A Red Team/Blue Team Assessment of Functional Analysis Methods for Malicious Circuit Identification

*Adam Waksman †Jeyavijayan Rajendran *Matthew Suozzo *Simha Sethumadhavan

*Computer Architecture and Security Technologies Lab
Department of Computer Science
Columbia University
New York, NY, USA
{waksman,simha}@cs.columbia.edu
{ms4249@columbia.edu}

†Department of Electrical and Computer Engineering
New York University
New York, NY, USA
{jv.ece}@nyu.edu

ABSTRACT

Recent advances in hardware security have led to the development of FANCI (Functional Analysis for Nearly-Unused Circuit Identification), an analysis algorithm that identifies stealthy, malicious circuits within hardware designs that can perform backdoor operations to compromise security. Evaluations of such methods using benchmarks and academically known attacks are not always equivalent to the dynamic attack scenarios that can arise in the real world. For this reason, we apply a red team/blue team approach to stress-test the abilities of the FANCI prototype.

In the Embedded Systems Challenge (ESC) 2013, teams from research groups from multiple continents created designs with backdoors hidden in them as part of a red team effort to circumvent FANCI. Notably, these backdoors were not placed into *a priori* known designs. The red team was allowed to create arbitrary, unspecified designs. Two interesting results came out of this effort. The first was that FANCI was surprisingly resilient to this wide variety of attacks and was not circumvented by any of the stealthy backdoors created by the red teams. The second result is that frequent-action backdoors, which are non-stealthy backdoors, were often successful. These results emphasize the importance of combining FANCI with a reasonable degree of validation testing. The blue team efforts also exposed some areas where the FANCI prototype could be made more performant, which motivates further development of the prototype.

Categories and Subject Descriptors

B.6.2 [**Hardware**]: Logic Design—*Security and Trust*

Keywords

hardware; security; backdoors; functional analysis; intellectual property

1. INTRODUCTION

Hardware security and trust is a subject of rapidly increasing global concern [1, 2, 3, 4]. The economic drive for newer and better computing technology demands global cooperation and the sharing of intellectual property. However, as third-party IP is increasingly used by technology companies, trust issues are exacerbated [5, 6, 7]. A single malicious circuit, often called a backdoor or trojan, hidden in hardware design IP, can have catastrophic effects [8, 9].

Recently, static analysis was proposed as a method for combating third-party IP backdoors. A tool called FANCI (Functional Analysis for Nearly-Unused Circuit Identification) was developed that specifically targets a large class of backdoors in digital designs, called *stealthy* backdoors [10]. FANCI has performed extremely well on academic designs and benchmarks, but the current set of benchmarks is limited. As further evaluation, we present in this paper a red team/blue team approach to stress-testing FANCI. A variety of red teams designed different backdoors, both the types of stealthy attacks FANCI was designed to stop and alternative types of attacks. Our blue team applied FANCI to the red teams' designs and applied minimal manual analysis of the results of FANCI (about one hour per design) to attempt to track down the backdoor in each design.

The goal of this endeavor is on the one hand to stress-test the tool to see if it achieves everything the implemented algorithm should achieve and on the other hand to violate various axioms of the system to see how the tool responds to attacks it was not designed to handle. This was all performed as part of the 2013 Embedded Systems Challenge (ESC). We discuss the results and our observations, and we provide comments on future directions for functional analysis-based security.

2. OVERVIEW OF BLUE TEAM (DEFENSE) METHODS

The Columbia blue team used the recently proposed FANCI [10] tool. This tool is a prototype version of a new algorithm for detecting any and all stealthy logic with a digital hardware design or gatelist. The idea behind FANCI is the following. An organization wants to acquire a third-party hardware design, either as source code or as a gatelist. Some degree of validation and/or verification will be applied to the produced design to make sure that it is at least similar to that specification. A malicious provider can include backdoors in this third-party design, but by necessity the design is likely to be similar to the true specification. Thus, the added backdoor circuitry is likely to be *stealthy*, *i.e.,* hidden and largely un-

Algorithm 1 Compute Control Value

1: $count \leftarrow 0$
2: $c \leftarrow$ Column(w_1)
3: $T \leftarrow$ TruthTable(w_2)
4: **for all** Rows r in T **do**
5:     $x_0 \leftarrow$ Value of w_2 for $c = 0$
6:     $x_1 \leftarrow$ Value of w_2 for $c = 1$
7:     **if** $x_0 \neq x_1$ **then**
8:         $count$++
9:     **end if**
10: **end for**
11: $result \leftarrow \frac{count}{size(T)}$

Algorithm 2 How FANCI Flag Suspicious Wires in a Design

for all modules m **do**
    for all gates g in m **do**
        for all output wires w of g **do**
            $T \leftarrow$ TruthTable(FanInTree(w))
            $V \leftarrow$ Empty vector of control values
            for all columns c in T **do**
                Compute control of c
                Add control(c) to vector V
            end for
            Compute heuristics for V
            Denote w as suspicious or not suspicious
        end for
    end for
end for

used. It may not be the case that all stealthy circuits are malicious, but it is usually the case that all malicious circuits are stealthy (this observation is supported by analysis of hardware benchmarks) [10]. Thus, FANCI has been designed as a tool that identifies all stealthy circuits within a design. Once this has been done, we have a set of stealthy circuits which should be a superset of the set of malicious circuits. There can be false positives within that set, but if the tool works correctly, there should not be false negatives.

FANCI finds stealthy circuits by finding wires that do not often impact other wires they are connected to. We call these wires *weakly-affecting*, because they drive values into other wires but do not often change the digital outputs of those wires. We quantitatively measure the degree of impact one wire has on other using a metric called *control value*. The control value of an input w_1 on an output w_2 quantifies how much the truth table representing the computation of w_2 is influenced by the input column corresponding to w_1. Specifically, control value is a number between zero and one quantifying what fraction of the rows in the truth table for a circuit are directly influenced by w_1. Importantly, this is independent of particular tests inputs that might be supplied during validation. The algorithm to compute the control value of w_1 on w_2 is presented as Algorithm 1. We note that in step 3, we do not construct the exponentially large truth table. We instead construct the corresponding boolean function. Since the sizes of truth tables grow exponentially, we approximate control values by evaluating a constant-sized (parameterizable) subset of the rows in the truth table.

Once we have computed the control values for a given wire (a wire here means an output of any internal gate), we have a vector of floating point values that we can combine to make a judgement about stealth. We have found that using simple aggregating heuristics are effective for identifying stealthy wires. For ESC, we focus on two metrics. The first is the arithmetic mean, which is helpful usually for detecting wires that are part of the trigger. The other metric is called triviality and tends to be helpful in detecting payload wires. Triviality is described in full in [10] but can be thought of simply as the measure of how often the output is equal to zero (or by symmetry one). For example, a circuit that always outputs zero is completely trivial, while a circuit that outputs zero in half of cases is more normal. Our overall algorithm is summarized in Algorithm 2.

For an example of a backdoor circuit and how FANCI detects it, consider a classic 'ticking timebomb' trigger. A backdoor is designed to go off after 2^{40} cycles of operation. This is implemented with a 40-bit counter and a comparator against the hard-coded value $2^{40}-1$. The triviality of the output of the comparator is $\frac{1}{2^{40}}$. When looking at the payload circuit, it would have all 40 wires from the counter as inputs, meaning its vector of control values would contain at least 40 low values, making the median value $\frac{1}{2^{40}}$. In this case, both the trigger and payload circuits are caught due to the obviously stealthy nature of the triggering logic.

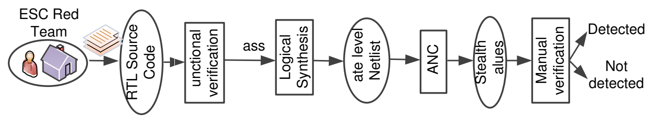

Figure 1: The steps involved in the contest, including both red team and blue team actions.

3. RED TEAM (ATTACK) METHODS AND COMPETITION RULES

We outline the rules of the contest. Since the goal of the contest was to attack FANCI at its axioms, we allowed several of the original rules to be broken. The main rule that we did not allow to be broken was that all attacks had to be digitally defined in the design. This means that we did not process any analog logic or allow for backdoors that require environmental or physical factors to turn on. We also only processed Verilog code, although adapting the FANCI tool for VHDL, Bluespec or other languages would be possible. Each submitted design contained exactly one backdoor. The blue team was given roughly three days to process and analyze the submissions. The flow of the contest is depicted in Figure 1.

The original rules were that a submission should include a single module with 10,000 or less gates, including both source code and gatelist, using standard logical gates and flip flops, as well as documentation of the module's specification. We ended up making a variety of exceptions, including 1) designs larger than 10,000 gates, 2) designs composed of many modules, 3) designs lacking source code, 4) designs lacking specification, 5) backdoors that rely on non-standard clock/reset usage, and 6) designs that use non-standard flip flops.

Red teams were allowed to choose or create their own baseline designs. They were not forced to insert backdoors into previously existing designs. This added an interesting dimension to the blue team analysis, because some of the designs being analyzed had not been seen before, and some of those were also without source code. The red teams did not intentionally obfuscate their gatelists, which is a direction that might be interesting in the future. Code obfuscation should not impact FANCI, since the tool ignores wire names, but it would add difficulty to the manual interpretation of the results that ultimately must happen.

Red teams were given points for backdoors that went undetected.

They received more points for stealthy backdoors, ranging from one point (for frequently-active backdoors) to five points (for backdoors stealthy enough to go unnoticed even by high quality validation testing). Tiebreaks were determined by a panel of judges based on how interesting the design choices were.

4. RESULTS AND EFFICACY OF BLUE TEAM ANALYSIS

The results of the blue team analysis were overall better than anticipated. While many of FANCI's axioms were targeted by attacks, the tool held up well. The stealthy attacks were detected with relative ease. We found that the best way to attack FANCI in this contest was to either use a frequently-active backdoor or to use a 'pathological pipeline.'[1] These are the two cases where a backdoor is more likely to bypass FANCI. In the first case, frequently-on backdoors are not stealthy. Hence, FANCI by design ignores them. In the second case, extreme pipelining obfuscates the results and stress-tests those portions of the tool prototype that are least developed, though some of these attacks were detected by FANCI nonetheless. We include further details for the three red teams that were deemed the best by the judges.

- **UCF Team:** The UCF team created three backdoors, which highlight different aspects of the FANCI prototype. They created two frequent-action backdoors (which were not detected) and one stealthy backdoor (which was detected). The stealthy backdoor uses a 66-bit cheat code that comes from a pseudo-random input source. This backdoor highlights the importance of using intermediate wires, which was one of the main contributions of FANCI. If FANCI only computed the impact of module inputs on module outputs, this type of backdoor might not be caught, but since we analyze all intermediate wires, noticing a 66-input AND gate is easy. The other two backdoors were frequent-action backdoors. One is turned on by the *reset* signal, and one is turned on only during the negative clock edge. Both of these are frequently enough active that they are not detectable. These are the types of backdoors that emphasize the need to use minimal validation testing in addition to FANCI. They also illuminate the fact that signals like reset and the clock, which might be given less attention during cursory validation testing, are still relevant, especially when dealing with an adversary (as opposed to simply looking for unintentional bugs).

- **Team NYU-Poly1 (NYU AD):** The NYU AD team submitted three backdoors on independent baseline designs, one that was submitted for the contest and two that we looked at post-competition. While FANCI detected the three attacks, two of them required non-trivial manual analysis. The first backdoor was designed into a DES accelerator and was built off of an aggregation of finite state machines (FSMs), somewhat similar to UCONN's approach[2]. They used the state machines to control round scheduling and weaken the strength of encryption, which would expose the ciphertext to software-based attacks upon release. This was a case where we were able to detect the trigger circuit because of the combinational logic that controls the state machines, though we note that this backdoor took more manual analysis to track down than the other ones we detected. The second backdoor was a similar round-scheduling attack, this time on XTEA. This one acquires its stealth by counting the number of one-bits in the input. The combinational circuit for doing the counting was flagged. The third backdoor also attacked crypto hardware, this time a cellular automata PRNG. This one was a sequence cheat code where the user supplies one bit of the trigger at a time. This attack would be difficult for validation to catch, because it hides on the negative edge of the clock (similar to one of the UCF attacks). However, since FANCI ignores clock timing, it can detect the trigger logic.

- **Team UT Tyler:** The UT Tyler team produced a backdoor in a stream cipher module. This attack is essentially one of the attacks suggested in the original FANCI paper [10]. The attack is a combination of a frequent-action backdoor and a pathological pipeline backdoor. The trigger fires frequently but not always, meaning the stealth scores are not particularly low. The design is also heavily pipelined, with roughly one flip flop between each pair of complex logic gates for critical portions of the design. Looking at the gatelist, it is immediately obvious that the design has been compromised, but identifying what the exact backdoor payload is can be quite hard, and FANCI does not easily detect this type of attack. As mentioned in the original paper, going after this type of attack reliably requires either validation tests or oversight from an integration engineer to notice the pathological nature of the design.

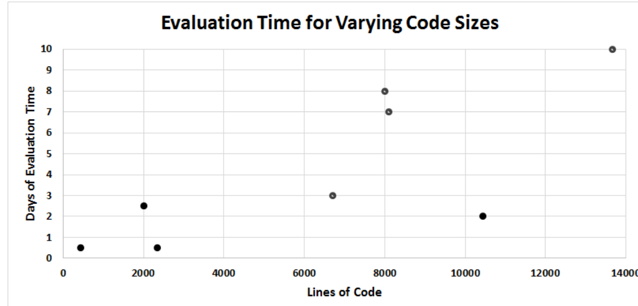

Figure 2: Trade-off between lines of code analyzed and runtime. The darker points denote designs that finished completely in the contest time frame. The lighter points represent designs that were analyzed partially for the contest. For those lighter points, the full runtime is estimated.

5. OBSERVATIONS ON RESULTS

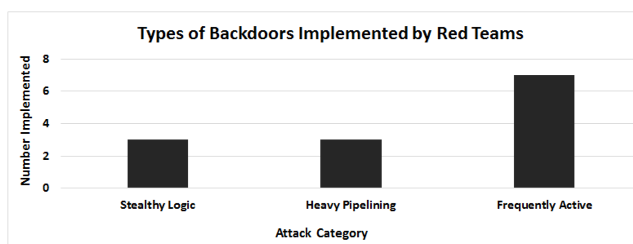

Figure 3: The amount of each type of backdoor implemented by the red teams.

We include a few observations and takeaways based on the results of the contest and our experiences.

- **Runtime and Scability** Figure 2 shows the runtime of the tool as a function of the number of lines of code in the various designs, using one primary design from each red team. Naturally, FANCI runs slower on larger designs, but the slowdown is more or less linear, which makes analysis feasible. These tests were performed

[1] Pathological pipeline is the term used in the original FANCI paper [10] to describe backdoor triggers that rely on heavy pipelining to obfuscate combinational logic.

[2] UCONN constructed an FSM-based, heavily pipelined backdoor that took a long time to find. The various teams worked independently and simultaneously.

on a single core of a commodity machine.

• **Attack Categorization:** A positive result of the contest was the discovery that many of the red teams designed attacks similar to the types we anticipated when first designing FANCI. In [10], we mentioned three general attack avenues against FANCI: frequently active (non-stealthy) backdoors, heavily pipelined backdoors, and false positive flooding. While the third option was not employed by the red teams, the first two were used by multiple teams. The breakdown is shown in Figure 3. This evidence supports that a more formal taxonomy could be derived from this initial survey. Additionally, it supports our belief that FANCI and validation testing should to be used together synergistically. Ideally, validation testing should be designed with the assumption that FANCI will detect anything stealthy. This would allow validation teams to focus their efforts on other avenues, such as some of the attacks we saw that target reset or the negative clock edge. Figure 4 shows the overall breakdown by team and also divides the frequently-active backdoors from the stealthy backdoors. While FANCI caught all the stealthy backdoors, it caught only a few of the frequently active ones, as expected.

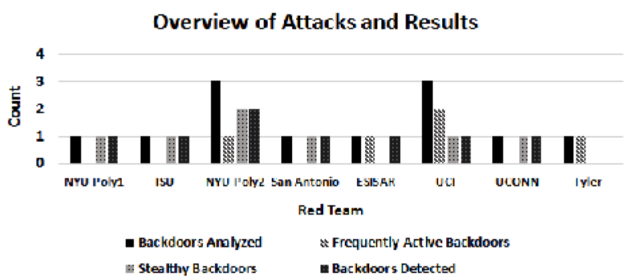

Figure 4: Overview of red team attacks and blue team results.

• **Algorithm vs. Implementation:** While the contest did not expose any deficiencies in the FANCI algorithm, the tool itself was stressed in some cases. Two issues stand out. First, runtime became an issue for large designs. Some modules would have taken more than the given three days to analyze, and so incomplete analysis runs were performed. The tool is configurable for this, allowing for hasty passes. However, in the future, parallelization could do a much better job of alleviating this problem. The second issue is the way the tool handles pipelining. The core of the tool works on combinational logic, so flip-flops have to be dealt with. We believe the best way to handle flip-flops is to treat them as identity gates, so that simply inserting dummy flip-flops does not hide stealthy logic. On the other hand, this can create loops in the logic, which have to be dealt with. For most cases, our tool currently treats flip-flops as a barrier and does not analyze past them. This did not prevent us from catching any stealthy backdoors in this contest, but it made data interpretation more difficult. Improving this aspect of the prototype would be beneficial.

• **Primary Takeaways:** The primary takeaways from the contest appear to be that FANCI handles a wide variety of backdoors and that effort should be spent on improving the usability of the prototype and on making static analysis and validation testing a tandem for future designs.

6. CONCLUSIONS

The ability to identify and understand hardware backdoors during the design phase using static analysis is critical for allowing continued use of third-party hardware intellectual property. Using a red team/blue team approach, we stress-tested FANCI, a state-of-the-art tool for identifying stealthy backdoors. Using this approach, we saw examples of stealthy attacks designed to target FANCI specifically, as well as examples of non-stealthy backdoors. Overall, the results of the contest were promising, as they demonstrated the effectiveness and flexibility of the FANCI approach. However, the results continue to highlight the importance of security awareness when integrating hardware designs, including FANCI, validation testing and reasonable oversight.

7. ACKNOWLEDGEMENTS

This work was funded in part by the NSF grant CNS 0958510 and Intel and supported by grants FA 99500910389 (AFOSR), FA 865011C7190 (DARPA), FA 87501020253 (DARPA), CCF/TC 1054844 (NSF) and a gift from Synopsys Inc. This work is also supported through an Alfred P. Sloan Foundation Fellowship and the Department of Defense NDSEG Fellowship.

References

[1] Sally Adee. The Hunt for the Kill Switch. *IEEE Spectrum Magazine*, 45(5):34–39, 2008.

[2] Marianne Swanson, Nadya Bartol, and Rama Moorthy. Piloting Supply Chain Risk Management Practices for Federal Information Systems. In *National Institute of Standards and Technology*, page 1, 2010.

[3] Adam Waksman and Simha Sethumadhavan. Silencing Hardware Backdoors. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, pages 49–63, Oakland, California, 2011.

[4] Hassan Salmani, Mohammad Tehranipoor, and Jim Plusquellic. New Design Strategy for Improving Hardware Trojan Detection and Reducing Trojan Activation Time. In *Hardware-Oriented Security and Trust, 2009. HOST '09. IEEE International Workshop on*, pages 66 –73, 2009.

[5] M Tehranipoor, H. Salmani, Xuehui Zhang, Xiaoxiao Wang, R. Karri, J. Rajendran, and K Rosenfeld. Trustworthy Hardware: Trojan Detection and Design-for-Trust Challenges. *Computer*, 44(7):66–74, 2011.

[6] Adam Waksman, Julianna Eum, and Simha Sethumadhavan. Practical, Lightweight Secure Inclusion of Third-Party Intellectual Property. In *Design and Test, IEEE*, pages 8–16, 2013.

[7] E. Love, Yier Jin, and Y Makris. Proof-Carrying Hardware Intellectual Property: A Pathway to Trusted Module Acquisition. *Information Forensics and Security, IEEE Transactions on*, 7(1):25–40, 2012.

[8] Samuel T. King, Joseph Tucek, Anthony Cozzie, Chris Grier, Weihang Jiang, and Yuanyuan Zhou. Designing and Implementing Malicious Hardware. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pages 5:1–5:8, Berkeley, CA, USA, 2008. USENIX Association.

[9] R. Karri, J. Rajendran, K Rosenfeld, and M Tehranipoor. Trustworthy Hardware: Identifying and Classifying Hardware Trojans. *Computer*, 43(10):39–46, 2010.

[10] Adam Waksman, Matthew Suozzo, and Simha Sethumadhavan. FANCI: Identification of Stealthy Malicious Logic Using Boolean Functional Analysis. In *ACM Conference on Computer and Communications Security*, pages 697–708, 2013.