Trustworthy IT Systems – where is the trust boundary?

Prof. Dr. Jean-Pierre Seifert
TU Berlin & Deutsche Telekom Laboratories, Berlin, Germany
jpseifert@sec.t-labs.tu-berlin.de
Agenda

- Recent news
- Problem visibility, relevance & complexity
- Real incidents
- New idea for problem solution
- A „sw only exploitation“ of the problem
- Q&A
Motivation
Motivation

High-tech warfare
Something wrong with our **** chips today
Kill switches are changing the conduct and politics of war
Apr 7th 2011 | from the print edition
Motivation

High-tech warfare

IN THE 1991 Gulf war Iraq’s armed forces used American-made colour photocopiers to produce their battle plans. That was a mistake. The circuitry in some of them contained concealed transmitters that revealed their position to American electronic-warfare aircraft, making bomb and missile strikes more precise. The operation, described by David Lindahl, a specialist at the Swedish Defence Research Agency, a government think-tank, highlights a secret front in high-tech warfare: turning enemy assets into liabilities.

The internet and the growing complexity of electronic circuitry have made it much easier to install what are known as “kill switches” and “back doors”, which may disable, betray or blow up the devices in which they are installed. Chips can easily contain 2 billion transistors, leaving plenty of scope to design a few that operate secretly. Testing even a handful of them for anomalies requires weeks of work.
Motivation

An obvious countermeasure is to keep critical defence equipment off the net. But that is only a partial solution. Chips can be designed to break down at a certain date. An innocent-looking component or even a bit of soldering can be a disguised antenna. When it receives the right radio signal, from, say, a mobile-phone network, aircraft or satellite, the device may blow up, shut down, or work differently.

Wesley Clark, a retired general who once headed NATO’s forces, says that “rampant” fears of kill switches make American-backed defence co-operation agreements a harder sell. David Kay, a notable United Nations weapons inspector in Iraq, bemoans “scepticism and paranoia”. You just can’t trust anybody these days, even in the weapons business.
Motivation

The Hunt for the Kill Switch

Are chip makers building electronic trapdoors in key military hardware? The Pentagon is making its biggest effort yet to find out.
Motivation

BY SALLY ADEE // MAY 2008

Last September, Israeli jets bombed a suspected nuclear installation in northeastern Syria. Among the many mysteries still surrounding that strike was the failure of a Syrian radar—supposedly state-of-the-art—to warn the Syrian military of the incoming assault. It wasn’t long before military and technology bloggers concluded that this was an incident of electronic warfare—and not just any kind.

Post after post speculated that the commercial off-the-shelf microprocessors in the Syrian radar might have been purposely fabricated with a hidden “backdoor” inside. By sending a preprogrammed code to those chips, an unknown antagonist had disrupted the chips’ function and temporarily blocked the radar.
Motivation

And the cool thing was that it seems that Israel was able to do this cyber attack from the air.

That ability of nonstealthy Israeli aircraft to penetrate without interference rests in part on technology, carried on board modified aircraft, that allowed specialists to hack into Syria’s networked air defense system, said U.S. military and industry officials in the attack’s aftermath.

Network raiders can conduct their invasion from an aircraft into a network and then jump from network to network until they are into the target’s communications loop. “Whether the network is wireless or wired doesn’t matter anymore,” says a U.S. industry specialist.
Motivation

BY SALLY ADEE // MAY 2008

Last September, Israeli jets bombed a suspected nuclear installation in northeastern Syria. Among the many mysteries still surrounding that strike was the failure of a Syrian radar--supposedly state-of-the-art--to warn the Syrian military of the incoming assault. It wasn't long before military and technology bloggers concluded that this was an incident of electronic warfare--and not just any kind.

Post after post speculated that the commercial off-the-shelf microprocessors in the Syrian radar might have been purposely fabricated with a hidden “backdoor” inside. By sending a preprogrammed code to those chips, an unknown antagonist had disrupted the chips’ function and temporarily blocked the radar.

That same basic scenario is cropping up more frequently lately, and not just in the Middle East, where conspiracy theories abound. According to a U.S. defense contractor who spoke on condition of anonymity, a “European chip maker” recently built into its microprocessors a kill switch that could be accessed remotely. French defense contractors have used the chips in military equipment, the contractor told IEEE Spectrum. If in the future the equipment fell into hostile hands, “the French wanted a way to disable that circuit,” he said. Spectrum could not confirm this account independently, but spirited discussion about it among researchers and another defense contractor last summer at a military research conference reveals a lot about the fever dreams plaguing the U.S. Department of Defense (DOD).
Pentagon, Darpa Fear Enemies Could Tamper With Chips

The U.S. military's heavy dependence on overseas-made chips has got the government thinking about how to prevent tampering prior to delivery.

"The shift from United States to foreign IC manufacture endangers the security of classified information embedded in chip designs; additionally, it opens the possibility that 'Trojan horses' and other unauthorized design inclusions may appear in unclassified integrated circuits used in military applications," the board's report said. It added, "Neither extensive electrical testing nor reverse engineering is capable of reliably detecting compromised microelectronics components."

Among the concerns are that ICs could be doctored crudely in design or manufacture to fail early—for example, by changing chemical composition, by reducing material thicknesses or placing wires too close together. Alternatively, chips could be engineered to misbehave under more specialized circumstances with functional blocks serving as embedded "Trojan horses." That raises the prospect of weapon systems that could appear to be in perfect working order during tests or deployment but which could "switch off" in combat.
Summary of motivation

Figure 1. Simple Trojan. During normal operation, the chip sends encrypted data to the output. When the Trojan is active, the chip bypasses the crypto module and sends plaintext to the output, betraying the chip designer’s intentions.
Richard A. Clarke, Robert K. Knake

Richard A. Clarke warned America once before about the havoc terrorism would wreak on our national security—and he was right. Now he warns us of another threat, silent but equally dangerous. Cyber War is a powerful book about technology, government, and military strategy; about criminals, spies, soldiers, and hackers.

It explains clearly and convincingly what cyber war is, how cyber weapons work, and how vulnerable we are as a nation and as individuals to the vast and looming web of cyber criminals.

This is the first book about the war of the future—cyber war—and a convincing argument that we may already be in peril of losing it.
Problem origin
Problem origin
Trend: Fabless

The fabless/foundry business model has grown to 16% of the U.S. chip industry. The trend is strongest in the leading process technology portion of the industry.

Source: SIA
Trend: Globalization of high-tech manufacturing

Common Platform technology

Common Platform Foundries
IBM East Fishkill, USA
Chartered Woodlands Campus, Singapore
Samsung S1 Line, South Korea

Common GDS to Multiple Fabs
Fab process synchronized

Development Site
IBM East Fishkill, USA

Capacity

Deutsche Telekom Laboratories
Problem visibility, relevance, and complexity
Malicious router “design study”

Designing and implementing malicious hardware

Samuel T. King, Joseph Tucek, Anthony Cozzie, Chris Grier, Weihang Jiang, and Yuanyuan Zhou
University of Illinois at Urbana Champaign, Urbana, IL 61801

Figure 1: Hardware differences when shadow mode is active.

Figure 2: Overview of the login attack.
Isn’t it easy to detect?

Designing and implementing malicious hardware

Samuel T. King, Joseph Tucek, Anthony Cozzie, Chris Grier, Weihang Jiang, and Yuanyuan Zhou

*University of Illinois at Urbana Champaign, Urbana, IL 61801*

<table>
<thead>
<tr>
<th>Processor</th>
<th>Logic gates</th>
<th>Lines of VHDL code</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline CPU</td>
<td>1,787,958</td>
<td>11,195</td>
</tr>
<tr>
<td>CPU + memory access</td>
<td>1,788,917</td>
<td>11,263</td>
</tr>
<tr>
<td>CPU + shadow mode</td>
<td>1,789,299</td>
<td>11,312</td>
</tr>
</tbody>
</table>

Table 1: This table summarizes the circuit-level impact of our IMPs compared to a baseline (unmodified) Leon3 processor. We show the impact of an IMP that includes our memory access mechanism and an IMP that includes our shadow mode mechanism.

Figure 3: Time perturbations are measured relative to the baseline (non-attack) tests.
Problem relevance

- What would happen if our routers as deployed inside our
  - Home
  - Government
  - Telcos
  
  would be infected with such backdoors?

- This scenario also shows that
  - the national security

  and not only the military would be threatened by such hw trojans.
Real incidents

- In May 2010, for example, the FBI’s Operation Network Raider seized more than 700 pieces of counterfeit Cisco network hardware and labels with an estimated retail value of more than $143 million.

- While that scheme was likely conceived for financial gain, designers of integrated circuits, or microchips, also need to protect military, financial, transportation and other critical digital infrastructure from Trojans inserted by intruders with other criminal or military intentions. Like the Trojan horses of Greek mythology, cyber Trojans appear to be harmless but instead steal information or harm a system once it is in operation.
Real incidents

Spotting Fake Chips in the Supply Chain

By Christopher Tarnovsky @seimcduktor & Gunter Ollmann @gollmann

In the information security world we tend to focus upon vulnerabilities that affect the application and network architecture layers of the enterprise and, very often, some notable physical devices. Through various interrogatory methods we can typically uncover any vulnerabilities that may be present and, through discussion with the affected business units, derive a relative statement of risk to the business as a whole.

IOActive.com

Blog Archive

▼ 2013 (18)
▼ April (2)

What Would MacGyver Do?
Spotting Fake Chips in the Supply Chain
Real incidents

Dissecting a ST19XT34 Microprocessor

In early 2012 samples of the ST19XT34 were ordered from www.hkinventory.com. The ST19XT34 is a secure microprocessor designed for very large volume and cost-effective secure portable applications (such as smartcards used within Chip&PIN technologies). The ST19X platform includes an internal Modular Arithmetic Processor (MAP) and DES accelerator - designed to speed up cryptographic calculations using Public Key Algorithms and Secret Key Algorithms.

The ST19XT34 chips that IOActive were charged to investigate were encapsulated within a standard SOIC package and were supposed to have 34kb of EEPROM.

Upon visual analysis the devices appeared to be correct. However, after decapsulation, it was clear that the parts provided were not what had been ordered.

In the above image we have a ‘fake’ ST19XT34 on the left with a sample of the genuine chip on the right. It is almost impossible to tell the left device was altered unless you have a known original part.
After decapsulation of the various parts it was easy to immediately recognize the difference between the two SOIC parts. The left ‘fake’ device was actually an ST ST19AF08 with the right being the genuine ST19XT34.

The ST19AF08 is a 600 nanometer 3 metal device (on left). It contains an 8 KB EEPROM.

The ST19XT34 is a 350 nanometer 3 metal device (on right). It contains a 34 KB EEPROM making the die
Real incidents

Breakthrough silicon scanning discovers backdoor in military chip

Sergei Skorobogatov, Christopher Woods

http://www.cl.cam.ac.uk/~sps32
email: sps32@cam.ac.uk

http://www.quovadislabs.com
email: chris@quovadislabs.com
Problem complexity
Boolean Circuits

- A Boolean circuit with \( n \) input bits is a directed acyclic graph in which every node (usually called gates in this context) is either an input node of in-degree 0 labeled by one of the \( n \) input bits, an AND gate, an OR or a NOT gate.
- One of these gates is designated as the output gate.
- Such a circuit naturally computes a function, say \( f \), of its \( n \) inputs.
- The size of a circuit is the number of gates it contains and its depth is the maximal length of a path from an input gate to the output gate.
An example
Problem complexity
Where to test?

64 Bit Adder
- 256 gates
- 2048 transistors
- 2 transistors mis-designed to cause arithmetic errors in the 61st bit of the adder
What to test?

Example 1 – Tests Performed at the Transistor Level

Example 2 – Tests Performed at the Functional Level
“Transistors are free!”
Testing of IC’s

- Testing a chip with hidden functionality shouldn’t be too difficult, right?
  - Wrong!

- Although manufacturers test their very large chips very, very carefully, they can never test every single transistor or all the transistors in concert.
  - Instead they test only for the intended functions.

- All other transistors not used for the intended functions are not tested.
  - “... it is impossible to test all the non-specified or not intended functions ...“ by Professor Ruby Lee from Princeton.
Testing of IC's

- Moreover, manufacturers cannot test very single chip.

- Instead:
  - Assuming an uniformly distributed production process, a small number of random samples is drawn and inspected:
    - thinned layer by layer
    - imaged by a scanning electronic microscope,
    - ...
    - until all layers are inspected to a certain degree.

- But to find in this process very tiny alterations incl. only „redoped regions“ changing simply a bit the timing behavior is absolutely impossible.

- But for mission critical chips you have to test every single chip with destroying the chip itself.
Summary of problem complexity

Figure 2. Proposed hardware Trojan taxonomy based on five different attributes.
New solution idea
Areas of Interest

• **CASE1**: Given an IC corresponding to a known design, does the IC that is delivered do what it is supposed to do and nothing more? This is the case when the Fabrication facility is not trusted but the design process is. The problem is to determine whether the IC hardware received has been modified in order to determine that the fabrication can be trusted.

• **CASE2**: Given a specification and an IC design is the design true to the specification? In this case one assessing the trust of the design software and synthesis tools. The design itself must be validated.

• **CASE3**: Given a re-configurable IC, does the configurable data (bit stream) in the device accurately represent what was intended by the specification, design and VHDL synthesis?
A new attempt to tackle the problem

- **Idea:**
  - Can we design ICs in such a way that no matter what the adversary \((\text{in the untrusted fab})\) does with our IC design, we can at least ensure equivalent functionality?
  - Tiny alterations don’t change the IC functioning.
  - Massive alterations are detected are otherwise.
What are the requirements?

64 Bit Adder
- 256 gates
- 2048 transistors
- 2 transistors mis-designed to cause arithmetic errors in the 61st bit of the adder
DARPA Metrics Challenge

Example 1 – Tests Performed at the Transistor Level

Example 2 – Tests Performed at the Functional Level

Deutsche Telekom Laboratories
The good old world of a random adversary

- von Neumann (1956): The gates of a Boolean circuit fail independently with a probability bounded by a constant:

There is a transformation that takes a circuit $C$ into another circuit $C'$ such that:

1. The transformation takes polynomial time in the size of $C$.
2. The size of $C'$ is $O(S(C) \log (S))$ where $S(C)$ denotes the size of circuit $C$.
3. The depth of $C$ is $O(D(C))$ where $D(C)$ is the depth of circuit $C$.
4. There exist $\varepsilon > 0$ and $p < 1/2$ such that if the gates of $C'$ fail independently with probability bounded by $\varepsilon$ then for every input $x$ the probability that $C(x) \neq C'(x)$ is at most $p$. 
But when the adversary is not random?

- If the adversary biases a few critical gates or the output bit then the circuit inadvertently outputs an incorrect value.

- So?

- We could inspect at least the last few transistors by hand, i.e., the output gate.
New model of adversary

- We would like to build *resilient circuits* that give the correct output even if at each level a small but maliciously chosen constant fraction of the gates are changed/malfunctioning.

- This can be thought of as using a constant number of absolutely reliable gates for the last few levels of the circuit.
Loose computation

Definition:

For any computational device $M$ we say that $M$ $\delta$-loosely computes $f$ if

1. Whenever $f(x) = 1$ then $M(x) = 1$
2. If $f(z) = 0$ for every $z$ with $d(x, z) \leq \delta*n$ then $M(x) = 0$.

Here $d(x, z)$ denotes the Hamming distance between $x$ and $z$, and $n$ the number of input bits to $f$.

$M$ can output an arbitrary value or no value at all if input $x$ does not belong to the above two categories.
Another definition

Definition:

- For an error correcting code $E_n$ with codewords of length $q_n$ and for a function $f$ we define

$$f \cdot E_n : \{0,1\}^{q_n} \rightarrow \{0,1\}$$

as follows

1. $(f \cdot E_n)(z) = 0$ for all $z$ where $z$ is not a codeword of the code $E_n$

2. If $z = E_n(x)$ then $(f \cdot E_n)(z) = f(x)$
Proposition:

- If the Hamming distance of any two codewords in $E_n$ is at least $q_n$ and $M$ is a computational device that computes $f \circ E_n$ in a $\delta$-loose manner then

$$M(E_n(x)) = f(x)$$

on any input $x$. 
It is known from coding theory that there exist linear binary codes $E_n$ with the following properties:

- The matrix of $E_n$ can be polynomially computed in $n$.
- This also means that the length $q_n$ of the codewords is also polynomial in $n$.
- The Hamming distance of any two codewords in $E_n$ is at least $\delta q_n$ for some small constant $\delta > 0$. 
Theorem: Let $C$ be a Boolean circuit and let $f: \{0,1\}^n \rightarrow \{0,1\}$ be the Boolean function computed by $C$. For every $\delta > 0$, there exists a code $E = E_C$ and a circuit $C'$ such that $C'$ computes $f \circ E$ in a $\delta$-loose manner – even if an adversary modifies a constant $\gamma(\delta) > 0$ fraction of the gates on every level including the input level. Moreover $E$ and $C'$ have the following properties:

1. $|E(x)| \leq q(|x|)$ for some polynomial $q$ independent of $C$.
2. The Hamming distance between any two codewords of $E$ is at least $\delta_0 * |E|$ for some $0 < \delta_0 < 1$ independent of $C$ ($\delta_0$ is a function of $\delta$).
3. $D(C') = O(\log S(C))$, i.e., $S(C')$ is also polynomial in $S(C)$.
4. $C'$ can be computed from $C$ in polynomial time and $E(x)$ can be computed from $C$ and $x$ in polynomial time.
Thus

- We can design ICs in such a way that no matter what the adversary (in the untrusted fab) does with our IC design, we can at least ensure equivalent functionality,
- provided that the adversary only changes a constant fraction of gates per circuit layer.
A „sw only exploitation“ of the problem or a much easier attack vector
Bug Attacks

Eli Biham¹, Yaniv Carmeli¹, and Adi Shamir²

¹ Computer Science Department,
Technion - Israel Institute of Technology,
Haifa 32000, Israel
{biham, yanivca}@cs.technion.ac.il
http://www.cs.technion.ac.il/~{biham, yanivca}

² Computer Science Department,
The Weizmann Institute of Science,
Rehovot 76100, Israel
adi.shamir@weizmann.ac.il

Abstract. In this paper we present a new kind of cryptanalytic attack which utilizes bugs in the hardware implementation of computer instructions. The best known example of such a bug is the Intel division bug, which resulted in slightly inaccurate results for extremely rare inputs. Whereas in most applications such bugs can be viewed as a minor nuisance, we show that in the case of RSA (even when protected by OAEP), Pohlig-Hellman, elliptic curve cryptography, and several other schemes, such bugs can be a security disaster: Decrypting ciphertexts on any computer which multiplies even one pair of numbers incorrectly can lead to full leakage of the secret key, sometimes with a single well-chosen ciphertext.
The “Bellcore attack” or Fault-based cryptanalysis of the RSA-CRT signature scheme

Given:

Two RSA-signatures $S$ and $S'$ of the same unknown message $m$, where $S$ is a correct signature and $S'$ is a faulty signature.

Moreover, let $S'$ be such that:

\[ S_p' := m^d \mod p \text{ is wrong, i.e. } S_p \neq S_p' \mod p, \text{ but } \]
\[ S_q' := m^d \mod q \text{ is correct, i.e. } S_q = S_q' \mod q. \]

Then it holds:

\[ S \neq S' \mod p, \text{ but } \]
\[ S = S' \mod q, \text{ i.e. } S - S' = 0 \mod q \iff S - S' = k \cdot q \]

and therefore

\[ \gcd(S - S', N) = \gcd(k \cdot q, p \cdot q) = q, \]

which gives the attacker $p = N / q$ and thus $d$. 
Exploiting „non-malicious“ bugs, ...
Correctness of ARM cores?

The Cortex-A15 Verification Story

Bill Greene
Micah McDaniel
December 7, 2011

Configurability

<table>
<thead>
<tr>
<th>System feature</th>
<th>Cortex-A15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of CPUs</td>
<td>1-4</td>
</tr>
<tr>
<td>Interrupt controller</td>
<td>Optional</td>
</tr>
<tr>
<td>Number of SIs</td>
<td>0-224 in steps of 32</td>
</tr>
<tr>
<td>Power management</td>
<td>Optional climp/power-gate control pins</td>
</tr>
<tr>
<td>Floating point / NEON</td>
<td>None, VFP Only, VFP and NEON</td>
</tr>
<tr>
<td>Error protection</td>
<td>None, L2 cache only, L1 and L2 cache</td>
</tr>
<tr>
<td>L2 cache size</td>
<td>512KB, 1MB, 2MB, 4MB</td>
</tr>
<tr>
<td>L2 tag and data slices</td>
<td>00, 01, 02, 11, 12</td>
</tr>
</tbody>
</table>

- 4*8*2*3*3*4*5*2 = 25920 total configurations
- Exhaustive crossing of slices, ECC/no-ECC, number of CPUs at unit, top, and system level
- Focused directed testing of less intrusive configuration choices, then pairwise crossing in random testing
Cortex A9 errata

2.2.5  (764269) Under very rare circumstances, a sequence of at least three writes merging in the same 64-bit address range might cause data corruption

Status
Affects: Product Cortex-A9, Cortex-A9 MPCore.
Fault Type: Programmer Category A (Rare)
Fault Status: Present in: All r2 revisions. Fixed in r3p0

Description
Under very rare timing circumstances, specific to the Cortex-A9 micro-architecture, a sequence involving at least three writes merging into the same 64-bit cacheable memory region, might cause data corruption.
Cortex A9 errata

The erratum requires the following:

- A cacheable write at address A, in a cache line which is currently being brought in the processor because of a previous request.

- A write at another address B, in a different cache line from A. This write must also miss its cache lookup inside the data cache.

- A second write in the same naturally-aligned 64-bit region as address A. This second write must merge in the same 64-bit slot as address A in the clock cycle when this slot is either merging, or checking its merging status, in the Bus Interface Unit.

- The write at address B which had missed should take priority over the second write at address A which is being replayed in the Bus Interface Unit.

- A cache line, different from the one containing address A, must be allocated in the following cycle.

- A third write in the same 64-bit region as address A. This third write must merge in the same 64-bit slot as the first write at address A. This merge must occur in exactly the same cycle as when cache line A is allocated in the data cache.

- When the write A is replayed on the data cache, it must miss.
Cortex A9 errata

As an example, the following loop is theoretically susceptible to the erratum:

```
loop  LDRB  A
     LDRB  B
     STRB  A
     STRB  B
     SUBS
     BNE   loop
```

Implications

When the erratum occurs, the allocation of the written 64-bit data slot at address A is wrongly cancelled, causing data corruption.

Workaround

There is no practical workaround for this erratum.
Status
Recognizing this enormous vulnerability, the DOD recently launched its most ambitious program yet to verify the integrity of the electronics that will underpin future additions to its arsenal. In December, the Defense Advanced Research Projects Agency (DARPA), the Pentagon’s R&D wing, released details about a three-year initiative it calls the Trust in Integrated Circuits program. The findings
Thank you for your attention!