Defeating Cisco Trust Anchor: A Case-Study of Recent Advancements in Direct FPGA Bitstream Manipulation

Jatin Kataria, Rick Housley, Joseph Pantoga, Ang Cui
$\{j, r, jp, a\}$@*redballoonsecurity.com*
Red Balloon Security

Abstract

Field-programmable gate arrays (FPGAs) are widely used in real-time, data-intensive, and mission critical system designs. In the space of trusted computing, FPGA-based security modules have appeared in a number of widely used security conscious devices. The Cisco Trust Anchor module (TAm) is one such example that is deployed in a significant number of enterprise network switches, routers, and firewalls. We discuss several novel direct FPGA bitstream manipulation techniques that exploit the relative simplicity of *input* and *output* pin configuration structures.

We present an analysis of the efficacy of Cisco TAm and discuss both the high-level architectural flaws of the TAm as well as implementation specific vulnerabilities in a TAm-protected Cisco router. By combining techniques presented in this paper with other recent advancements in FPGA bitstream manipulation, we demonstrate the feasibility of reliable remote exploitation of all Cisco TAms implemented using Xilinx Spartan-6 FPGAs. The TAm exploit described in this paper allows the attacker to fully bypass all Trust Anchor functionality, including hardware-assisted secure boot, and to stealthily inject persistent malicious implants within both the TAm FPGA and the application processor. Lastly, we discuss the applicability of our bitstream manipulation techniques to other FPGA-based devices and propose several practical mitigations.

1 Introduction

Since the initial implementation of a secure bootstrapping mechanism for computer systems [1], system designs have increased in diversity and complexity. To meet the security requirements of such a variety of architectures, many secure and trusted boot implementations have been proposed. Generally, these implementations require dedicated hardware for security such as a Trusted Platform Module (TPM) [16] or extensions to the processor feature-set such as ARM TrustZone [2] or Intel Trusted Execution Technology [18]. Such technologies have been rapidly adopted in platforms with homogeneous hardware architectures such as consumer PCs, servers, and mobile devices. However, they may not meet the requirements of equipment manufacturers who want a unified secure boot implementation across a set of diverse hardware platforms such as those found in the industrial control or telecommunications industries. Additionally, these companies may want complete ownership over the system to avoid external dependencies or vendor lock-in. With these requirements, FPGAs are an attractive and cost-effective alternative to traditional hardware trust anchors. This has been recognized and embraced by FPGA manufacturers such as Microsemi [20], who have developed products to meet this demand.

This paper describes vulnerabilities in the proprietary Cisco Trust Anchor module on the ASR 1001-X router, an FPGA-anchored secure boot implementation, that allow for the exploitation of Cisco Secure Boot on that device. These vulnerabilities exist due to two critical faults. First, the TAm implementation is vulnerable to unauthorized hardware and software modification. Second, the FPGA hardware that loads the TAm bitstream has no means to authenticate its own configuration, and the bitstream remains mutable during operation and across power cycles. Regardless of hardware improvements to the design, the TAm will likely remain flawed so long as it aims to achieve immutability through reprogrammable hardware. We demonstrate that recent advancements in FPGA bitstream manipulation techniques combined with the fundamental design flaws of the Cisco Secure Boot process, have made TAm bypass attacks feasible.

The techniques described in this paper can be applied to other FPGA-based security modules, as well as FPGA-based devices in other applications. The main insights that enabled the FPGA bitstream manipulation attack described herein are the following:

1. Although modern FPGAs have hundreds of thousands of logical cells, they only have several hundred physical pins. For example, devices from the Spartan-6 family of Xilinx FPGA have between 150 and 550 pins.

2. Reverse engineering this very small set of physical pin configurations is much simpler and more practical than performing netlist reconstruction on large numbers of logical cells.
3. Since FPGAs must ultimately communicate with the rest of the system through these I/O pins, the attacker can manipulate the functionality of the FPGA by modifying I/O pin behavior without having to perform analysis of the logic.

In the following sections, we provide an architectural overview of the Cisco TAm, describe the analysis necessary to make its exploitation possible, and discuss the specifics of our attack. Finally, we discuss possible mitigations, ideas for future work, and present our conclusions.

2 Cisco Trust Anchor Secure Boot

First commercially introduced in 2013, Cisco implemented a proprietary secure boot mechanism known as the Trust Anchor module (TAm). An FPGA is used as the root of trust to validate the bootloader image for the next stage in the secure boot process.

Figure 1 from the Cisco TAm patent [15] shows a high-level example of the secure boot process facilitated by the TAm. The FPGA containing the TAm verifies the integrity of the application processor's microloader stored in its own bitstream, as well as the bootloader firmware stored in external storage. The processor, executing the authenticated microloader and bootloader, then performs authentication of the OS before allowing execution of the next stage in the boot process.

2.1 Cisco ASR 1001-X Secure Boot

The Cisco Secure Boot process can be broken down into two discrete stages. First, the FPGA-based TAm loads its configuration bitstream from an external serial peripheral interface (SPI) flash chip ("the bitstream SPI flash"). Once configured, the TAm performs integrity verification in order to provide a "trusted" microloader binary to the Cisco Route Processor (RP), an Intel Xeon-based CPU. The content of this microloader is stored in the Block RAM (BRAM) section of the Xilinx FPGA bitstream. Second, the RP executes the "trusted" microloader binary by loading it via a SPI interface emulated by the TAm. Once activated, the TAm performs the verification of the bootloader firmware, and will reset the RP upon failed validation after 100 seconds.

2.2 Boot Stage 0: FPGA Initialization

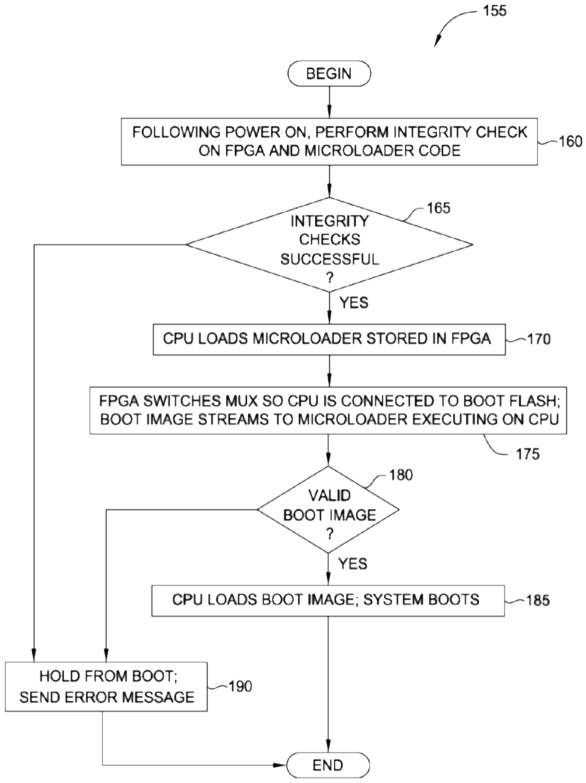

155
BEGIN
FOLLOWING POWER ON, PERFORM INTEGRITY CHECK ON FPGA AND MICROLOADER CODE
160
INTEGRITY CHECKS SUCCESSFUL ?
165
YES
CPU LOADS MICROLOADER STORED IN FPGA
170
FPGA SWITCHES MUX SO CPU IS CONNECTED TO BOOT FLASH; BOOT IMAGE STREAMS TO MICROLOADER EXECUTING ON CPU
175
VALID BOOT IMAGE ?
180
YES
CPU LOADS BOOT IMAGE; SYSTEM BOOTS
185
HOLD FROM BOOT; SEND ERROR MESSAGE
190
END

Figure 1: Patent App. Pub. US 2012/0303941 A1 Fig. 1B

The Cisco ASR 1001-X follows the secure boot mechanism described in Section 2, using the hardware identified in Figure 3. First, the FPGA is configured using the bitstream stored in the bitstream SPI flash. The FPGA then performs authentication on the microloader and the bootloader (stored in the bootloader flash). The TAm emulates a SPI flash memory slave and provides the microloader to the RP, which fetches the bootloader from redundant bootloader flash and continues the typical boot process of a Linux-on-x86 platform. A summary of the Cisco IOS XE boot chain is described in Figure 2.

Visual evidence, obtained via electromagnetic spectrum analysis, of the order of these first boot stages is shown in the waterfall diagrams in Figure 4.

Here, a magnetic field probe, attached to a Keysight 9030B spectrum analyzer centered at 145 MHz, was affixed over the FPGA bitstream SPI flash, the FPGA, and the RP's power circuitry during boot. We can clearly see two power spikes around the 12th harmonic of the SPI clock (~12 MHz), corresponding to the initial reads of the FPGA bitstream from flash, preceding the startup of the RP itself.

2.3 Boot Stage 1: Cisco RP Initialization

The authenticated bootloader follows the UEFI specification [14] to boot the device into Linux running Cisco IOS XE [7]. Cisco has also implemented a proprietary Pre-EFI module (PEIM) [14] (Pre-ROMmon) to manage the Cisco ROM Monitor (ROMmon) [8], a proprietary, interactive boot environment. This module reads the result of the boot image validation performed by the FPGA, displays the system integrity status on the console, and also manages the bootloader upgrade process. The Cisco ASR 1001-X has two redundant SPI bootloader flash chips, as illustrated in Figure 3, which store the same copy of the bootloader. While upgrading the bootloader, the secondary SPI flash is updated with the new image from Linux.

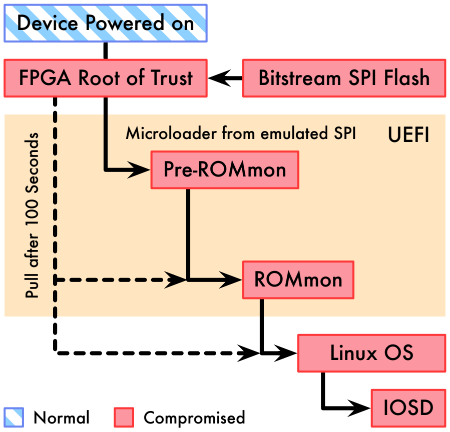


Figure 2: Compromised Boot Chain of Cisco IOS XE

On the next boot cycle, Pre-ROMmon boots from the primary SPI flash, checks the upgrade flag and, if set, validates the new copy stored in the secondary SPI bootloader flash. On successful validation, the secondary SPI flash is copied over the primary SPI flash, effectively applying the update. On failed validation, in case of a modified or corrupted firmware update, Pre-ROMmon makes three attempts to perform the upgrade. After these attempts, it boots from the original copy of the bootloader firmware, as illustrated in Figure 5.

ROMmon, implemented as a Driver eXecution Environment (DXE) module [14], provides an interactive console to boot firmware packages and also validates the IOS XE Linux kernel's digital signature. Once Linux has booted, it sets up the initial environment and starts Cisco IOS as a user-space process.

2.4 100 Seconds of Solitude

Any modification to the system's bootloader firmware stored in the SPI bootloader flashes is allowed to execute on the RP, printing a system integrity status to standard output with detected failures, shown in Figure 7. However, after approximately 100 seconds, the RP is halted and reset by the Trust Anchor, shown in Figure 2, if the bootloader cannot be validated. This delayed reset is a design flaw, and allows 100 seconds of execution to perform dynamic analysis of not only the bootloader firmware UEFI [14], but also of the microloader stored inside TAm as mentioned in Section 2.

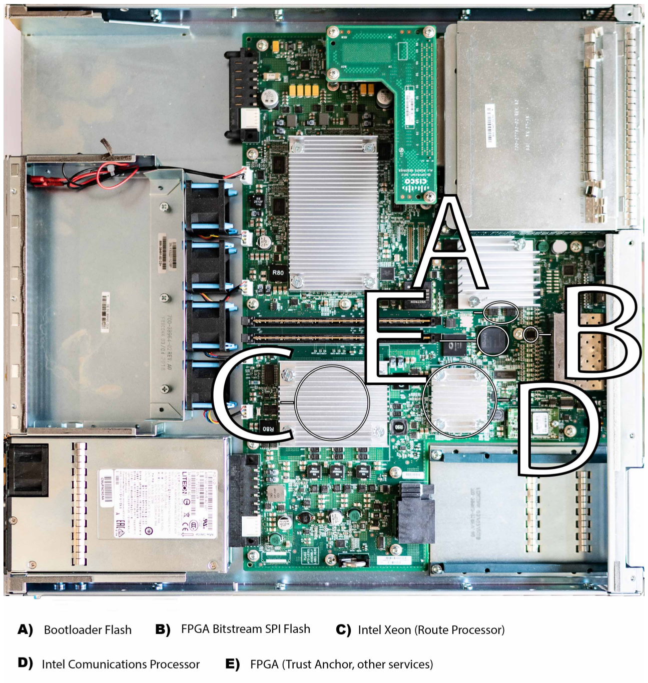

A
B
C
D
E
A) Bootloader Flash
B) FPGA Bitstream SPI Flash
C) Intel Xeon (Route Processor)
D) Intel Comunications Processor
E) FPGA (Trust Anchor, other services)

Figure 3: Cisco ASR1001-X PCB

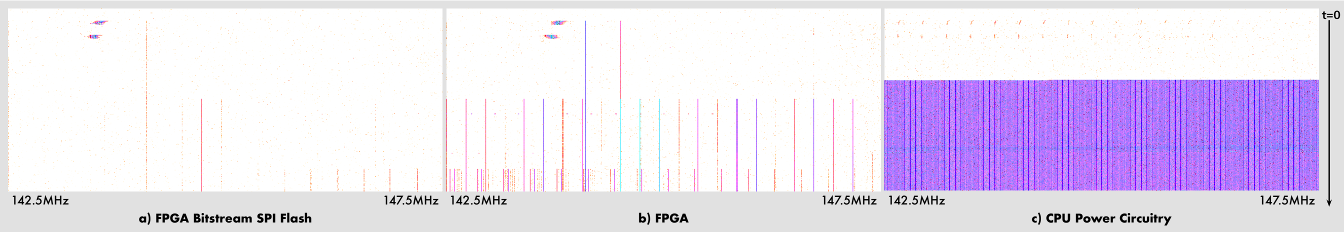
2.5 Trust Anchor module Bitstream Update

The Cisco IOS XE [7] firmware used by the ASR 1001-X runs Cisco IOS [7] as a daemon on Linux. Our analysis of IOS XE shows that the underlying Linux kernel provides a utility, ***fpga_mb_program***, which provides a feature to update the configuration bitstream stored in the bitstream SPI flash. This utility performs input/output control ("ioctl") system calls to the cpld1NG.ko kernel driver, where "1NG" represents the RP's subtype. The kernel driver provides an interface to read and write to the SPI flash that contains the TAm FPGA configuration bitstream. One notable flaw in the utility and the driver is that no authentication mechanisms are utilized to verify the user supplied configuration bitstream. Therefore, the root of trust of the Cisco Secure Boot chain can be mutated with any loadable bitstream.

3 TAm Bypass Attack Principles

We assume the attack model of an adversary who aims to achieve persistence and load modified firmware onto the Cisco router. The adversary can achieve this in three different ways, as illustrated in Figure 6:

1. Change the configuration bitstream of the TAm to prevent the FPGA from resetting the system. While this

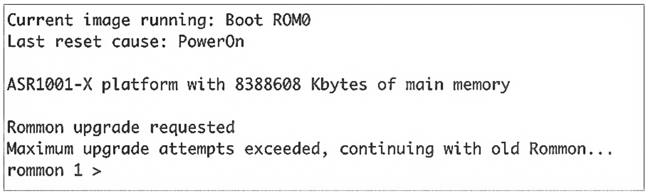


a) FPGA Bitstream SPI Flash

b) FPGA

c) CPU Power Circuitry

Figure 4: Electromagnetic Spectrum During Boot at 145 MHz (5 MHz Span)



```
Current image running: Boot ROM0
Last reset cause: PowerOn

ASR1001-X platform with 8388608 Kbytes of main memory

Rommon upgrade requested
Maximum upgrade attempts exceeded, continuing with old Rommon...
rommon 1 >
```

Figure 5: Failed Bootloader Upgrade Attempt

could cause errors to be printed to standard out, that can be prevented by modifying the bootloader firmware.

2. Change the configuration bitstream of the TAm to disable the validation of the bootloader firmware.
3. Change the configuration bitstream of the TAm to update the signature of the modified bootloader firmware.

All of the aforementioned attack scenarios require updating the configuration bitstream of the mutable TAm. We chose the first scenario to manipulate the FPGA bitstream as it circumvents the need to perform register-transfer-level (RTL) reconstruction. RTL reconstruction is highly complex and is currently infeasible without intimate knowledge of the specific FPGA hardware design.

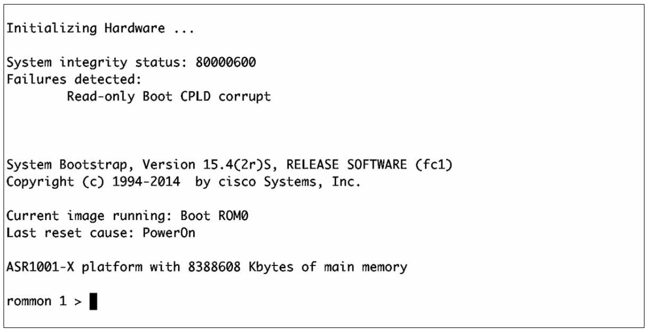
As mentioned in Section 2.5, one of the challenges in updating the firmware is accessing the driver interface provided by cpld1NG.ko. The adversary needs root privilege access to the Linux kernel to access the driver interface.

3.1 Cisco TAm Attack Paths

Table 2 in Appendix C summarizes the vulnerabilities disclosed, in coordination with Cisco, as a result of this research. Aside from the TAm vulnerability that is the subject of this paper, we also disclosed a command injection vulnerability that allows an authenticated remote attacker to execute commands on the underlying Linux shell with root privileges. These vulnerabilities can be chained together to create a remotely exploitable attack chain that reliably bypasses the Cisco Secure Boot. To demonstrate the practicality of such an attack chain, Appendix B lists recent vulnerability disclosures since 2018 that could potentially provide an attacker with the level of access required to target the TAm.

4 Cisco Trust Anchor Exploitation

The Cisco ASR 1001-X uses an SRAM-based FPGA as its Trust Anchor module (TAm). Since SRAM-based FPGAs are volatile by nature, the TAm is susceptible to bitstream interception and manipulation. In this section, we discuss the implementation of our TAm bypass, involving novel methods of reliably manipulating FPGA functionality through bitstream analysis and modification while sidestepping the need to perform RTL reconstruction. We are able to reduce the complexity of this process by only reverse engineering the FPGA's input and output interfaces.



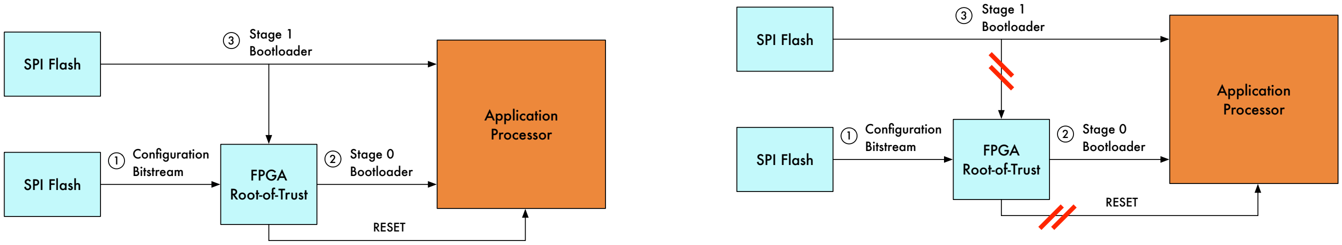
```
Initializing Hardware ...

System integrity status: 80000600
Failures detected:
        Read-only Boot CPLD corrupt



System Bootstrap, Version 15.4(2r)S, RELEASE SOFTWARE (fc1)
Copyright (c) 1994-2014  by cisco Systems, Inc.

Current image running: Boot ROM0
Last reset cause: PowerOn

ASR1001-X platform with 8388608 Kbytes of main memory

rommon 1 > █
```

Figure 7: Reported Boot Failures

The use of our methods of manipulation create numerous possibilities in the exploitation of critical embedded systems that utilize configurable logic such as FPGAs or complex programmable logic devices (CPLDs). Similar techniques can be applied to other families of FPGAs that do not support hardware-based authentication mechanisms [12] to remove functionality, alter behavior, and add functionality [6], all without RTL reconstruction. We have created a framework to support this analysis and modification of bitstreams.

4.1 TAm Bitstream Reversing

Due to the scope and purpose of this paper, we will not be discussing the FPGA architecture in detail. We encourage readers to refer to [3, 6, 24, 28, 29] for an in-depth look at FPGA architecture, configuration, and bitstream structure. Instead, our focus will be on TAm exploitation.



(a) Boot Process Signals

(b) Targeted Signals

Figure 6: Targeted Signals in Secure Boot Process

To disable TAm from resetting the RP, the adversary must first reverse the configuration bitstream. Prior research [6, 10, 11, 13, 21, 26] aimed at reverse engineering proprietary bitstream file formats of FPGAs has shown that it is not currently feasible to fully reverse the entire bitstream of Xilinx FPGAs. In general, bitstream reverse engineering is a time-consuming and complicated task. In order to sidestep this fundamental problem, we focused on disabling the reset signal from the FPGA to the RP. We performed our initial analysis on a reference bitstream for the Xilinx Spartan-6 development board [27]. Our approach of reverse engineering the configuration bitstream works as follows:

1. Parse the bitstream into its configuration commands and configuration data [29].
2. Parse the configuration data into its configuration frames. There are three types of configuration frames:
   (a) Type 0: Core components: Configurable Logic Block(CLB), DSP, Input/Output Interconnect (IOI), clocking
   (b) Type 1: Block RAM
   (c) Type 2: Input/Output Buffer (IOB)
3. Extract the package layout information using a reference bitstream. The reference bitstream can be generated using the officially supported tools from Xilinx or unofficial tools such as RapidSmith2 [22]. The internal layout of all the FPGA resources are fixed per package per Xilinx device family series.
4. Create a correlation between these components and configuration bits in the TAm bitstream. IOB and IOI components form the input/output interface for the FPGA. Our independently developed correlation techniques are similar to the techniques mentioned in [13].

Through the above process, we determined the layout of a Spartan-6 FPGA. It is a two-dimensional layout with four rows, each row having its own structure of columns. Correlation between resources and their bitstream location is calculated once per FPGA family. The results of this process can be applied to other packages within the same family. Our framework provides interfaces where visualizations can be generated from the layout information to help with the reverse engineering of bitstreams by highlighting the resources used in the FPGA. Figure 8 shows the FPGA resources of one row based on our analysis from a reference bitstream for the Spartan-6 development board.

4.2 The TAm Achilles Heel: IOB/IOI

FPGAs in the Spartan-6 family are provided with a large number of resources, and reverse engineering all them is infeasible and error-prone. As mentioned in Section 2.3, the FPGA (i.e. the TAm) controls the other components in the system with its input/output pins. The number of IO pins in the Spartan-6 family ranges between 150 and 550, which is considerably less than tens of thousands of logic cells or CLBs, themselves consisting of a switch matrix and flip flops. The exact number of these resources for Xilinx Spartan-6 family is described in [28]. Therefore, the complexity of reverse engineering the bitstream encoding for only those IOBs and IOIs that are connected to the limited number of IO pins is significantly lower than reversing the logic performed by CLBs.

See Appendix A for the generic layout of a Spartan-6 device illustrating different resources and their connection to physical IO pins.

Analysis has shown that all the CLBs, BRAMs, and IOBs are grouped and laid out sequentially in the bitstream. Our framework creates the mapping between these input/output resources and their location in the bitstream as follows:

1. For every pin in the package:
   (a) Generate a bitstream while pulling the pin connected to an IOB high.
   (b) Generate a bitstream while pulling the same pin low.
   (c) Perform XOR of the bitstreams to determine the changes between the two.
   (d) Extract the encoding of that pin and its corresponding IOB and IOI in the bitstream.

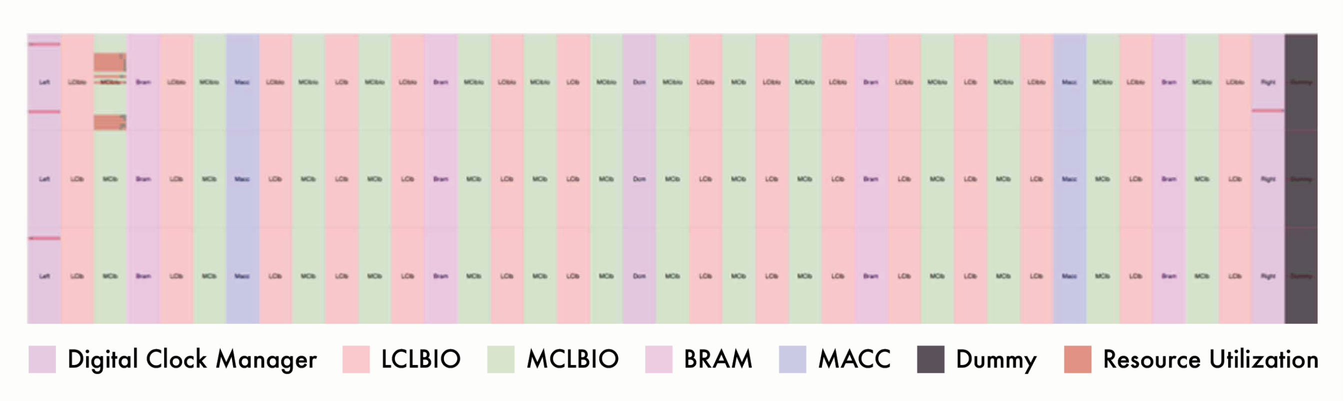


Figure 8: Visual representation of physical resource utilization of a row from a reference bitstream for Spartan-6 XC6SLX45T

We now can use trial and error to determine which IOB and IOIs are responsible for resetting the Cisco ASR 1001-X if unauthenticated firmware is loaded.

This novel approach of reverse engineering only these IO components allows us to control the behavior of the TAm and fully compromise the root of trust in Cisco's proprietary implementation.

4.3 TAm Bitstream Manipulation

For the Xilinx Spartan-6 bitstreams, FPGAs can be configured to do a configuration integrity check [29]. The Spartan-6 family uses a 22-bit CRC which we reverse engineered and incorporated into our framework. Disabling the CRC check is also possible as mentioned in [6, 29], but we have not tested this approach.

From our analysis, only 15 bytes of data in the targeted bitstream need to be modified in order to disable the FPGA's ability to reset the RP. The modification also disables the bitstream SPI flash from being updated again. Finally, after patching the bitstream, our framework must modify an additional 154 bits to fix-up the seven corresponding CRC commands with the correct CRC.

4.4 TAm Bitstream Update

As mentioned in Section 2.5, *fpga_mb_program* makes ioctl calls to the cpld1NG.ko driver that provides the capability to update the FPGA configuration bitstream. Our analysis found that the SPI flash on a Cisco ASR 1001-X contains two copies of the same bitstream. cpld1NG.ko is hardcoded to update only the second bitstream and lacks an API to update the first bitstream. As we could not determine how to force the TAm to select the second bitstream for its configuration during boot, we reverse engineered the cpld1NG.ko to determine the procedure to update the SPI flash. Since we obtained root privilege on the Linux shell using the privilege escalation vulnerability mentioned in Section 3.1, we were able to hijack an existing driver to write our own exploit by performing the following steps:

1. Allocate a buffer for the existing configuration bitstream.
2. Flush any cached data from previous reads of the bitstream SPI flash.
3. Read the SPI flash containing the configuration bitstream.
4. Apply the IOB/IOI and CRC patch to the read buffer.
5. Erase the whole bitstream SPI flash and write the patched buffer back to it.

5 Mitigations

The aforementioned vulnerabilities have been disclosed in coordination with Cisco and patches have either been released or are underway. However, we believe new vulnerabilities may still emerge as the Cisco proprietary secure boot process on the ASR 1001-X is implemented with a mutable root of trust and no hardware-based confidentiality and authentication mechanisms [12]. The Cisco implementation also fails to perform attestation of the integrity of the root of trust (i.e. the configuration bitstream of the FPGA) after secure boot, while the device is in service. These factors leave the device vulnerable to undetectable alteration of the root of trust via remote exploitation. A proper fix for the proprietary secure boot process requires hardware changes, however, as the lifetime of these network infrastructure devices can be one to two decades, we suggest addressing the flaws discussed in this paper as an initial step towards making these devices more secure.

We make several recommendations to future FPGA designs that will make the exploitation of the vulnerabilities described in this paper more challenging::

1. Use all available free space in the configuration bitstream of FPGA. The vendor toolchain can be modified

to fill the unused resources within the FPGA device with dummy logic [19]. This proposed scheme makes the hardware trojan attacks a non-trivial task.

2. Use encrypted I/O communication whenever possible and do not use an FPGA pin to control processor reset.
3. Use a verifiably immutable root of trust.
4. Use continuous runtime attestation to ensure the integrity of all the components of the trust chain.

6 Related Work

Attacks have been described against numerous secure boot implementations in recent years, such as in [5, 9, 25]. More recently, Han et al. [17] demonstrated a set of attacks against a modern Trusted Platform Module by exploiting its improper handling of power state changes. Techniques like these, which subvert the boot process by bypassing integrity checks, are necessary in the case of an immutable root of trust.

However, as the TAm resides in an FPGA, we also identify several recent advances in the study of FPGA reverse engineering and modification. Chakraborty et al. [6] first describe the insertion of a hardware trojan directly into an FPGA bitstream. Their technique, self-described as a Type-1 trojan, requires unused fabric (analogous to a code cave in software) within the FPGA in which to insert their malicious logic. Swierczynski [23] demonstrated the first bitstream modification attack against a commercial bitstream. This was achieved by modifying critical AES constants located in the BRAM section of the targeted bitstream. Most recently, Ender et al. [13] identified several challenges in developing bitstream modification tools and discuss similar efforts to those in this paper targeting a Xilinx Spartan-6 FPGA. While their research seeks to modify signal routing internally within the FPGA, we focus on a previously unexplored structure in the bitstream, the IOI directly connected to input/output pins, to alter the behavior of the system. This strategy, much like those in [4, 11, 21], relies on specific tools provided by the vendor. By focusing directly on the inputs and outputs of the FPGA, we are able to sidestep these requirements.

7 Conclusion

We presented an analysis of the Cisco Trust Anchor module (TAm) at an architectural level. The TAm is the root of trust that is used to implement the secure boot process in many of Cisco's enterprise switches, routers, and firewalls. The TAm is an FPGA-based root of trust security module that is independent of the main application processor. In order for any real-world implementation of the TAm to be effective, it must be *immutable* to the adversary before, during, and after the secure boot process. However, the analysis of the Cisco ASR1001-X router presented in this paper showed that the TAm was implemented using a Xilinx Spartan 6 FPGA. Since FPGAs are inherently reprogrammable components, the hardware design choice of using FPGAs to implement the TAm was a mismatch for its security requirement of *immutability*.

Having identified an architectural design flaw of the TAm, we presented a collection of direct FPGA bitstream analysis and modification techniques that demonstrate the feasibility, and relative simplicity, of bitstream-level FPGA manipulation attacks. Specifically, we focused our bitstream manipulation attack on altering the configuration of physical input/output pins on the FPGA. Since the number of physical I/O pins on an FPGA is vastly smaller than the number of logic blocks, our FPGA IOB based manipulations proved to be reliable and significantly less computationally complex than CLB reversing or RTL reconstruction. We conclude the case-study of the TAm bypass attack against the Cisco ASR1001-X by detailing a complete remotely exploitable attack chain that results in the persistent bypass of the TAm. Lastly, we presented several recommendations to system designers to secure both existing FPGA-based security modules as well as future designs.

While it is simple to recommend the use of more secure FPGAs going forward, this paper has demonstrated the importance of building security controls around the FPGA bitstream if one insists on building an immutable security module using hardware that is inherently reprogrammable.

References

[1] William A Arbaugh, David J Farber, and Jonathan M Smith. A secure and reliable bootstrap architecture. In *Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on*, pages 65–71. IEEE, 1997.

[2] ARM. ARM security technology - building a secure system using TrustZone technology. ARM Technical White Paper, 2009.

[3] Ronak Bajaj and Suhaib Fahmy. Mapping for maximum performance on FPGA DSP blocks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35:1–1, 01 2015.

[4] Florian Benz, André Seffrin, and Sorin A Huss. Bil: A tool-chain for bitstream reverse-engineering. In *22nd International Conference on Field Programmable Logic and Applications (FPL)*, pages 735–738. IEEE, 2012.

[5] Yuriy Bulygin, Andrew Furtak, and Oleksandr Bazhaniuk. A tale of one software bypass of windows 8 secure boot. *Black Hat USA*, 2013.

[6] Rajat Subhra Chakraborty, Indrasish Saha, Ayan Palchaudhuri, and Gowtham Kumar Naik. Hardware trojan insertion by direct modification of FPGA configuration bitstream. *IEEE Design & Test*, 30(2):45–54, 2013.

[7] Cisco. Cisco ios. https://www.cisco.com/c/en/us/products/ios-nx-os-software/index.html.

[8] Cisco. ROM Monitor overview. https://www.cisco.com/c/en/us/td/docs/routers/access/4400/hardware/installation/guide4400-4300/C4400_isr/rommon.pdf.

[9] Ang Cui and Rick Housley. BADFET: Defeating modern secure boot using second-order pulsed electromagnetic fault injection. In *11th USENIX Workshop on Offensive Technologies WOOT 17*, 2017.

[10] K. Dang Pham, E. Horta, and D. Koch. BITMAN: A tool and API for FPGA bitstream manipulations. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, pages 894–897, March 2017.

[11] Zheng Ding, Qiang Wu, Yizhong Zhang, and Linjie Zhu. Deriving an NCD file from an FPGA bitstream: Methodology, architecture and evaluation. *Microprocessors and Microsystems - Embedded Hardware Design*, 37:299–312, 2013.

[12] Saar Drimer. Authentication of FPGA bitstreams: Why and how. In *ARC*, 2007.

[13] Maik Ender, Pawel Swierczynski, Sebastian Wallat, Matthias Wilhelm, Paul Martin Knopp, and Christof Paar. Insights into the mind of a trojan designer: The challenge to integrate a trojan into the bitstream. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, ASPDAC '19, pages 112–119, New York, NY, USA, 2019. ACM.

[14] Unified Extensible Firmware Interface Forum. Unified extensible firmware interface specification - version 2.7. https://uefi.org/sites/default/files/resources/UEFI_Spec_2_7.pdf, May 2017.

[15] Anthony H Grieco, Chirag K Shroff, and Robert T Bell. Method and apparatus for securing cpus booted using attached flash memory devices, September 29 2015. US Patent 9,147,074.

[16] Trusted Computing Group. TCG TPM specification version 1.2 - part 1 design principles revision 116, 2011.

[17] Seunghun Han, Wook Shin, Jun-Hyeok Park, and HyoungChun Kim. A bad dream: subverting trusted platform module while you are sleeping. In *27th USENIX Security Symposium USENIX Security 18)*, pages 1229–1246, 2018.

[18] Intel. Hardware-enabled security powered by Intel® technology. https://www.intel.com/content/www/us/en/architecture-and-technology/trusted-execution-technology/trusted-execution-technology-security-paper.html.

[19] Behnam Khaleghi, Ali Ahari, Hossein Asadi, and Siavash Bayat Sarmadi. Fpga-based protection scheme against hardware trojan horse insertion using dummy logic. *IEEE Embedded Systems Letters*, 7:46–50, 2015.

[20] Microsemi. Secure boot. https://www.microsemi.com/product-directory/security/4885-secure-boot.

[21] Jean-Baptiste Note and Éric Rannaud. From the bitstream to the netlist. In *Proceedings of the 16th International ACM/SIGDA Symposium on Field Programmable Gate Arrays*, FPGA '08, pages 264–264, New York, NY, USA, 2008. ACM.

[22] RapidSmith2. Rapidsmith2 open source FPGA CAD tool. https://github.com/byuccl/RapidSmith2, 2017.

[23] Pawel Swierczynski. *Bitstream-based attacks against reconfigurable hardware*. PhD thesis, Ruhr-Universität Bochum, 2018.

[24] T. J. Todman, G. A. Constantinides, S. J. E. Wilton, O. Mencer, W. Luk, and P. Y. K. Cheung. Reconfigurable computing: architectures and design methods. *IEE Proceedings - Computers and Digital Techniques*, 152(2):193–207, March 2005.

[25] Rafal Wojtczuk and Corey Kallenberg. Attacks on UEFI security. In *Proc. 15th Annu. CanSecWest Conf.(CanSecWest)*, 2015.

[26] Project X-Ray. Project X-Ray Xilinx series 7 bitstream documentation. https://symbiflow.github.io/prjxray-db/, 2017.

[27] Xilinx. Spartan6-devboard. https://www.xilinx.com/products/boards-and-kits/dk-s6-embd-g.html.

[28] Xilinx. Xilinx Spartan-6 family overview. https://www.xilinx.com/support/documentation/data_sheets/ds160.pdf, October 2011.

[29] Xilinx. Spartan-6 FPGA configuration. https://www.xilinx.com/support/documentation/user_guides/ug380.pdf, 2019.

8 Appendix A: Xilinx Spartan-6 Family Generic Layout

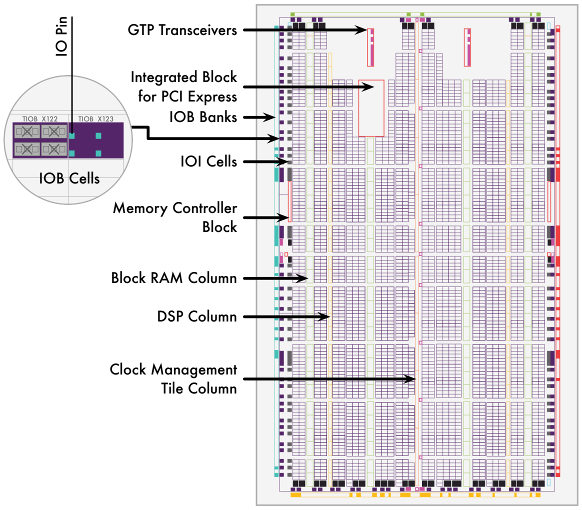


Figure 9: Spartan-6 Family Generic Layout

9 Appendix B: Previously Discovered Privilege Escalation Vulnerabilities

CVE ID	Severity
CVE-2019-1756	High
CVE-2019-1743	High
CVE-2018-0315	Critical
CVE-2018-0176	High
CVE-2018-0477	High
CVE-2018-0481	High
CVE-2018-0169	High

Table 1: Privilege Escalation Vulnerabilities

10 Appendix C: Discovered Vulnerabilities and Affected Products

CVE ID	Severity
CVE-2019-1649	High
CVE-2019-1862	High

Table 2: Discovered Vulnerabilities

Table 3: Affected Devices

Product	Cisco Bug ID	Fixed Release Availability
Cisco ASA 5506-X with FirePOWER Services	CSCvn77246	Firmware Release 1.1.15 (image name: asa5500-firmware-1115.SPA) (Available)
Cisco ASA 5506H-X with FirePOWER Services	CSCvn77246	Firmware Release 1.1.15 (image name: asa5500-firmware-1115.SPA) (Available)
Cisco ASA 5506W-X with FirePOWER Services	CSCvn77246	Firmware Release 1.1.15 (image name: asa5500-firmware-1115.SPA) (Available)
Cisco ASA 5508-X with FirePOWER Services	CSCvn77246	Firmware Release 1.1.15 (image name: asa5500-firmware-1115.SPA) (Available)
Cisco ASA 5516-X with FirePOWER Services	CSCvn77246	Firmware Release 1.1.15 (image name: asa5500-firmware-1115.SPA) (Available)
Cisco Firepower 2100 Series	CSCvn77248	Cisco Firepower Threat Defense 6.2.2.5 (Available) Cisco Firepower Threat Defense 6.2.2.12 (Available) Cisco Firepower Threat Defense 6.3.0.3 (Available) Cisco Firepower Threat Defense 6.4.0.1 (Available)
Cisco Firepower 4000 Series	CSCvn77249	Firmware bundle package v1.0.18 with ROMMON rev 1.0.15 and FPGA rev 2.0: (Image Names: fxos-k9-fpr4k-firmware.1.0.18.SPA and fxos-k9-fpr9k-firmware.1.0.18.SPA) (Available)
Cisco Firepower 9000 Series	CSCvn77249	Firmware bundle package v1.0.18 with ROMMON rev 1.0.15 and FPGA rev 2.0: (Image Names: fxos-k9-fpr4k-firmware.1.0.18.SPA and fxos-k9-fpr9k-firmware.1.0.18.SPA) (Available)
10Gbps Optical Encryption Line Card for the Cisco NCS 2000 Series and Cisco ONS 15454 MSTP (15454-M-WSE-K9)	CSCvn77191	11.1 (Jul 2019)
CBR-8 Converged Broadband Router	CSCvn77185	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco 1-Port Gigabit Ethernet WAN Network Interface Module (NIM-1GE-CU-SFP)	CSCvn77218	Cisco IOS XE Software Release 16.3.9 (Jul 2019) Cisco IOS XE Software Release 16.6.7 (Oct 2019) Cisco IOS XE Software Release 16.9.4 (Aug 2019) Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco 1120 Connected Grid Router	CSCvn89140	Cisco IOS Software Release 15.9(3)M (Aug 2019) Cisco IOS Software Release 15.8(3)M3 (Aug 2019) Cisco IOS Software Release 15.7(3)M5 (Sep 2019) Cisco IOS Software Release 15.6(3)M7 (Sep 2019)

Table 4: Affected Devices (continued)

Cisco 2-Port Gigabit Ethernet WAN Network Interface Module (NIM-2GE-CU-SFP)	CSCvn77218	Cisco IOS XE Software Release 16.3.9 (Jul 2019) Cisco IOS XE Software Release 16.6.7 (Oct 2019) Cisco IOS XE Software Release 16.9.4 (Aug 2019) Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco 3000 Series Industrial Security Appliances	CSCvn89146	Firmware release 1.0.05 (image name: isa3000-firmware-1005.SPA) (Available)
Cisco 4000 Series Integrated Services Router Packet 1024-Channel High-Density Voice DSP Module (SM-X-PVDM-1000)	CSCvn77212	Cisco IOS XE Software Release 16.3.9 (Jul 2019) Cisco IOS XE Software Release 16.6.7 (Oct 2019) Cisco IOS XE Software Release 16.9.4 (Aug 2019) Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco 4000 Series Integrated Services Router Packet 2048-Channel High-Density Voice DSP Module (SM-X-PVDM-2000)	CSCvn77212	Cisco IOS XE Software Release 16.3.9 (Jul 2019) Cisco IOS XE Software Release 16.6.7 (Oct 2019) Cisco IOS XE Software Release 16.9.4 (Aug 2019) Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco 4000 Series Integrated Services Router Packet 3080-Channel High-Density Voice DSP Module (SM-X-PVDM-3000)	CSCvn77212	Cisco IOS XE Software Release 16.3.9 (Jul 2019) Cisco IOS XE Software Release 16.6.7 (Oct 2019) Cisco IOS XE Software Release 16.9.4 (Aug 2019) Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco 4000 Series Integrated Services Router Packet 768-Channel High-Density Voice DSP Module (SM-X-PVDM-500)	CSCvn77212	Cisco IOS XE Software Release 16.3.9 (Jul 2019) Cisco IOS XE Software Release 16.6.7 (Oct 2019) Cisco IOS XE Software Release 16.9.4 (Aug 2019) Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco 4221 Integrated Services Router	CSCvn77153	Utility File Name: isr4200_cpld_update_v1.1_SPA.bin (Jun 2019)
Cisco 4321 Integrated Services Router	CSCvn77156	Utility File Name: isr4300_cpld_update_v1.1_SPA.bin (Jun 2019)
Cisco 4331 Integrated Services Router	CSCvn77156	Utility File Name: isr4300_cpld_update_v1.1_SPA.bin (Jun 2019)
Cisco 4351 Integrated Services Router	CSCvn77156	Utility File Name: isr4300_cpld_update_v1.1_SPA.bin (Jun 2019)
Cisco 4431 Integrated Services Router	CSCvn77155	Utility File Name: isr4400_cpld_update_v1.1_SPA.bin (Jun 2019)
Cisco 4451-X Integrated Services Router	CSCvn77155	Utility File Name: isr4400_cpld_update_v1.1_SPA.bin (Jun 2019)
Cisco 4461 Integrated Services Router	CSCvn77154	Utility File Name: isr4400_cpld_update_v1.1_SPA.bin (Jun 2019)
Cisco 5000 Series Enterprise Network Compute System	CSCvn77150	Release no. TBD (Jul 2019)
Cisco 809 Industrial Integrated Services Routers	CSCvn89138	Cisco IOS Software Release 15.8(3)M2a (May 2019) Cisco IOS Software Release 15.7(3)M4b (May 2019) Cisco IOS Software Release 15.6(3)M6b (May 2019)

Table 5: Affected Devices (continued 2)

Cisco ASR 1000 Embedded Services Processor, 200G (ASR1000-ESP200)	CSCvn77159	Release no. TBD (Jun 2019)
Cisco ASR 1000 Fixed Ethernet Line Card (6x10GE) (ASR1000-6TGE)	CSCvn89144	Release no. TBD (Jun 2019)
Cisco ASR 1000 Fixed Ethernet Line Card, 2x10GE + 20x1GE (ASR1000-2T53X1GE)	CSCvn89144	Release no. TBD (Jun 2019)
Cisco ASR 1000 Series 100-Gbps Embedded Services Processor (ASR 1000-ESP100)	CSCvn77160	Release no. TBD (Jun 2019)
Cisco ASR 1000 Series Modular Interface Processor (ASR1000-MIP100)	CSCvn77158	Release no. TBD (Jun 2019)
Cisco ASR 1000 Series Route Processor 3 (Cisco ASR1000-RP3)	CSCvn77167	Release no. TBD (Jun 2019)
Cisco ASR 1001-HX Router	CSCvn77162	ASR1K-fpga_prog.16.0.0.xe.bin (Available)
Cisco ASR 1001-X	CSCvn89145	ASR1K-fpga_prog.16.0.0.xe.bin (Available)
Cisco ASR 1002-HX Router	CSCvn77166	ASR1K-fpga_prog.16.0.0.xe.bin (Available)
Cisco ASR 900 Series Route Switch Processor 2 - 128G, Base Scale (A900-RSP2A-128)	CSCvn77168	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco ASR 900 Series Route Switch Processor 2 - 64G, Base Scale (A900-RSP2A-64)	CSCvn77168	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco ASR 900 Series Route Switch Processor 3 - 200G, Large Scale (A900-RSP3C-200)	CSCvn77169	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco ASR 900 Series Route Switch Processor and Controller 400G (A900-RSP3C-400/W)	CSCvn77169	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco ASR 920 Series Aggregation Services Routers 10GE and 2-10GE - Passively Cooled DC model (ASR-920-10SZ-PD), Cisco ASR920 Series - 20GE SFP, 4Cu and 4-10GE: Modular PSU (ASR-920-20SZ-M)	CSCvn77171	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco ASR 920 Series Aggregation Services Routers 12 x 1/10GE SFP, AC Model (ASR-920-12SZ-A)	CSCvn77171	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco ASR 920 Series Aggregation Services Routers 12 x 1/10GE SFP, DC Model (ASR-920-12SZ-D)	CSCvn77171	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco ASR 920 Series Aggregation Services Routers 12GE and 2-10GE - AC model (ASR-920-12CZ-A)	CSCvn77171	Cisco IOS XE Software Release 16.12.1 (Jul 2019)

Table 6: Affected Devices (continued 3)

Cisco ASR 920 Series Aggregation Services Routers 12GE and 2-10GE - DC model (ASR-920-12CZ-D)	CSCvn77171	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco ASR 920 Series Aggregation Services Routers 24GE Copper and 4-10GE – Modular PSU (ASR-920-24TZ-IM)	CSCvn77172	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco ASR 920 Series Aggregation Services Routers 24GE Copper and 4-10GE – Modular PSU (ASR-920-24TZ-M)	CSCvn77172	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco ASR 920 Series Aggregation Services Routers 24GE Fiber and 4-10GE – Modular PSU (ASR-920-24SZ-M)	CSCvn77172	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco ASR 920 Series Aggregation Services Routers 2GE and 4-10GE - AC model (ASR-920-4SZ-A)	CSCvn77171	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco ASR 920 Series Aggregation Services Routers 2GE and 4-10GE - DC model (ASR-920-4SZ-D)	CSCvn77171	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco ASR 920 Series Aggregation Services Routers Conformal Coated - 12GE and 4-10GE, 1 IM Slot (ASR-920-12SZ-IM-CC), Cisco ASR920 Series - 12GE and 4-10GE, 1 IM slot (ASR-920-12SZ-IM)	CSCvn77170	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco ASR 9900 Route Processor 3 for Packet Transport (A99-RP3-TR)	CSCvn77175	Cisco IOS XR Software Release 7.0.1 (Jul 2019)
Cisco ASR 9900 Route Processor 3 for Service Edge (A99-RP3-SE)	CSCvn77175	Cisco IOS XR Software Release 7.0.1 (Jul 2019)
Cisco Catalyst 6800 16-port 10GE with Integrated DFC4-XL (C6800-16P10G-XL)	CSCvn77182	Cisco IOS XE Software Release 15.5(1)SY4 (Sep 2019)
Cisco Catalyst 6800 32-port 10GE with Dual Integrated Dual DFC4-XL (C6800-32P10G-XL)	CSCvn77182	Cisco IOS XE Software Release 15.5(1)SY4 (Sep 2019)
Cisco Catalyst 6800 8-port 10GE with Integrated DFC4-XL (C6800-8P10G-XL)	CSCvn77182	Cisco IOS XE Software Release 15.5(1)SY4 (Sep 2019)
Cisco Catalyst 6800 8-port 40GE with Dual Integrated Dual DFC4-EXL (C6800-8P40G-XL)	CSCvn77182	Cisco IOS XE Software Release 15.5(1)SY4 (Sep 2019)
Cisco Catalyst 6800 Series Supervisor Engine 6T XL	CSCvn77181	Cisco IOS XE Software Release 15.5(1)SY4 (Sep 2019)
Cisco Catalyst 6816-X-Chassis (Standard Tables) (C6816-X-LE)	CSCvn77183	Cisco IOS Software Release 15.5(1)SY4 (Sep 2019)
Cisco Catalyst 6824-X-Chassis and 2 x 40G (Standard Tables) (C6824-X-LE-40G)	CSCvn77183	Cisco IOS Software Release 15.5(1)SY4 (Sep 2019)
Cisco Catalyst 6832-X-Chassis (Standard Tables) (C6832-X-LE)	CSCvn77183	Cisco IOS Software Release 15.5(1)SY4 (Sep 2019)

Table 7: Affected Devices (continued 4)

Cisco Catalyst 6840-X-Chassis and 2 x 40G (Standard Tables) (C6840-X-LE-40G)	CSCvn77183	Cisco IOS Software Release 15.5(1)SY4 (Sep 2019)
Cisco Catalyst 9300 Series Switches	CSCvn77209	Utility name: cat9k_iosxe.16.00.00fpgautility.SPA.bin (Available)
Cisco Catalyst 9500 Series High-Performance Switch with 24x 1/10/25G Gigabit Ethernet + 4x 40/100G Uplink (C9500-24Y4C)	CSCvn89150	Utility name: cat9k_iosxe.16.00.00fpgautility.SPA.bin (Available)
Cisco Catalyst 9500 Series High-Performance Switch with 32x 100 Gigabit Ethernet (C9500-32C)	CSCvn89150	Utility name: cat9k_iosxe.16.00.00fpgautility.SPA.bin (Available)
Cisco Catalyst 9500 Series High-Performance Switch with 32x 40 Gigabit Ethernet (C9500-32QC)	CSCvn89150	Utility name: cat9k_iosxe.16.00.00fpgautility.SPA.bin (Available)
Cisco Catalyst 9500 Series High-Performance Switch with 48x 1/10/25G Gigabit Ethernet + 4x 40/100G Uplink (C9500-48Y4C)	CSCvn89150	Utility name: cat9k_iosxe.16.00.00fpgautility.SPA.bin (Available)
Cisco Catalyst 9500 Series Switch with 12x 40G Gigabit Ethernet (C9500-12Q)	CSCvn77220	Utility name: cat9k_iosxe.16.00.00fpgautility.SPA.bin (Available)
Cisco Catalyst 9500 Series Switch with 16x 1/10G Gigabit Ethernet (C9500-16X)	CSCvn77220	Utility name: cat9k_iosxe.16.00.00fpgautility.SPA.bin (Available)
Cisco Catalyst 9500 Series Switch with 24x 40G Gigabit Ethernet (C9500-24Q)	CSCvn77220	Utility name: cat9k_iosxe.16.00.00fpgautility.SPA.bin (Available)
Cisco Catalyst 9500 Series Switch with 40x 1/10G Gigabit Ethernet (C9500-40X)	CSCvn77220	Utility name: cat9k_iosxe.16.00.00fpgautility.SPA.bin (Available)
Cisco Catalyst 9600 Supervisor Engine-1	CSCvn95346	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco Catalyst 9800-40 Wireless Controller	CSCvn77165	C9800-40_fpga_prog.16.0.0.xe.bin (Available)
Cisco Catalyst 9800-80 Wireless Controller	CSCvn77163	C9800-80_fpga_prog.16.0.0.xe.bin (Available)
Cisco IC3000 Industrial Compute Gateway	CSCvp42792	Firmware Release 1.0.2 (image name IC3000-K9-1.0.3.SPA) (Jul 2019)
Cisco MDS 9000 Family 24/10 SAN Extension Module (DS-X9334-K9)	CSCvn77141	Cisco NX-OS Software Release 8.4.1 (June 2019)
Cisco NCS 200 Series 10/40/100G MR Muxponder (NCS2K-MR-MXP-K9)	CSCvn77191	11.1 (Jul 2019)
Cisco NCS 5500 12X10, 2X40 2XMPA Line Card Base (NC55-MOD-A-S)	CSCvn77202	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS 5500 Series 24 Ports of 100GE and 12 Ports of 40GE High-Scale Line Card (NC55-24H12F-SE)	CSCvn77202	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS 5500 Series 36 ports of 100GE High-Scale Line Card (NC55-36X100G-A-SE)	CSCvn77202	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS 5504 Fabric Card (NC55-5504-FC)	CSCvn77202	Cisco IOS XR Software Release 7.1.1 (Nov 2019)

Table 8: Affected Devices (continued 5)

Cisco NCS 5516 Fabric Card (NC55-5516-FC)	CSCvn77202	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS 55A2 Fixed 24X10G + 16X25G MPA Chassis (NCS-55A2-MOD-S)	CSCvn77201	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS 55A2 Fixed 24X10G + 16X25G MPA Chassis, Temperature Hardened (NCS-55A2-MOD-HD-S)	CSCvn77201	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS 55A2 Fixed 24X10G + 16X25G MPA Chassis, Temperature Hardened with Conformal Coating (NCS-55A2-MOD-HX-S)	CSCvn77201	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS 55A2 Fixed 24X10G + 16X25G MPA Scale Chassis (NCS-55A2-MOD-SE-S)	CSCvn77201	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS 55A2 Fixed 24X10G + 16X25G MPA Scale Chassis, Temperature Hardened with Conformal Coating (NCS-55A2-MOD-SE-H-S)	CSCvn77201	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS5501 - 40x10G and 4x100G Scale Chassis (NCS-5501-SE)	CSCvn77201	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS5501 Fixed 48x10G and 6x100G Chassis (NCS-5501)	CSCvn77201	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS5502 - 48x100G Scale Chassis (NCS-5502-SE)	CSCvn77201	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS5502 Fixed 48x100G Chassis (NCS-5502)	CSCvn77201	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS55A1 Fixed 24x100G Chassis (NCS-55A1-24H)	CSCvn77201	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS55A1 Fixed 36x100G Base Chassis (NCS-55A1-36H-S)	CSCvn77201	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco NCS55A1 Fixed 36x100G Scale Chassis (NCS-55A1-36H-SE)	CSCvn77201	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco Network Convergence System 1002	CSCvn77219	Cisco IOS XR Software Release 7.0.1 (Jul 2019)
Cisco Network Convergence System 5001	CSCvn77207	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco Network Convergence System 5002	CSCvn77205	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco Network Convergence System 5500 Series: 1.2-Tbps IPoDWDM Modular Line Card (NC55-6X200-DWDM-S)	CSCvn77202	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco Network Convergence System 5500 Series: 36X100G MACsec Modular Line Cards (NC55-36X100G-S)	CSCvn77202	Cisco IOS XR Software Release 7.1.1 (Nov 2019)
Cisco Nexus 31108PC-V, 48 SFP+ and 6 QSFP28 ports (N3K-C31108PC-V)	CSCvn77245	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Cisco Nexus 31108TC-V, 48 10Gbase-T RJ-45 and 6 QSFP28 ports (N3K-C31108TC-V)	CSCvn77245	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Cisco Nexus 3132C-Z Switches (N3K-C3132C-Z)	CSCvn77245	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Cisco Nexus 3264C-E Switches (N3K-C3264C-E)	CSCvn77245	Cisco NX-OS Software Release 9.3(2) (Aug 2019)

Table 9: Affected Devices (continued 6)

Cisco Nexus 7000 M3-Series 48-Port 1/10G Ethernet Module (N7K-M348XP-25L)	CSCvn77141	Cisco NX-OS Software Release 8.4.1 (June 2019)
Cisco Nexus 7700 M3-Series 12-Port 100G Ethernet Module (N77-M312CQ-26L)	CSCvn77141	Cisco NX-OS Software Release 8.4.1 (June 2019)
Cisco Nexus 7700 M3-Series 24-Port 40G Ethernet Module (N7K-M324FQ-25L)	CSCvn77141	Cisco NX-OS Software Release 8.4.1 (June 2019)
Cisco Nexus 7700 M3-Series 48-Port 1/10G Ethernet Module (N77-M348XP-23L)	CSCvn77141	Cisco NX-OS Software Release 8.4.1 (June 2019)
Cisco Nexus 7700 Supervisor 3 (N77-SUP3E)	CSCvn77141	Cisco NX-OS Software Release 8.4.1 (June 2019)
Cisco Nexus 9332C ACI Spine Switch with 32p 40/100G QSFP28, 2p 1/10G SFP (N9K-C9332C)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Cisco Nexus 9364C ACI Spine Switch with 64p 40/100G QSFP28, 2p 1/10G SFP (N9K-C9364C)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Cisco Nexus 9500 4-Core/4-Thread Supervisor (N9K-SUP-A)	CSCvn77142	
Cisco Nexus 9500 6-Core/12-Thread Supervisor (N9K-SUP-B)	CSCvn77142	
Cisco Packet-over-T3/E3 Service Module (SM-X-1T3/E3)	CSCvn77147	Release no. TBD (Oct 2019)
Cisco cBR-8 Integrated CCAP 40G Remote PHY Line Card (CBR-CCAP-LC-40G-R)	CSCvn77184	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
Cisco cBR-8 Integrated CCAP Line Card includes 2 DS D3.1 Modules as well as 1 US D3.1 Module (CBR-LC-8D31-16U31)	CSCvn77184	Cisco IOS XE Software Release 16.12.1 (Jul 2019)
MDS 9700 48-Port 32-Gbps Fibre Channel Switching Module (DS-X9648-1536K9)	CSCvn77141	Cisco NX-OS Software Release 8.4.1 (June 2019)
Nexus 9200 with 36p 40G 100G QSFP28 (N9K-C9236C)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Nexus 9200 with 48p 1/10G/25G SFP+ and 6p 40G QSFP or 4p 100G QSFP28 (N9K-C92160YC-X)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Nexus 9200 with 48p 10/25 Gbps and 18p 100G QSFP28 (N9K-C92300YC)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Nexus 9200 with 48p 100M/1GT, 4p 10/25G & 2p 40/100G QSFP28 (N9K-C92348GC)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Nexus 9200 with 56p 40G QSFP+ and 8p 100G QSFP28 (N9K-C92304QC)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Nexus 9200 with 72p 40G QSFP+ (N9K-C9272Q)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Nexus 9300 with 48p 1/10G/25G SFP and 6p 40G/100G QSFP28, MACsec, and Unified Ports Capable (N9K-C93180YC-FX)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)

Table 10: Affected Devices (continued 7)

Nexus 9300 with 48p 100M/1G BASE-T, 4p 10/25G SFP28 and 2p 40G/100G QSFP28 (N9K-C9348GC-FXP)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Nexus 9300 with 48p 10G BASE-T and 6p 40G/100G QSFP28, MACsec Capable (N9K-C93108TC-FX)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Nexus 9K Fixed with 32p 100G QSFP28 (N9K-C9232C)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Nexus 9K Fixed with 48p 1/10G/25G SFP and 12p 40G/100G QSFP28 (N9K-C93240YC-FX2)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Nexus 9K Fixed with 48p 1/10G/25G SFP and 6p 40G/100G QSFP28 (N9K-C93180YC-EX)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Nexus 9K Fixed with 48p 10G BASE-T and 6p 40G/100G QSFP28 (N9K-C93108TC-EX)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Nexus 9K Fixed with up to 32p 40/50G QSFP+ or up to 18p 100G QSFP28 (N9K-C93180LC-EX)	CSCvn77143	Cisco NX-OS Software Release 9.3(2) (Aug 2019)
Supervisor A+ for Nexus 9500 (N9K-SUP-A+)	CSCvn77142	
Supervisor B+ for Nexus 9500 (N9K-SUP-B+)	CSCvn77142	
Analog Voice Network Interface Modules for Cisco 4000 Series ISRs (NIM-2FXO, NIM-4FXO, NIM-2FXS, NIM-4FXS, NIM-2FXS/4FXO, NIM-2FXSP, NIM-4FXSP, NIM-2FXS/4FXOP, NIM-4E/M, NIM-2BRI-NT/TE, NIM-4BRI-NT/TE)	CSCvn77151	Release no. TBD (Sep 2019)
Cisco 4000 Series Integrated Services Router T1/E1 Voice and WAN Network Interface Modules (NIM-1MFT-T1/E1, NIM-2MFT-T1/E1, NIM-4MFT-T1/E1, NIM-8MFT-T1/E1, NIM-1CE1T1-PRI, NIM-2CE1T1-PRI, NIM-8CE1T1-PRI)	CSCvn77152	Release no. TBD (Sep 2019)