The ZeroTrust Initiative
There is no Security without Transparency

Paweł Jakub Dawidek
<pjd@ZeroTrust.org> <p.dawidek@wheelsystems.com> <pjd@FreeBSD.org>

CTO, Wheel Systems
Founder of the ZeroTrust Initiative
The ZeroTrust Initiative aims to improve overall IT security by removing forced trust
Problems...

- we are forced to trust the vendors
Problems...

- we are forced to trust the vendors
- no source code for proprietary products
Problems...

- we are forced to trust the vendors
- no source code for proprietary products
- no reproducible builds for open-source
Problems...

- we are forced to trust the vendors
- no source code for proprietary products
- no reproducible builds for open-source
- trusted build environment?
- secure distribution?
- reproducible installs?
- Spectre / Meltdown
Who is right?

Theodore Ts'o

05.09.2013

I am so glad I resisted pressure from Intel engineers to let /dev/random rely only on the RDRAND instruction. To quote from the article below:

"By this year, the Sigint Enabling Project had found ways inside some of the encryption chips that scramble information for businesses and governments, either by working with chipmakers to insert back doors...."

Relying solely on the hardware random number generator which is using an implementation sealed inside a chip which is impossible to audit is a BAD idea.

David Johnston

06.09.2013

I'm pissed that people keep telling people that there's an NSA back door in my RNG. There isn't.
„If it cannot be verified, it is not secure”
Why is that important, exactly?
“Be suspicious of commercial encryption software, especially from large vendors. My guess is that most encryption products from large US companies have NSA-friendly back doors, and many foreign ones probably do as well. It's prudent to assume that foreign products also have foreign-installed backdoors. Closed-source software is easier for the NSA to backdoor than open-source software.”

Bruce Schneier
“Thanks to the recent NSA leaks, people are more worried than ever that their software might have backdoors. If you don't believe that the software vendor can resist a backdoor request, the onus is on you to look for a backdoor. What you want is software transparency.”

prof. Edward W. Felten
No reproducible builds: different...

- compilers
No reproducible builds: different...

- compilers
- compilation options
No reproducible builds: different...

- compilers
- compilation options
- headers
No reproducible builds: different...

- compilers
- compilation options
- headers
- libraries
No reproducible builds: different...

- compilers
- compilation options
- headers
- libraries
- time
No reproducible builds: different...

- compilers
- compilation options
- headers
- libraries
- time
- build environments metadata
No reproducible builds: different...

- compilers
- compilation options
- headers
- libraries
- time
- build environments metadata
- file system metadata in archives
No reproducible builds: different...

- compilers
- compilation options
- headers
- libraries
- time
- build environments metadata
- file system metadata in archives
- signatures
No reproducible builds: different...

- compilers
- compilation options
- headers
- libraries
- time
- build environments metadata
- file system metadata in archives
- signatures
- profile-guided optimizations
No reproducible builds

- How small can a backdoor be?
No reproducible builds

◆ How small can a backdoor be?

OpenSSH 3.0.2 (CVE-2002-0083) - privilege escalation to root

- if (id < 0 || id > channels_alloc) {
+ if (id < 0 || id >= channels_alloc) {
No reproducible builds

- How small can a backdoor be?

Assembly

```assembly
    cmpl $0x0,0x8(%ebp)       cmpl $0x0,0x8(%ebp)
    js   16                   js   16
    mov 0x4,%eax              mov 0x4,%eax
    cmp %eax,0x8(%ebp)        cmp %eax,0x8(%ebp)
    jle  30                  jl  30
    mov 0x8(%ebp),%eax        mov 0x8(%ebp),%eax
    mov %eax,0x4(%esp)        mov %eax,0x4(%esp)
    movl $0x4c,(%esp)         movl $0x4c,(%esp)
    call 25                  call 25
```
No reproducible builds

- How small can a backdoor be?

```
Binary

39 45 08 7e 1a 8b 45
39 45 08 7c 1a 8b 45
```
No reproducible builds

- How small can a backdoor be?

Binary

```
39 45 08 7e 1a 8b 45
01111110
```

```
39 45 08 7c 1a 8b 45
01111100
```
No reproducible builds

- How small can a backdoor be?

Binary

A single bit!
No reproducible builds

- Huge effort to verify TrueCrypt
- On-going work on reproducible builds (TOR, Debian, FreeBSD)
- More awareness among developers needed
- Reflections on Trusting Trust, 1984 Ken Thompson
- Countering Trusting Trust through Diverse Double-Compiling, David A. Wheeler
End-to-end independent verification

- How can you feel secure without it?
Cryptography

- publicly available algorithms
Cryptography

- publicly available algorithms
- extensive peer review
Cryptography

- publicly available algorithms
- extensive peer review
- publicly available cryptoanalysis results
Cryptography: the result?

- secret, home-grown crypto uncommon
Cryptography: the result?

- secret, home-grown crypto uncommon
- the strongest link in the chain
Why not to trust?

- agencies can ask or force organizations to put backdoors
Why not to trust?

- agencies can ask or force organizations to put backdoors
- people can be criminals
Why not to trust?

♦ agencies can ask or force organizations to put backdoors
♦ people can be criminals
♦ people can be bribed
Why not to trust?

- agencies can ask or force organizations to put backdoors
- people can be criminals
- people can be bribed
- people can be intimidated
Why not to trust?

- agencies can ask or force organizations to put backdoors
- people can be criminals
- people can be bribed
- people can be intimidated
- people can be incompetent
Why not to trust?

- agencies can ask or force organizations to put backdoors
- people can be criminals
- people can be bribed
- people can be intimidated
- people can be incompetent
- people’s computers can be hacked
The Solution

- don’t destroy business
The Solution

- don’t destroy business
- propose a license for auditing/reporting purpose
The Solution

- don’t destroy business
- propose a license for auditing/reporting purpose
- encourage and promote reproducible builds
The Solution

- don’t destroy business
- propose a license for auditing/reporting purpose
- encourage and promote reproducible builds
- talk to toolchain vendors
The Solution

- don’t destroy business
- propose a license for auditing/reporting purpose
- encourage and promote reproducible builds
- talk to toolchain vendors
- talk to platform vendors to make verification possible
The Solution

- don’t destroy business
- propose a license for auditing/reporting purpose
- encourage and promote reproducible builds
- talk to toolchain vendors
- talk to platform vendors to make verification possible
- propose ways to protect IP
The Ultimate Goal

ZeroTrust as a natural element of security hygiene
Though questions / Common concerns
We make money by selling our software and we don’t want to destroy our business by giving it away for free.
**V**: We make money by selling our software and don’t want to destroy our business by giving it away for free.

**ZT**: The ZTI doesn’t expect your company to start giving products for free. ZTI will propose a license that will allow to release the source code, but only for auditing and reporting purposes.
Common concerns

V: We don’t want our competitors to use our code which we will release as Open Source.
Common concerns

**V**: We don’t want our competitors to use our code which we will release as Open Source.

**ZT**: With ZTI license that would be illegal. Your competitor will also have disadvantage, because of not releasing the code.
Our current code is a mess. We also have binary blobs from other vendors and no chance to get the source code for that.
**Common concerns**

**V:** Our current code is a mess. We also have binary blobs from other vendors and no chance to get the source code for that.

**ZT:** Then don’t release it. We fully understand it might be too expensive and too risky to release current source code. But when you start building a new product, do it according to the ZTI ideology.
Common concerns

V: It won’t work, nobody will be interested, we are too big to try.
Common concerns

**V:** It won’t work, nobody will be interested, we are too big to try.

**ZT:** Start in small steps. Release ZeroTrust version of your product, with limited functionality and see what the market will choose.
V: How about, to slow down the competitors, we will release the source code some time after releasing the binaries?
Common concerns

**V:** How about, to slow down the competitors, we will release the source code some time after releasing the binaries?

**ZT:** Bad idea. This means people who care, will need to wait for your product to become possible to verify.
Common concerns

V: Opening the source code solves nothing! No one will ever be able to audit my entire code anyway!
Common concerns

**V:** Opening the source code solves nothing! No one will ever be able to audit my entire code anyway!

**ZT:** That’s possible, of course, but that’s not crucial. People may want to audit the code once they suspect something. Independent parties may audit the code and I can choose who to trust. It is much more risky to put a backdoor into a product with open source.
Common concerns

V: Open source software less secure, because it is easier to find security bugs.
Common concerns

V: Open source software less secure, because it is easier to find security bugs.

ZT: Yes, it is easier to find bugs, but...
Common concerns

Time to find a security bug

- **Open Source**
  - Agencies
  - Cybercriminals
  - Whitehat community

- **Closed Source**
  - Agencies
  - Cybercriminals
  - Whitehat community
Common concerns

Time the bug can be exploited by Cybercriminals

- **Open Source**
  - Agencies: 0m, 12m
  - Cybercriminals: 12m, 12m<24m
  - Whitehat community: 0m, 12m, 24m

- **Closed Source**
  - Agencies: 0m, 12m
  - Cybercriminals: 12m, 24m
  - Whitehat community: 0m, 12m, 24m
Common concerns

Time the bug can be exploited by Government Agencies

- **Open Source**
  - Agencies: 12m
  - Cybercriminals: 18m
  - Whitehat community: 18m

- **Closed Source**
  - Agencies: 36m
  - Cybercriminals: 24m
  - Whitehat community: 18m<36m

Legend:
- Agencies
- Cybercriminals
- Whitehat community
Common concerns

V: For my product to work effectively I cannot disclose the source code. For example spammers will quickly learn how to bypass my anti-spam solution.
Common concerns

**V:** For my product to work effectively I cannot disclose the source code. For example spammers will quickly learn how to bypass my anti-spam solution.

**ZT:** Sure, it is your call. Release as much source code as you can and let your customers decide if this explanation convinces them or maybe they will prefer ZT alternative. You may also design your software so that binary-only functionality is closed in a tight sandbox (look out for side-channel attacks).
Common concerns

**V**: How can the ZTI ideology be applied to cloud service providers?
Common concerns

V: How can the ZTI ideology be applied to cloud service providers?

ZT: We don’t know yet, but tarsnap, sync.com.
**Common concerns**

**V:** I’m a vendor from the USA and after Edward Snowden leaks nobody trusts me anymore. What do I do?
Common concerns

V: I’m a vendor from the USA and after Edward Snowden leaks nobody trusts me anymore. What do I do?

ZT: Boy, do we have great news for you! Join the ZTI and rebuild your trust!
To sum up...

- don’t blindly trust the vendors
To sum up...

- don’t blindly trust the vendors
- having source code is always better, but be sure the source code matches the binaries
To sum up...

- don’t blindly trust the vendors
- having source code is always better, but be sure the source code matches the binaries
- start looking for ZeroTrust products
To sum up...

- don’t blindly trust the vendors
- having source code is always better, but be sure the source code matches the binaries
- start looking for ZeroTrust products
- support vendors that apply ZTI even if they provide alternative versions of their products - show them that you care
To sum up...

- don’t blindly trust the vendors
- having source code is always better, but be sure the source code matches the binaries
- start looking for ZeroTrust products
- support vendors that apply ZTI even if they provide alternative versions of their products - show them that you care
- imagine your whole IT infrastructure build on top of ZeroTrust products and it will be so!
Questions?