adversary compromises a vendor who sells third-party soft IP intended for synthesis into the application design. This IP can be provided through the FPGA

development software's IP libraries or directly to the end user by a vendor. The compromise of the third-party IP (3PIP) vendor can occur during the IP design phase in a manner similar to the attack seen in TD 3: *Adversary compromises application design cycle*. Additionally, the attack could take place during the distribution of the IP to the application design team.

Criteria:

- **Access – *A single available point of access*** – This attack can be conducted by an insider or via computer network penetration at the 3PIP vendor or in its distribution chain.
- **Technology – *Existing public technology*** – This attack requires the same synthesis and design tools used in FPGA designs.
- **Investment – *Minimal investment of resources*** – Because an attacker doing this can have access to the 3PIP HDL, the process of carrying out this attack occurs in a straightforward design process.
- **Value of effect – *Disable or subvert a system*** – This depends on the piece of IP replaced. Some IP would only enable pre-positioning or lower utility attacks. Other IP would provide greater opportunities. In general, 3PIP is complex and has substantial responsibility in any system, opening the avenue to substantial effects.
- **Targetability – *Inherently targetable and controllable*** – In the case where IP is distributed directly to the end user, the opportunity for a targeted attack is significant. Further, the third party IP is frequently used in a way that provides it access to data about the outside world, either directly or indirectly.

Examples:

- A third-party supply house owned by an adversary provides a compromised complex IP block to the program. This IP monitors an input for a trigger signature to enable a kill feature.
- An adversary impersonates a trusted third-party supply house and provides a compromised security IP block to the program. The block fails to apply its security countermeasures after a set time period designed to avoid detection in testing.

2.6 TD 6 LoA1: Adversary swaps configuration file on target

In this threat, an adversary obtains access to systems during or after assembly and is able to modify the behavior of the device via the configuration file.

Criteria:

- **Access – *A single available point of access*** – A single bad actor can conduct this attack at manufacturing, in shipping, in the field, or via network penetration.
- **Technology – *Existing public technology*** – This threat involves swapping out an original configuration file with a modified one. Commercially available protections (such as built-in configuration file authentication) could elevate an attack on the same system to a more expensive and complex threat level. Details of this are in the best practices document for each level of assurance.
- **Investment – *Minimal investment of resources*** – The same example referenced in the preceding technology section shows that this attack can be carried out by a team of a few people and with relative speed.
- ***Value of effect – Disable or subvert a system*** – This attack gives the adversary full control over the behavior of the FPGA. They can design an arbitrary behavior that will not be immediately apparent, but which has useful behaviors when triggered.
- **Targetability – *Inherently targetable and controllable*** – This threat is inherently targetable and carried out on a specific program. Additionally, it allows the adversary to insert a Trojan with a controllable trigger, as the adversary has full control of the configuration file they inject into the target system.

Examples:

- An adversary intercepts the system in the field and swaps in a modified configuration file in order to compromise the mission.
- An adversary remotely reprograms a fielded FPGA-based part with a new configuration file, adding arbitrary functionality.

2.7 TD 7 LoA1: Adversary substitutes modified FPGA software design suite

In this threat, an adversary replaces the design suite used by the application designers with one modified to subvert the design at synthesis, implementation, or configuration

file generation. Such an adversary would need both the means to modify the vendor tool and to insert it into the supply chain.

Criteria:

- **Access – *A single available point of access*** – This threat only requires a single insider to provide modified software to the FPGA application designer. The software is readily available via commercial channels and malicious modification of the software is achievable with known commercial techniques. This attack can also be performed by compromising the distribution flow of the tools or identifying cases where unofficial tools are being used.
- **Technology – *Existing public technology*** – This threat utilizes existing commercial technology to decompile, reverse engineer, and modify a vendor's software for redistribution to the FPGA application designer.
- **Investment – *Minimal investment of resources*** – This attack can be carried out with few people in a relatively short period of time.
- **Value of effect – *Disable or subvert a system*** – The synthesis software can have arbitrary effects, as it is "trusted" to translate the design into the configuration file to be loaded.
- **Targetability – *Inherently targetable and controllable*** – Since the software is being provided directly to the intended program, it is inherently targetable. The adversary can add program-specific functions that can be triggered. This same threat could also be used to pre-position for a future attack.

Examples:

- A compromised FPGA tool generates a configuration file that does not disconnect Joint Test Action Group ports but reports that it did.
- A compromised FPGA tool can provide a medium to insert compromised critical 3PIP into an end-user design.
- A compromised FPGA tool generates a configuration file that disables tamper responses but reports that they are in place and functioning as expected.
- A compromised FPGA tool generates a synthesis that does not meet timing thresholds but reports that the design does.

2.8 TD 8 LoA1: Adversary modifies FPGA platform family at design

In this threat, an adversary compromises the FPGA platform during the design stage such that it will compromise the security of devices when in use. This also includes threats in which an adversary compromises a piece of third-party hard IP that is used by the platform design team. While the access required to do this is comparable to the access required for compromising the application design, the targetability is significantly more difficult.

Criteria:

- **Access – *A single available point of access*** – This attack only requires a single insider or network exploitation to compromise the design of the FPGA.
- **Technology – *Existing public technology*** – This threat involves potentially changing the Verilog or other hardware description language. As a result, it is relatively straightforward to implement.
- **Investment – *Minimal investment of resources*** – Malicious register transfer language changes can easily be developed by an individual or small team.
- **Value of effect – *Disable or subvert a system*** – A malicious modification to the platform during design can have a wide array of effects on the behavior of the device or overall health of the system in which it is incorporated. Some parts of the design will have predictable connections to external communications and these qualify this threat for LoA1. Most other parts of the design are of concern for establishing vulnerabilities only.
- **Targetability – *Inherently targetable and controllable*** – This is highly dependent on the specific block of the FPGA being attacked. Some attacks could be triggered for a denial of service based on a given command while others could provide pre-positioning for future access.

Examples:

- An insider subverts the authentication mechanisms for remote configuration to allow a compromise of the protections against unauthorized remote programming.
- An insider modifies the design of a high-speed transceiver designed to process a known protocol, such as 10G Ethernet. The transceiver waits for a known IP packet and triggers an immediate hardware failure.

2.9 TD 9 LoA1: Adversary compromises single-board computing system (SBCS)

In this threat, an adversary compromises a single-board computing system (SBCS) purchased by a program for use in a system. An SBCS is a commercial off-the-shelf (COTS) product consisting of a printed circuit board (PCB) with FPGAs and computer processing resources. These boards are ubiquitous throughout DoD systems as they are readily available in the marketplace. Additionally, their relative technical simplicity and low expense do not justify the fabrication of custom solutions by programs. Under this threat, the program does not have control of the manufacturing process of the SBCS, forcing the program to rely upon a verification-heavy approach to mitigating threats. Of primary concern in this scenario are threats to the:

- Authenticity of the FPGA devices themselves,
- Configuration methodology,
- PCB connections, and
- Test interfaces.

PCB concerns are left to the program to resolve with the assistance of the JFAC PCB Executive Agent.

Criteria:

- **Access - *A single available point of access*** – This attack only requires a single insider or network exploitation to compromise the design of the SBCS.
- **Technology – *Existing public technology*** – This threat involves potentially changing the board design or the configuration process on SBCS, or utilizing counterfeit FPGA devices in the build.
- **Investment – *Minimal investment of resources*** – Design changes or swapping in counterfeit parts is not costly.
- **Value of Effect – *Disable or subvert a system*** – A malicious modification to the SBCS during design can have a wide array of effects on the behavior of the device or overall health of the system in which it is incorporated. Some parts of the design will have predictable connections to external communications and these qualify this threat for LoA1.
- **Targetability – *Inherently targetable and controllable*** – This is highly dependent on the specific block of the FPGA being attacked. Some attacks could

be triggered for a denial of service based on a given command while others could provide pre-positioning for future access.

Examples:

- An insider subverts the authentication mechanisms for remote configuration to compromise the protections against unauthorized programming.
- An insider swaps in counterfeit FPGAs without configuration file protections or tamper detect functions.

2.10 TD 10 LoA2: Adversary compromises the FPGA fabrication process

In this threat, a bad actor compromises the fabrication facility where an FPGA is manufactured. They change the design in a way that is beneficial to the adversary and has the result manufactured, either in addition to the normal parts, or in their place. This modification would be limited because of the re-configurable nature of an FPGA. However, valid attack scenarios could be enabled by such a modification.

Criteria:

- **Access – *A single available point of access*** – Multiple individuals in the fabrication process would be required to effect useful mask or process changes and conceal the results. However, in at least some cases, these individuals together could be considered a group of associated foreign nationals.
- **Technology – *Existing public technology*** – This threat involves making one or more changes in the physical processing of the FPGA silicon itself. This can be accomplished using existing tools and training.
- **Investment – *A large multidisciplinary team*** – The technology to do this is commercially available, but the process of redesigning an integrated circuit would involve specialists from a wide array of disciplines beyond the normal FPGA application design.
- **Value of Effect – *Establish vulnerabilities*** – This threat assumes a change during fabrication that does not represent the production of an entirely new counterfeit. Because of technical limitations, such an attack would more likely be to pre-position a device for a future compromise by introducing a vulnerability or as only part of a complete attack.

- **Targetability – *Inherently targetable and controllable*** – This threat occurs too early in the production line and therefore is far removed from the intended device. The adversary would have to ensure that whatever effect is inserted does not trigger outside of very controlled circumstances selected by the adversary. This is a large constraint on the adversary, but achievable.

Examples:

- An adversary compromises several mask manufacturing company employees to obtain and modify design masks to introduce a malicious function into an FPGA.
- An adversary compromises several foundry employees to modify the fabrication process on a wafer run in order to compromise fuse-based security features in the FPGA.

2.11 TD 11 LoA3: Adversary modifies FPGA software design suite

In this threat, an adversary has compromised the development of the vendor production FPGA synthesis and configuration file generation software such that it will compromise the security of all devices when in use. While the access required is minimal, the targeting is significantly more difficult.

Criteria:

- **Access – *A single available point of access*** – This attack only requires a single insider or network exploitation to compromise the source code and/or executables.
- **Technology – *Existing public technology*** – This threat involves making one or more changes to the FPGA synthesis algorithms or configuration file generation. This can be accomplished using existing tools and training.
- **Investment – *Minimal investment of resources*** – The technology is commercially available and the investment in tools and techniques would be relatively minimal.
- **Value of Effect – *Establish vulnerabilities*** – Because this threat occurs so far removed from any particular application, it is suitable for introducing vulnerabilities for later use, rather than to introduce a specific effect.
- **Targetability – *Blind attacks*** – This threat occurs too early in the production line and therefore is too far removed from the intended device to be targetable. Such a threat represents a great risk of actor exposure.

Example:

- An adversary compromises an FPGA vendor employee to introduce a malicious function into the configuration file generation tools. The result compromises the security features across an entire family of devices.

3 Summary

All of these categories focus specifically on assurance-related threats. They are not inclusive of confidentiality, security, or tamper-related attacks except in areas where they overlap assurance concerns. As a program utilizing FPGAs seeks to apply appropriate mitigations for a given Level of Assurance, that program will be expected to provide mitigations against the threat categories listed under the corresponding threat level. The mitigations suggested by the JFAC best practices have been evaluated to mitigate those threats to the appropriate level. As expected, for higher LoAs, the program will need to address the same threats with higher LoA mitigations. JFAC is available to guide programs through this process. Additional information for JFAC may be found at https://jfac.navy.mil.

Appendix A: Standardized terminology

The following terms are used in the Joint Federated Assurance Center Field Programmable Gate Array Best Practices documents. These terms are modified from Defense Acquisition University definitions to support common understanding.

Application design – The collection of schematics, constraints, hardware description language (HDL), and other implementation files developed to generate an FPGA configuration file for use on one or many FPGA platforms.

Application domain – This is the area of technology of the system itself, or a directly associated area of technology. For instance, the system technology domain of a radar system implemented using FPGAs would be "radar" or "electronic warfare."

Configuration file – The set of all data produced by the application design team and loaded into an FPGA to personalize it. Referred to by some designers as a “bitstream”, the configuration file includes that information, as well as additional configuration settings and firmware, which some designers may not consider part of their “bitstream.”

Controllable effect – Program-specific, triggerable function allowing the adversary to attack a specific target.

Device/FPGA device – A specific physical instantiation of an FPGA.

External facility – An unclassified facility that is out of the control of the program or contractor.

Field programmable gate array (FPGA) – In this context FPGA includes the full range of devices containing substantial reprogrammable digital logic. This includes devices marketed as FPGAs, complex programmable logic devices (CPLD), system-on-a-chip (SoC) FPGAs, as well as devices marketed as SoCs and containing reprogrammable digital logic capable of representing arbitrary functions. In addition, some FPGAs incorporate analog/mixed signal elements alongside substantial amounts of reprogrammable logic.

FPGA platform – An FPGA platform refers to a specific device type or family of devices from a vendor.

Hard IP – Hard IP is a hardware design captured as a physical layout, intended to be integrated into a hardware design in the layout process. Hard IP is most typically distributed as Graphic Design System II (GDSII). In some cases, Hard IP is provided by a fabrication company and the user of the IP does not have access to the full layout, but simply a size and the information needed to connect to it. Hard IP may be distributed with simulation hardware description language (HDL) and other soft components, but is defined by the fact that the portion that ends up in the final hardware was defined by a physical layout by the IP vendor.

Level of assurance (LoA) – A Level of Assurance is an established guideline that details the appropriate mitigations necessary for the implementation given the impact to national security associated with subversion of a specific system, without the need for system-by-system custom evaluation.

Physical unclonable function (PUF) – This function provides a random string of bits of a predetermined length. In the context of FPGAs, the randomness of the bitstring is based upon variations in the silicon of the device due to manufacturing. These bitstrings can be used for device IDs or keys.

Platform design – The platform design is the set of design information that specifies the FPGA platform, including physical layouts, code, etc.

Soft IP – Soft IP is a hardware design captured in hardware description language (HDL), intended to be integrated into a complete hardware design through a synthesis process. Soft IP can be distributed in a number of ways, as functional HDL or a netlist specified in HDL, encrypted or unencrypted.

System – An aggregation of system elements and enabling system elements to achieve a given purpose or provide a needed capability.

System design – System design is the set of information that defines the manufacturing, behavior, and programming of a system. It may include board designs, firmware, software, FPGA configuration files, etc.

Target – A target refers to a specific deployed instance of a given system, or a specific set of systems with a common design and function.

Targetability – The degree to which an attack may have an effect that only shows up in circumstances the adversary chooses. An attack that is poorly targetable would be more likely to be discovered accidentally, have unintended consequences, or be found in standard testing.

Third-party intellectual property (3PIP) – Functions whose development are not under the control of the designer. Use of the phrase “intellectual property”, IP, or 3PIP in outlining this methodology of design review does not refer to property rights, such as, for example, copyrights, patents, or trade secrets. It is the responsibility of the party seeking review and/or the reviewer to ensure that any rights needed to perform the review in accordance with the methodology outlined are obtained.

Threat category – A threat category refers to a part of the supply chain with a specific attack surface and set of common vulnerabilities against which many specific attacks may be possible.

Utility – The utility of an attack is the degree to which an effect has value to an adversarial operation. Higher utility effects may subvert a system or provide major denial of service effects. Lower utility attacks might degrade a capability to a limited extent.

Vulnerability – A flaw in a software, firmware, hardware, or service component resulting from a weakness that can be exploited, causing a negative impact to the confidentiality, integrity, or availability of an impacted component or components.