Fraunhofer AISEC

Fraunhofer Institute for Applied and Integrated Security AISEC

FOSDEM 2025 Lightning Talk

Finding Anomalies in the Debian Packaging System to Detect Supply Chain Attacks

Tobias Specht

02.02.2025

heise online

Report: Malware and supply chain attacks threaten companies

The report "The State of Software Supply Chain" summarizes trends and risks in the software supply chain. Vulnerabilities remain unaddressed for years.

SOCRadar®
Your Eyes Beyond

LottieFiles Supply Chain Attack

Compromised npm
s
Wallets

sonatype

Lottie Player com
chain attack – all

October 31, 2024 By Ax Sharma

THN The Hacker News

Malicious Code in XZ Utils for Linux Systems Enables Remote Code Execution

Popular Linux compression tool XZ Utils found with backdoor. Threat actors can remotely execute code on your machine,...

02.04.2024

curity [Research]

Story of the Year: global IT outages and supply chain attacks

KASPERSKY SECURITY BULLETIN 09 DEC 2024

NATO OTAN NATO REVIEW

Securing Britain's and NATO's digital supply chains

Chris Luenen, Haydn Brooks

SECURITY
SECURITYMAGAZINE.COM

Software supply chain experiences almost 1 attack every 2 days

By Security Staff

Whoami

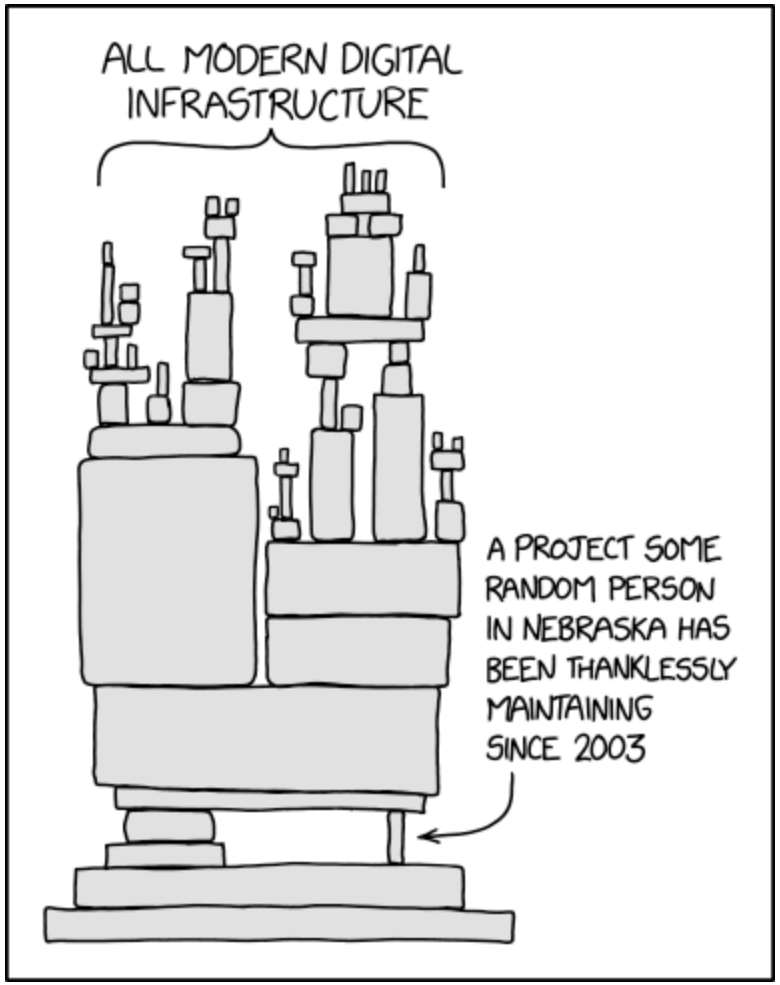
Tobias Specht

- **Cybersecurity researcher based in Germany**
- **Working at the Fraunhofer Institute for Applied and Integrated Security (AISEC)**
- **Research focus on**
- Static Code Analysis
- Offensive Security
- Automotive and Embedded Domain
- **https://github.com/peckto**
- **Matrix: @peckto:tchncs.de**

Agenda

- **Background**
- Supply chain attacks and the XZ backdoor
- Build system (C/C++ and Debian Packages)
- **Supply Graph**
- Concept
- Tracing the build process
- Building and analyzing the graph
- Finding the XZ backdoor
- Limitations
- **Future work**

Background
Supply Chain Attacks

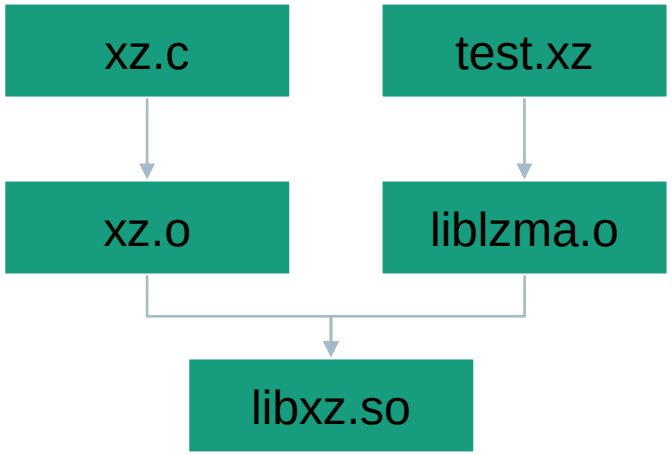


https://xkcd.com/2347

Background

XZ Backdoor CVE-2024-3094 (v5.6.1)

- Target of the attack was the openssh server, to gain remote code execution
- Infiltrating the indirect dependency xz
- Supply chain attack on xz was utilizing the build system to hide and inject its malicious code
- Attack on xz was only detected by coincidence, it’s unknown how many undetected attacks exist (or have existed)



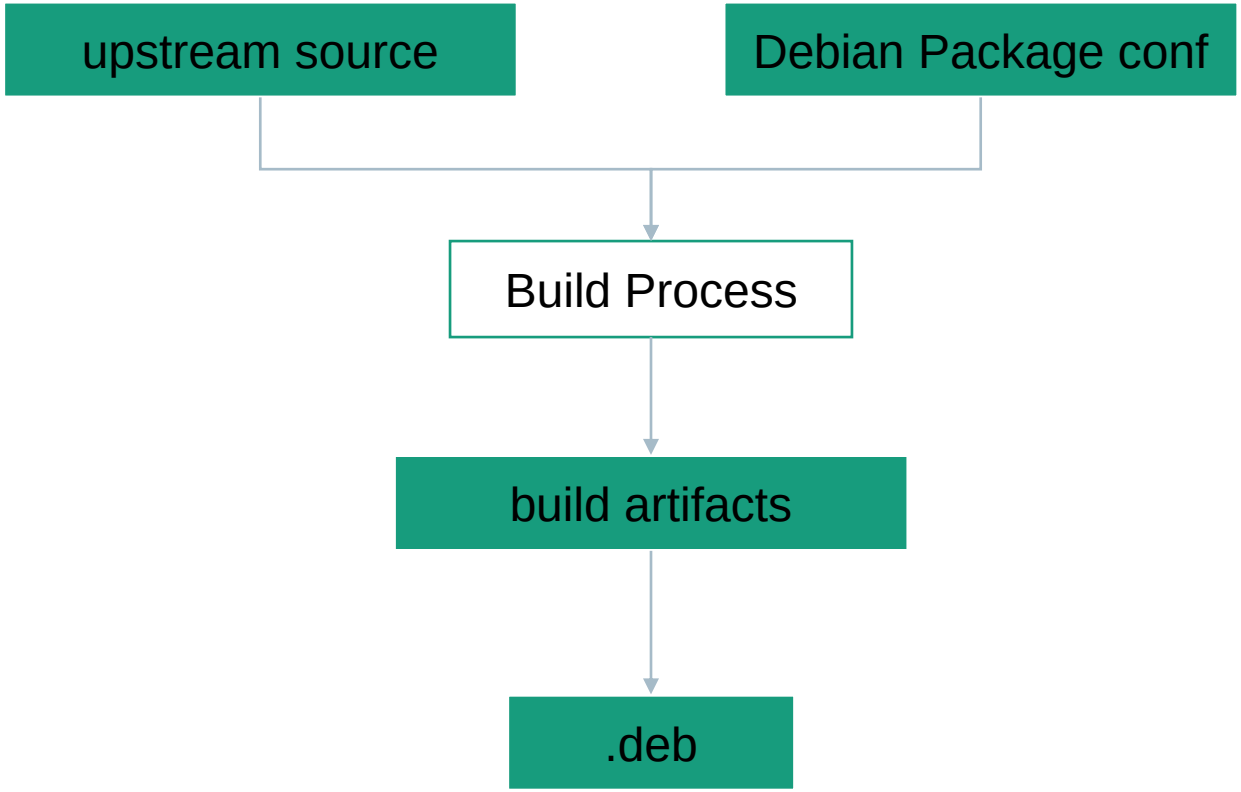
Background
C/C++ Build Process

- **C/C++ build systems are complex and diverse**
- Platform specific build systems (e.g. Windows Visual Studio)
- GNU autotools
- Make
- CMake
- **Build systems are often own, turing-complete programming languages**
- **Often recursive file structure and external resources are used**

→ Complex and intransparent systems are prune to attacks

Background

Debian Package Build Process



Supply Graph
Concept

Idea

- For open source projects, all program parts should be available as source code, no magic binaries linked to target

Solution

- Trace the build process
- Build a graph
- Traverse graph bottom-up to find violations of this rule

Supply Graph

Trace the Build Process

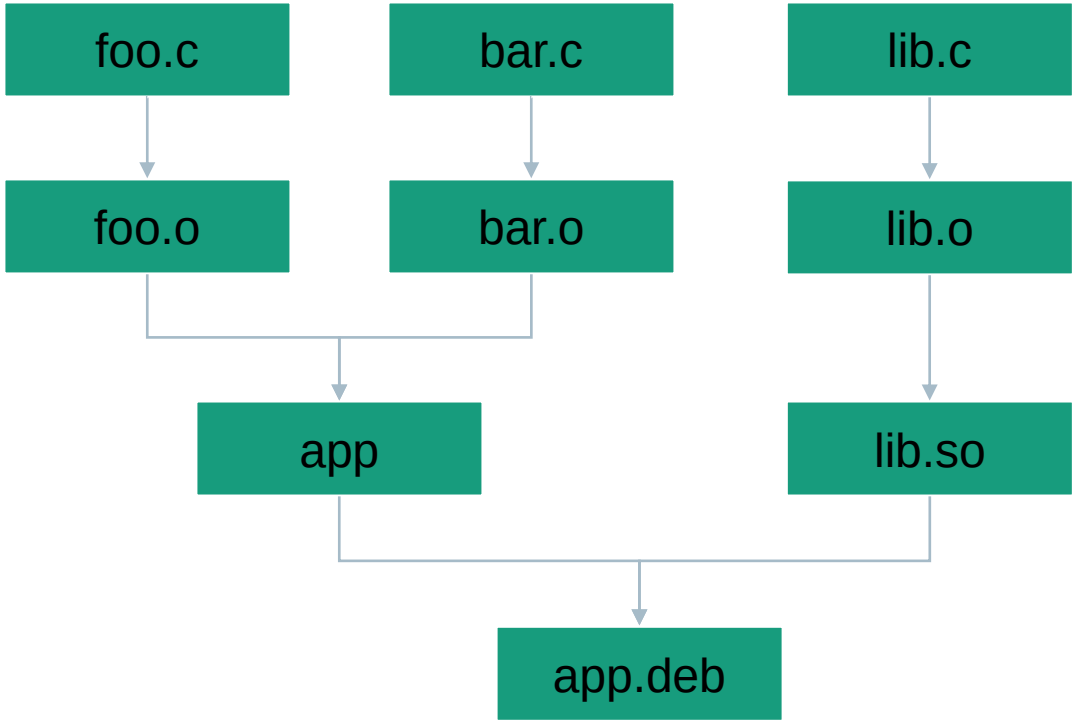
- **Using the tool CodeChecker to capture the compile commands**
- Uses LD_PRELOAD to intercept execv calls
- **Using CC_LOGGER_KEEP_LINK option**
- **Modification to capture additional commands**
- ar, as, cp, install
- **Extension of the compile_commands.json file**
- Adding “output” entry

Example:

```
{
    "arguments": [
        "/usr/bin/clang",
        "-c",
        "-o",
        "foo.o",
        „foo.c"
    ],
    "directory": "/src/",
    "file": "/src/foo.c",
    "output": "/src/foo.o"
}
```

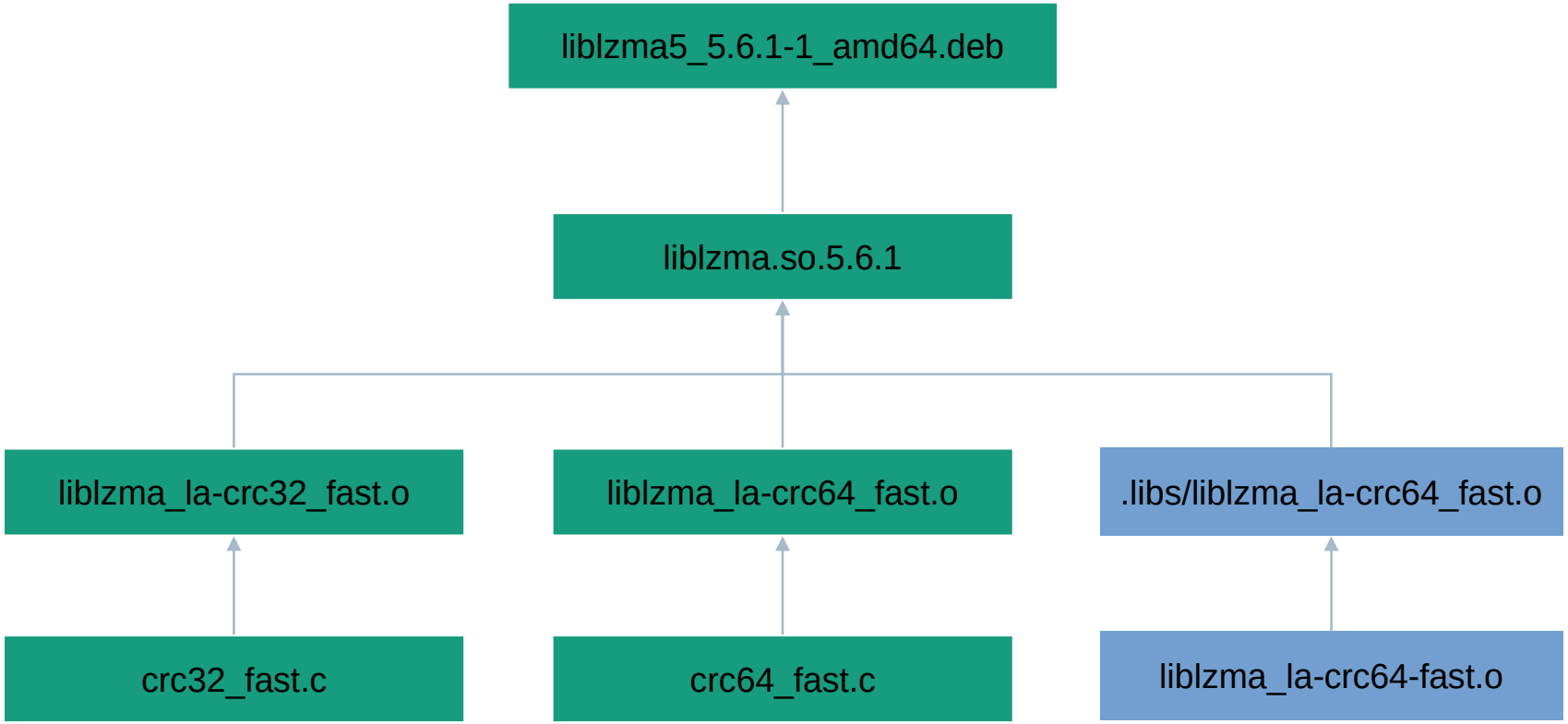
Supply Graph

Graph analysis



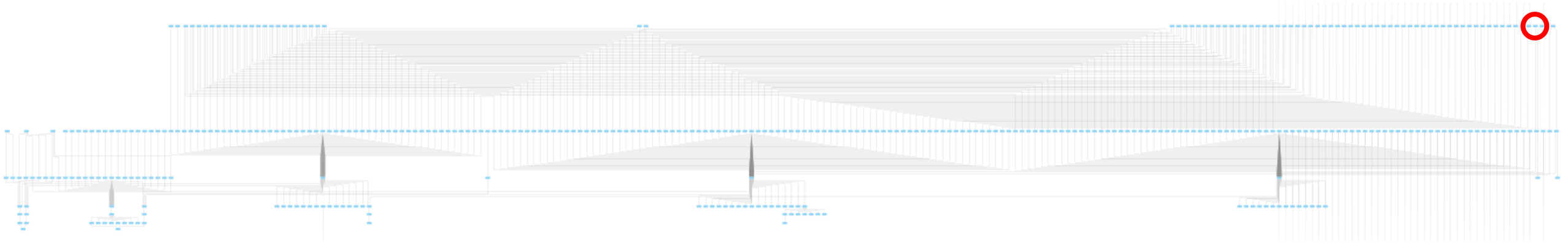
Supply Graph

Find the XZ Backdoor



Supply Graph

Find the XZ Backdoor



Supply Graph

Open Source

README Apache-2.0 license

Analyze build process

In the docker container, the following Debian packet builds are included:

- xz-5.6.1 (CVE-2024-3094)
- xz-5.6.2
- openssh-9.2p1
- openssl-3.0.15

Run the analysis:

```
docker run --rm -it supply-graph
analyze-build-graph xz-5.6.1
```

Identified anomalies in the supply graph are displayed at the end of the log:

```
[...]
Root files not part of upstream:
* /data/xz-5.6.1/xz-utils-5.6.1/debian/normal-build/src/liblzma/liblzma_la-crc64-fast.o
Binary files without corresponding source code:
* /data/xz-5.6.1/xz-utils-5.6.1/debian/normal-build/src/liblzma/liblzma_la-crc64-fast.o
```

https://github.com/Fraunhofer-AISEC/supply-graph

Supply Graph
Limitations

- **Unreliable tracing of build system**
- **Code generation as legitimate „anomaly“**
- E.g. openssl generates ASM code via perl scripts
- Protobuf

Future Work

- Scale it up and scan all Debian Packages
- Implement other analysis on the graph
- Prevent attacks on the build system in general (or at least to make it harder to hide them)

→ Having a descriptive build system, instead of programming

Acknowledgements

This work was funded by the German Federal Ministry of Education and Research (BMBF) as part of the ALPAKA project:

https://www.forschung-it-sicherheit-kommunikationssysteme.de/projekte/alpaka

Contact

Tobias Specht
Group Embedded Software Analysis
Department Product Protection & Industrial Security
Phone +49 89 32299 86 187
tobias.specht@aisec.fraunhofer.de

Fraunhofer Institute for Applied and Integrated Security AISEC
Lichtenbergstr. 11
85748 Garching
Germany

Fraunhofer AISEC

Fraunhofer Institute for Applied and Integrated Security AISEC

https://www.cybersecurity.blog.aisec.fraunhofer.de