Research Survey Paper Roy Lycke

The survey of the current research I carried out was a wide analysis of the current state of security in FPGAs. This means that my survey covers many of the topics facing FPGA security, but does not go in great depth on any one topic. Topics covered include current peemptive security practices for FPGAs, detection methods, and attack mitigation. This paper will discuss each paper in regards to its own research in a manner akin to an abstract as well as discussing each paper in regards to the other papers covered and the field of FPGA research in general.

Overview papers

This section looks at the papers I found that gave an overview of the current state of security regarding FPGAs. This section did not focus on implementation or problems of the current state, but rather an overview of what security for FPGAs means and the fields that affect it.

Trusted Hardware: Can it be Trustworthy
Irvine, C.E.; Levitt, K.;
Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE
Publication Year: 2007 , Page(s): 1 - 4
The first paper I read was the paper Trusted Hardware: Can it be Trustworthy to understand the overall issue of security with FPGAs and reconfigurable hardware. This paper explained the concept of trusted hardware and how attacks and vulnerabilities affect it. The authors discuss issues that can diminish the trustworthiness of the hardware such as design of the device, construction, assumed trusted hardware, and interaction with other devices. This paper sets up a good understanding of the key concepts that need understood when analysing the issues of security discusses in later papers.

FPGA Preemptive Security papers

The paper in this section discuss the current practices in use to increase the preemptive security of reconfigurable systems. These papers analyze issues inherent to the design of FPGA systems and offer means of mitigating any attacks before they can be formulated simply through better design and removal of untrustworthy processes.

Securing Boot of an Embedded Linux on FPGA
Devic, F.; Torres, L.; Badrignans, B.;

Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on
Digital Object Identifier: 10.1109/IPDPS.2011.141
Publication Year: 2011 , Page(s): 189 - 195
This paper discusses methods of securing the boot of a FPGA so that it can load its configuration, an OS in this paper, securely. Many points of attack exist for corrupting the configuration of the FPGA at start up but the authors focused the paper to loading from the non-volatile memory (NVM). The NVM is succeptable to attacks such as replay attacks and malicious bitstreams thus it must be secured. The NVM here is secured by encrypting the bitstream to the FPGA, then verifying it to be from the right send sender with the correct unique command key and another key verified for freshness via version key comparison. To ensure that the security of the bitstream was ensured other elements such as the flash memory and RAM were also updated to protect the keys or to implement asymmetric encryption. The authors then implemented their design on Xillinx FPGAs and found that the total performance and area overhead were minimal while providing the desired security performance. This validates the authors intentions of developing a secure boot of an embedded linux on an FPGA with regards to the NVM.

This study could have been extended to secure the boot from other attacks such as hardware Trojans or side channel attacks, but given the depth of focus on securing the NVM the authors made a stride in securing FPGAs.

Bitstream Encryption And Authentication with AES-GCM in Dynamically Reconfigurable Systems

Hori, Y.; Satoh, A.; Sakane, H.; Toda, K.;
Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on
Digital Object Identifier: 10.1109/FPL.2008.4629902
Publication Year: 2008 , Page(s): 23 - 28

This paper discusses the use of new encryption algorithms to prevent attacks to FPGAs akin to the previous paper. The concern with security discussed in this paper is the vulnerability for dynamic partially reconfigurable platforms to unintended bitstreams. Currently many FPGA systems do not properly secure incoming bitstreams, thus opening avenues of attack to those that would upload a different configuration for malicious intent. The authors propose their solution to this issues via the merger of several security techniques already in use. The authors created two security combinations; the first was a Galois/Counter Mode of operations (GCM) as a block cipher paired with AES, and secondly a SHA-2 secure hash algorithm paired with AES. Both of these approaches encrypt and authenticate the bitstreams coming to the FPGA. The application os encryption and authentication the the bitstream prevents a

great number of attacks such as spoofing or replay. The authors then tested the two algorithms implemented on several different Virtex boards. The results showed that the AES-GCM system out preformed the AES-SHA system on both speed (797 Mbps vs. 575 Mbps) and space (2687 slices vs. 2730 slices). Both encryption and authentication systems developed greatly increased throughput of the bitstream compared to current bitstream security implementations of the PowerPC and MicroBlaze systems. In conclusion, the authors created bitstream security system that not only provided necessary encryption and authentication, but also greater throughput when compared to current implementation methods.

Reconfigurable security architecture for embedded systems
Guy Gogniat, Tilman Wolf, and Wayne Burleson
Submitted to the Mobile Computing Hardware Architectures: Design and Implementation Design Symposium (MOCHA 2006)
January 4, 2006, Kauai, Hawaii, USA
Link: http://vcsg.ecs.umass.edu/essg/papers/MOCHASubmit.pdf (9-30-2011)

The approach to security for FPGAs presented by the authors in this paper was to create a new reconfigurable architecture that would ensure security on FPGAs and other integrated circuit devices. The problem of detecting hardware and software has been covered and in part solved, but those systems are static looking at only one section of the system and only one mode of security. This leads to security vulnerabilities if the implemented security is known and power issues for the device implementing the constantly active security system. This study performed by the authors was the development of a security architecture comprised of two parts: a monitoring network and and a reconfigurable security primitive system. The monitoring system is a low power option to constantly monitor the devices state and report to a security processor. If an attack is detected them the security processor engages mitigation protocols with application of the security primitives to stop the attack. The reconfigurable security primitives mentioned before are dedicated reconfigurable elements designed to optimize the performance of separate security tasks such as encryption/decryption or attack mitigation by allowing switching between loaded security primitive hardware during runtime. The advantage of the security primitives is a speed up of tasks that are programmed and ability to switch between different states as need be for different traits such as high security operation or low power consumption. The system developed was implemented on a Virtex-II board and the system was able to detect abnormal activities and switch to different modes for energy preservation. The development of this system did require additional hardware overhead, but given the gains in security and power management from other systems, well worth the tradeoff.

Detection papers

This section focuses on methods of detecting an attack on an FPGA. Once a device has been compromised or is in the process of being compromised, how would a user of the device be able to confirm the trustworthiness of the integrated circuit. The papers below offer methods to detect malicious hardware and code as pertaining to FPGAs.

A Power Analysis Based Approach to Detect Trojan Circuits

Li-Wei Wang; Hong-Wei Luo;
Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE), 2011 International Conference on
Digital Object Identifier: 10.1109/ICQR2MSE.2011.5976635
Publication Year: 2011 , Page(s): 380 - 384

This paper analyzes the method of detecting a Trojan or malicious components on chip by analyzing the power consumption of the chip over time. The analysis proposed considers both the static and dynamic power dissipation of the embedded hardware Trojan when detecting. The theory is that the power signature of the Trojan infected device will be different from those that are not infected (golden signatures), because and infected device will have a different circuit or even additional hardware consuming power due to the Trojan. Performing analysis on certified devices creates a golden signature power profile that all other devices can be compared to. The benefits of this approach are greatly reduced number of devices need to be destructively tested and increased speed of testing the devices. The authors of this paper tested this system on a FPGA testbed and found it to detect a high percentage of Trojan infected systems without performing any destructive analysis to the circuit and with very little change to the device under test.

RON: An on-chip Ring Oscillator Network for Hardware Trojan Detection

Xuehui Zhang; Tehranipoor, M.;
Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011
Publication Year: 2011 , Page(s): 1 - 6

This paper discusses the implementation of a new system to detect a Trojan in an FPGA or integrated circuit using multiple ring oscillators across the chip. This approach to analyzing a FPGA or IC is similar to the previous paper on Power Analysis Based Approach to Detect Trojan Circuits because ring oscillators detect changes in voltage, but unlike the previous paper the ring oscillators also detect changes in the timing of the device due to Trojan hardware. The man theory is that a ring oscillator around a particular block of the IC will have a golden signature of a good circuit. If an element of the circuit is changed at all, say a Trojan is inserted, then the output of the ring oscillator

will be different than the golden standard. By placing ring oscillators to cover your entire circuit a small change in any part of the circuit can be detected. The analysis performed with the ring oscillator network (RON) system is a simple outlier analysis where ICs under testing are compared to a golden standard for the same IC. The implementation of the of the ring oscillators is stated to be fairly light only requiring space for the oscillators' power strips, the counter, and LSFR; this makes the implantation of this Trojan detection system very light and attractive. Although if you want to get a more sensitive system you need to increase the number of ring oscillators you system uses and thus increase the hardware footprint. Still, the ROM system was found to be an effective method of detection Trojans on board after testing by the authors and verification of minimal overhead.

FPGA- based static analysis tool for detecting malicious binaries
Guinde, N.B.; Xin Tang; Sutaria, R.; Ziavras, S.G.; Manikopoulos, C.N.;
Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on
Volume: 2
Digital Object Identifier: 10.1109/ICCAE.2010.5451703
Publication Year: 2010 , Page(s): 639 - 643
The topic of this paper is the use of an FPGA for analyzing binary files in a static manner, i.e. not running the binary, to detect if they are malicious or not. Where as this doesn't directly link to FPGAs as the device being secured, if FPGAs were on an unsecured network for configuration updates, this would mitigate attacks on the destination FPGAs before they ever receive the malicious binaries. The method developed by the authors for detecting these malicious binaries was to analyze the files and detect byte patterns associated with malicious binaries. To obtain these malicious byte patterns the authors took 356 malicious files and 331 normal file and used them to train their system. This yielded 9 byte patterns for malicious binaries. These byte patterns are loaded onto an FPGA for high throughput analysis of malicious binaries over a network. The results of the study are not as promising as some of the other studies detected because only 83.5% of malicious binaries are detected and 20.7% or normal binaries are false positive detected. This system shows the possibility for very high throughput of network traffic, some of which may be destined for FPGAs, but still needs work to reduce the error rate of detection.

Where as this paper is innovative the high rate of false positives and missed malicious files sours it possibility as a great security system. That an the possibility that a new malicious binary format would go completely undetected until the system is trained for these new threats. Network security would improve the overall trustworthiness of the system, but currently this system has the possibility of causing more harm than good by rejecting valid binaries.

Attack mitigation papers

The articles highlighted in the section discuss particular attacks and methods of mitigating them. They are set apart from the preemptive articles simply for their focus on a single attack or few related attacks. These papers not only tell us how to prevent the attacks highlighted, but also give insight to areas of FPGA security weakness. Often there are trade-offs in security where only a subset of all possible attacks are prevented. Understanding how attacks are related can expand how many attacks can be prevented by a single mitigation strategy or how we can design multiple mitigation strategies that work in concert for security and functionality of the target hardware.

Multi-level Attacks: an Emerging Security Concern for Cryptographic Hardware
Ali, S.S.; Chakraborty, R.S.; Mukhopadhyay, D.; Bhunia, S.;
Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011
Publication Year: 2011 , Page(s): 1 - 4
This paper discusses the issue of an attack on hardware, including FPGAs, not from only one source but from multiple sources during the development of the device. The different levels of concern pointed out by this study are IC house, foundry, testing & development for IC design and application developer and testing & development for FPGA development. The authors discuss that an attack by any of these organizations can compromise the system, but if the attack is made by multiple levels working in conjunction or simply on their own independent attacks the attacks may be harder to detect. The reason the authors provide for this is due to the triggers set up across infected levels that have a minuscule chance for detection due to long trigger keys and power management of the Trojan. The paper concluded by inserting a multilevel Trojan onto a test FPGA and found that power consumption was below many detection systems and a trigger difficult to trigger by chance.

Covert and side channels due to processor Architecture
Zhenghong Wang; Ruby B. Lee;
Computer Security Applications Conference, 2006. ACSAC '06. 22nd Annual
Digital Object Identifier: 10.1109/ACSAC.2006.20
Publication Year: 2006 , Page(s): 473 - 482
The topic of this paper is the implementation of covert and side channels. The ones highlighted in this paper are manipulation of shared functional units to share data based on timing and monitoring shared cache and extrapolating key data from the access pattern. The authors of the paper have devised methods of extracting data from secured process by exploiting flaws in the processor hardware. In the case of manipulation of shared functional units, a functional unit is shared by a victim process and a spy process. The spy process runs its own activities, but due to the activities and message sent by the victim process there are timing variances in the output of

the spy process's data. This data can be used to decrypt the keys for encryption that the victim process is assumed to be running. The other attack is performed by using a shared cache where the spy fills up the cache and sees what is changed when the victim accesses cache. Based on the changes to the cache the spy can list the possible place in main memory that the victim drew from and derive what the key for an active encryption would be.

The authors are also to give solutions to their newly created security exploits. For the shared functional unit issue simply do not let two process share the same functional units at the same time. This leads to an issue of slowdown, but it does remove the exploit created by shared functional units. The other cache related exploit is solved by randomizing the cache indexes for each process so that a different process cannot map it to memory. The authors went on a deep discussion as how to perform this optimally and prevent the spy from ever learning by performing progressive randomization that only the owner process knows.

FPGA-Rootkits Hiding Malicious Code inside the Hardware

Kucera, M.; Vetter, M.;
Intelligent Solutions in Embedded Systems, 2007 Fifth Workshop on
Digital Object Identifier: 10.1109/WISES.2007.4408497
Publication Year: 2007 , Page(s): 262 - 272

This article focused on the implementation of rootkits in compromising reconfigurable hardware and FPGAs. The use and operation of rootkits is discussed by the authors, and this leads to their discussion as to how reconfigurable hardware can be susceptible to rootkits due to their high access level. The shortcomings of current hardware such as configuration stream encryption, authentication of the data stream, authentication of the configuration, configuration freshness, authorization of the device, and access control are all areas where the rootkit can be inserted, hide away, and comprimise the FPGA. Each of these the authors pointed out how they can be compromised or don't do enough to protect reconfigurable devices at the moment. Rootkits are then discussed in more depth as to how they can be implemented and used to attack FPGAs and reconfigurable hardware from a wide range of sources and for a large array of attacks. The authors cover possible solutions for securing the different areas highlighted from rootkit attacks in general, but do not give exact solutions or analysis of their possible solutions.

This paper provided a great deal of examples of the current short comings in security for FPGAs, often highlighting them with a simple method of attack that could be implemented. Even though many of the papers covered mitigate some of these security flaws caused by rootkits, others possible attacks such as using a light rootkit that hides

itself in the FPGA's memory elements or I/O during reconfiguration is hard to detect and remove if at all possible by the previously covered literature. This expands the possibilities for attack to the FPGA and give insights to areas that are not currently secured, but may need to be secured or re-secured utilizing new security methods.

Conclusion

The previous papers covered a large number of the current implementations of security for FPGAs in use today and in research. Across the many different security features we see that thousands of attacks can be prevented, mitigated, or at least detected using an array of hardware and software systems. The concern for security professionals is to combine as many security measures at once to improve the trustworthiness of their device while at the same time reducing the hardware footprint, performance overhead, power consumption, and cost of the security system. This in itself is a daunting task, but to improve security of reprogrammable hardware, like FPGAs, and other integrated circuits, these considerations must be considered.

The survey also shows that there are numerous security holes that can be exploited for malicious intent. Although many direct paths for malicious code or hardware Trojans have been locked down, there exist many indirect paths for attack, some of which have been discussed in this survey. Understanding and even creating new security exploits can teach us more about how to better implement transparent and strong security measures.