Critical Reading Seminar #1

Paris Call: Software Source Code as Heritage for Sustainable Development

Simon Bil

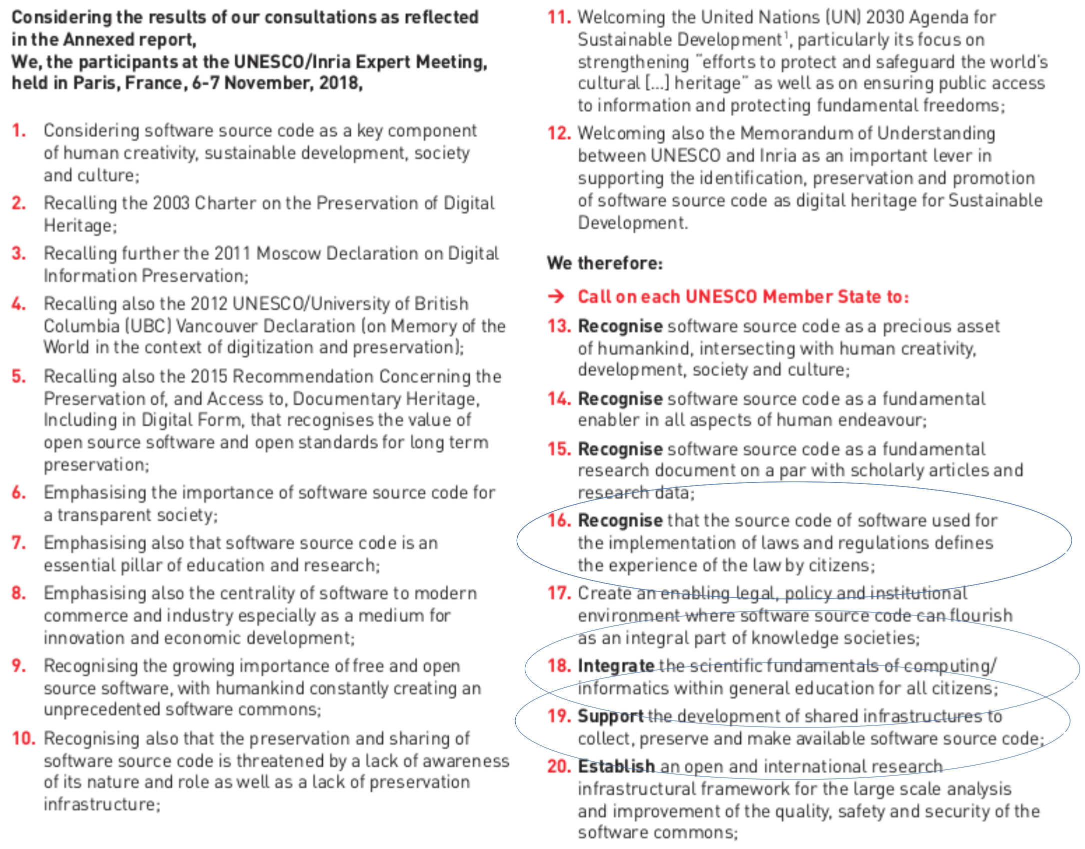
Introduction

- Joint effort by UNESCO and Inria
- preserving technology and scientific knowledge embodied in software source code
- Raises awareness of the current problems
- Argues why preserving software is an important lever for sustainable development

Introduction

- *Software is an essential mediator to access all digital information* ~ human right of access to information
- Our lives are run by software
  - Software runs factories
  - **Software tools power our society**
  - Software is a fundamental pillar of modern research
  - ...

Observations & Measures



Considering the results of our consultations as reflected in the Annexed report,
We, the participants at the UNESCO/Inria Expert Meeting, held in Paris, France, 6-7 November, 2018,

1. Considering software source code as a key component of human creativity, sustainable development, society and culture;
2. Recalling the 2003 Charter on the Preservation of Digital Heritage;
3. Recalling further the 2011 Moscow Declaration on Digital Information Preservation;
4. Recalling also the 2012 UNESCO/University of British Columbia (UBC) Vancouver Declaration (on Memory of the World in the context of digitization and preservation);
5. Recalling also the 2015 Recommendation Concerning the Preservation of, and Access to, Documentary Heritage, Including in Digital Form, that recognises the value of open source software and open standards for long term preservation;
6. Emphasising the importance of software source code for a transparent society;
7. Emphasising also that software source code is an essential pillar of education and research;
8. Emphasising also the centrality of software to modern commerce and industry especially as a medium for innovation and economic development;
9. Recognising the growing importance of free and open source software, with humankind constantly creating an unprecedented software commons;
10. Recognising also that the preservation and sharing of software source code is threatened by a lack of awareness of its nature and role as well as a lack of preservation infrastructure;
11. Welcoming the United Nations (UN) 2030 Agenda for Sustainable Development[1], particularly its focus on strengthening "efforts to protect and safeguard the world's cultural [...] heritage" as well as on ensuring public access to information and protecting fundamental freedoms;
12. Welcoming also the Memorandum of Understanding between UNESCO and Inria as an important lever in supporting the identification, preservation and promotion of software source code as digital heritage for Sustainable Development.

We therefore:

→ Call on each UNESCO Member State to:

13. **Recognise** software source code as a precious asset of humankind, intersecting with human creativity, development, society and culture;
14. **Recognise** software source code as a fundamental enabler in all aspects of human endeavour;
15. **Recognise** software source code as a fundamental research document on a par with scholarly articles and research data;
16. **Recognise** that the source code of software used for the implementation of laws and regulations defines the experience of the law by citizens;
17. Create an enabling legal, policy and institutional environment where software source code can flourish as an integral part of knowledge societies;
18. **Integrate** the scientific fundamentals of computing/informatics within general education for all citizens;
19. **Support** the development of shared infrastructures to collect, preserve and make available software source code;
20. **Establish** an open and international research infrastructural framework for the large scale analysis and improvement of the quality, safety and security of the software commons;

21. **Ensure** necessary exceptions to copyright and limitations on intermediary liability related to software for archival, preservation, accessibility, education and research purposes;

22. **Enable** effective independent auditing of software source code used to make decisions that may affect fundamental rights of human beings and where possible ensure it is made available under an open source license;

23. **Implement**, with support from UNESCO's Memory of the World Programme, the 2015 Recommendation concerning the Preservation of, and Access to, Documentary Heritage, including in Digital Form, inviting inter-alia Member States to facilitate access to proprietary codes, keys and unlocked versions of technology on a nonprofit basis.

→ **Call on UNESCO and Inria to:**

24. **Strengthen** UNESCO's support for the Software Heritage initiative, as a way of enhancing awareness of the importance of preserving and providing access to source code;

25. **Forge** more strategic partnerships in order to create greater recognition of software development activity as science and research, particularly by demonstrating how software source code can be appropriated as a research product worthy of preservation, while at the same time promoting its recognition as a valid field of both applied and academic enquiry, with reproducible or verifiable research results;

26. **Support** efforts for the development of an open Global Software Registry, which will help all stakeholders to recognize and enable software reuse as an important part of all modern software developments by providing a universal catalogue that will index all available software components, with the metadata needed to properly locate and reuse them.

→ **Call on software developers, memory institutions, the business sector, academia and civil society, within their competency, to:**

27. **Recognise** that software is the result of a significant part of the intellectual efforts of humankind over recent decades, and it is an important part of our cultural and industrial heritage;

28. **Support** efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive;

29. **Promote** software development as a valuable research activity, and research software as a key enabler for Open Science/Open Research, sharing good practices and recognising in the careers of academics their contributions to high quality software development, in all their forms;

30. **Recognize** the importance of contributions by people of all genders from all over the world to the software commons, supporting a diverse and inclusive environment for all aspects of software development and curation;

31. **Educating** decision makers on the specificities of software, and software source code in particular, raising awareness about the threats to the software commons and the importance to protect it;

32. **Encourage** all stakeholders to develop a common system of cataloguing to allow for easy identification and retrieval of software source code, even across the many platforms and infrastructures used to develop and/or distribute it;

33. **Support** stakeholders in developing a universal archive, as part of a broad effort at digital preservation, that will ensure persistence of and universal access to software source code;

34. **Encourage** multidisciplinary activity in the field of software preservation, and in particular collaboration with the humanities and social sciences whose contributions are essential to study the history of technology;

35. **Adapt** processes, workflows and licensing schemas in the software industry to ease the transition of future proprietary software source code into the software commons once it is no longer commercially viable;

36. **Foster** international collaboration to build a common framework for software preservation and access, and mutualise resources, in order to avoid the dispersion of efforts;

37. **Promote** the recording of the activity of software developers, captured as documentary heritage either in analogue or digital form, which are suitable for preservation in their own right and ensure that they are linked with the source code.

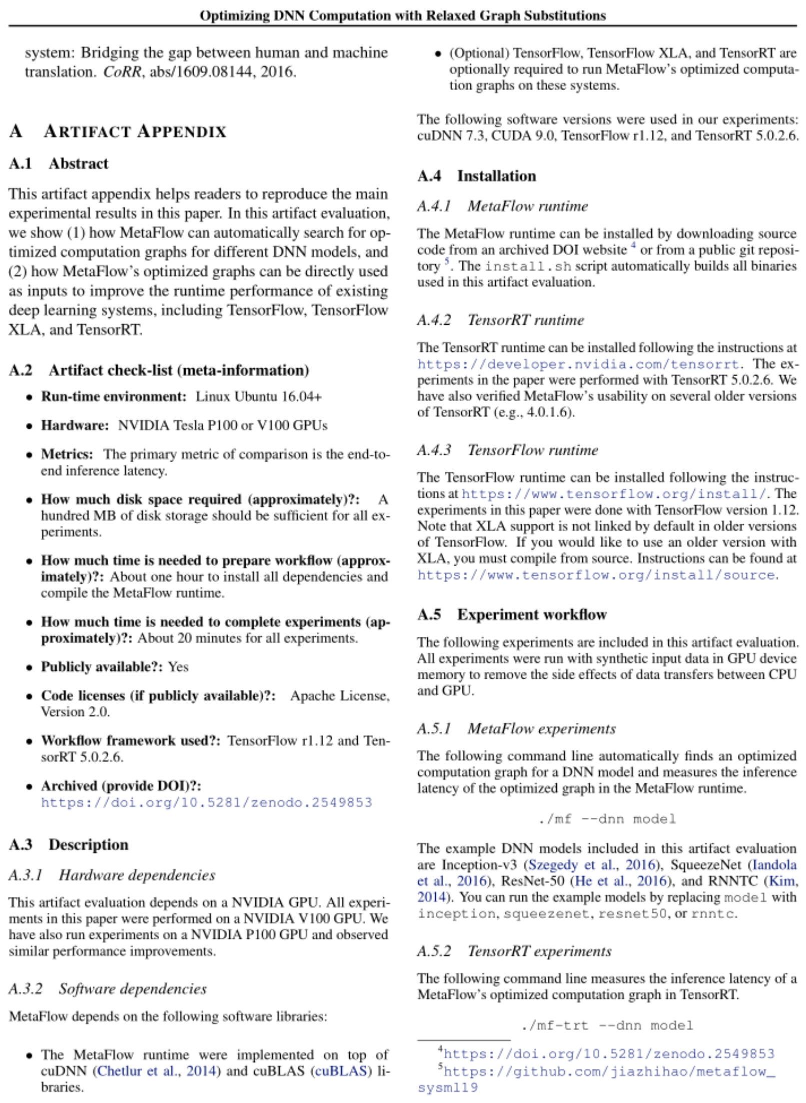
38. **Support** all stakeholders in developing the understanding that software source code is intertwined more and more with the fabric of our society, hence the utmost care needs to be used during its development process to manage its potential consequences on society and people.

Adopted on 7 November 2018, Paris, France

1. Source code as a precious asset of humankind

- Key component of human creativity, development, society and culture
- A needed prerequisite in many fields of activity
- Essential pillar of education and research
  - Reproducibility of research

Reproducibility of research



system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

A Artifact Appendix

A.1 Abstract

This artifact appendix helps readers to reproduce the main experimental results in this paper. In this artifact evaluation, we show (1) how MetaFlow can automatically search for optimized computation graphs for different DNN models, and (2) how MetaFlow's optimized graphs can be directly used as inputs to improve the runtime performance of existing deep learning systems, including TensorFlow, TensorFlow XLA, and TensorRT.

A.2 Artifact check-list (meta-information)

- **Run-time environment:** Linux Ubuntu 16.04+
- **Hardware:** NVIDIA Tesla P100 or V100 GPUs
- **Metrics:** The primary metric of comparison is the end-to-end inference latency.
- **How much disk space required (approximately)?:** A hundred MB of disk storage should be sufficient for all experiments.
- **How much time is needed to prepare workflow (approximately)?:** About one hour to install all dependencies and compile the MetaFlow runtime.
- **How much time is needed to complete experiments (approximately)?:** About 20 minutes for all experiments.
- **Publicly available?:** Yes
- **Code licenses (if publicly available)?:** Apache License, Version 2.0.
- **Workflow framework used?:** TensorFlow r1.12 and TensorRT 5.0.2.6.
- **Archived (provide DOI)?:** https://doi.org/10.5281/zenodo.2549853

A.3 Description

A.3.1 Hardware dependencies

This artifact evaluation depends on a NVIDIA GPU. All experiments in this paper were performed on a NVIDIA V100 GPU. We have also run experiments on a NVIDIA P100 GPU and observed similar performance improvements.

A.3.2 Software dependencies

MetaFlow depends on the following software libraries:

- The MetaFlow runtime were implemented on top of cuDNN (Chetlur et al., 2014) and cuBLAS (cuBLAS) libraries.
- (Optional) TensorFlow, TensorFlow XLA, and TensorRT are optionally required to run MetaFlow's optimized computation graphs on these systems.

The following software versions were used in our experiments: cuDNN 7.3, CUDA 9.0, TensorFlow r1.12, and TensorRT 5.0.2.6.

A.4 Installation

A.4.1 MetaFlow runtime

The MetaFlow runtime can be installed by downloading source code from an archived DOI website [4] or from a public git repository [5]. The `install.sh` script automatically builds all binaries used in this artifact evaluation.

A.4.2 TensorRT runtime

The TensorRT runtime can be installed following the instructions at https://developer.nvidia.com/tensorrt. The experiments in the paper were performed with TensorRT 5.0.2.6. We have also verified MetaFlow's usability on several older versions of TensorRT (e.g., 4.0.1.6).

A.4.3 TensorFlow runtime

The TensorFlow runtime can be installed following the instructions at https://www.tensorflow.org/install/. The experiments in this paper were done with TensorFlow version 1.12. Note that XLA support is not linked by default in older versions of TensorFlow. If you would like to use an older version with XLA, you must compile from source. Instructions can be found at https://www.tensorflow.org/install/source.

A.5 Experiment workflow

The following experiments are included in this artifact evaluation. All experiments were run with synthetic input data in GPU device memory to remove the side effects of data transfers between CPU and GPU.

A.5.1 MetaFlow experiments

The following command line automatically finds an optimized computation graph for a DNN model and measures the inference latency of the optimized graph in the MetaFlow runtime.

```
./mf --dnn model
```

The example DNN models included in this artifact evaluation are Inception-v3 (Szegedy et al., 2016), SqueezeNet (Iandola et al., 2016), ResNet-50 (He et al., 2016), and RNNTC (Kim, 2014). You can run the example models by replacing `model` with `inception`, `squeezenet`, `resnet50`, or `rnntc`.

A.5.2 TensorRT experiments

The following command line measures the inference latency of a MetaFlow's optimized computation graph in TensorRT.

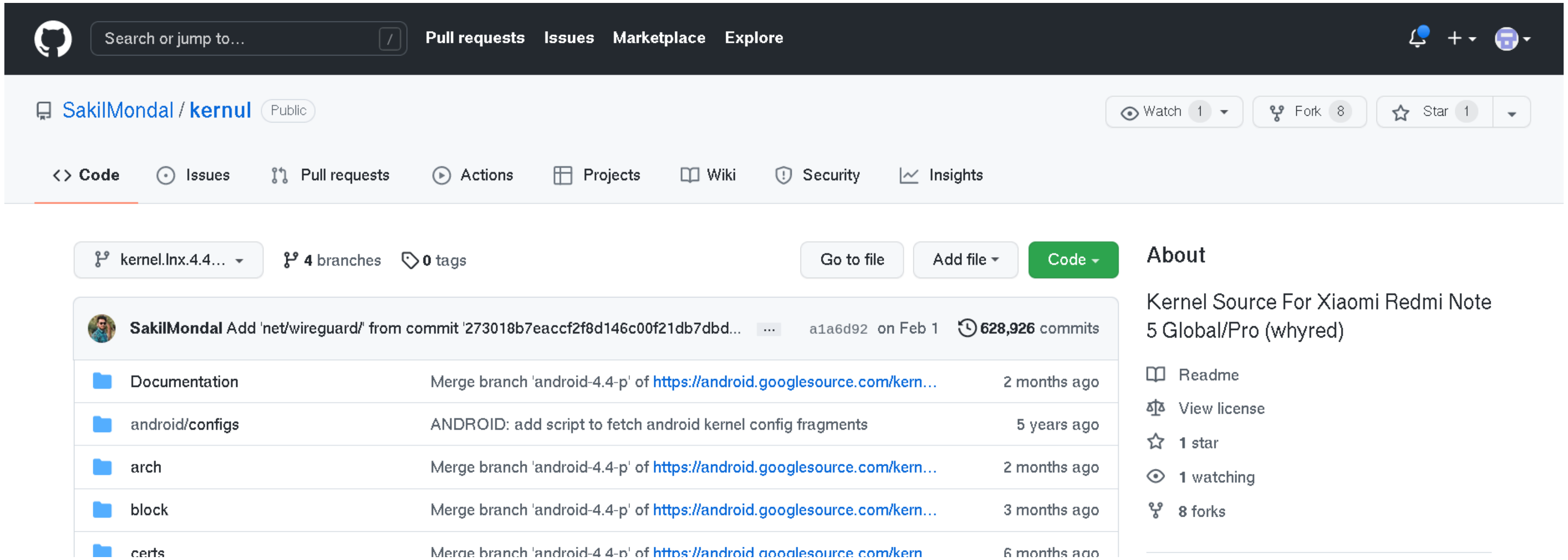
```
./mf-trt --dnn model
```

[4] https://doi.org/10.5281/zenodo.2549853
[5] https://github.com/jiazhihao/metaflow_sysml19

2. Threats to source code preservation and sharing

- Insufficient awareness among decision makers
- Lack or recognition for software creators
- Lack of incentives to release (legacy) source code
  - *There is little incentive to proper release source code of proprietary software when it is no longer commercially viable*
- Lack of an universal catalog & repository
  - There is no clearly defined way of naming, referencing, and citing software projects like with books
  - Makes exploring the Software Commons as a whole a difficult task
  - We lack a *universal, comprehensive, and long-term repository*

Lack of incentives to release – Linux Kernel Sources



SakilMondal / kernul Public

kernel.lnx.4.4… 4 branches 0 tags

SakilMondal Add 'net/wireguard/' from commit '273018b7eaccf2f8d146c00f21db7dbd... a1a6d92 on Feb 1 628,926 commits

Documentation	Merge branch 'android-4.4-p' of https://android.googlesource.com/kern…	2 months ago
android/configs	ANDROID: add script to fetch android kernel config fragments	5 years ago
arch	Merge branch 'android-4.4-p' of https://android.googlesource.com/kern…	2 months ago
block	Merge branch 'android-4.4-p' of https://android.googlesource.com/kern…	3 months ago
certs	Merge branch 'android-4.4-p' of https://android.googlesource.com/kern	6 months ago

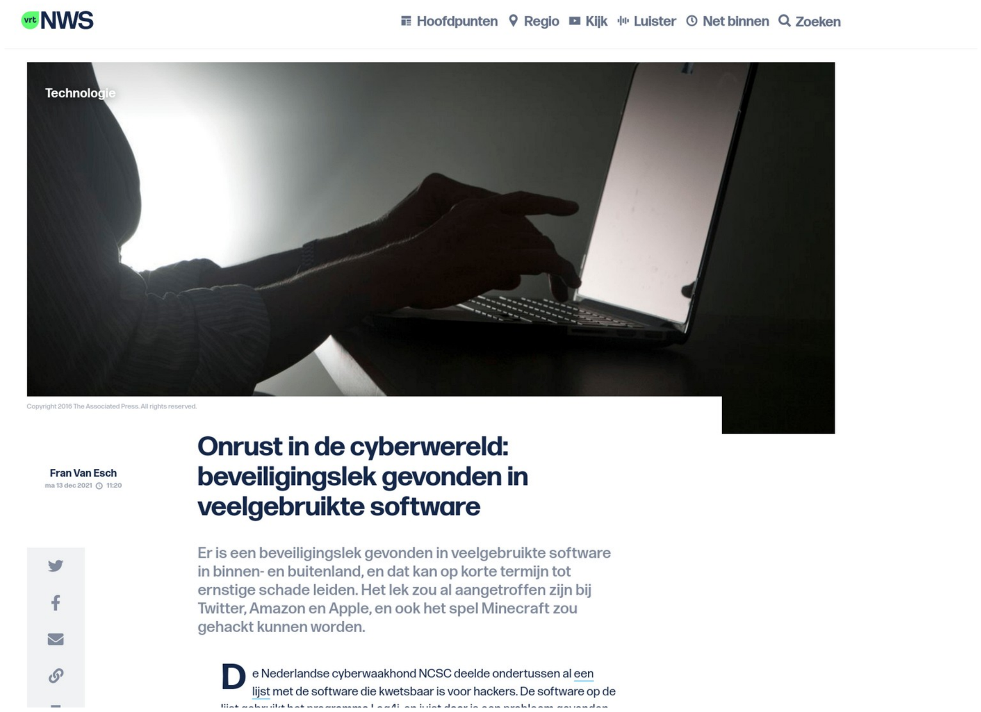
About

Kernel Source For Xiaomi Redmi Note 5 Global/Pro (whyred)

Readme
View license
1 star
1 watching
8 forks

Lack of recognition - log4j

- Left volunteers scrambling to fix it
- It emerged that the project was maintained by a team of three developers, on a part-time basis
- Only 3 GH sponsors
- Parallels can be drawn between this story and what happened with Heartbleed (OpenSSL)



Copyright 2016 The Associated Press. All rights reserved.

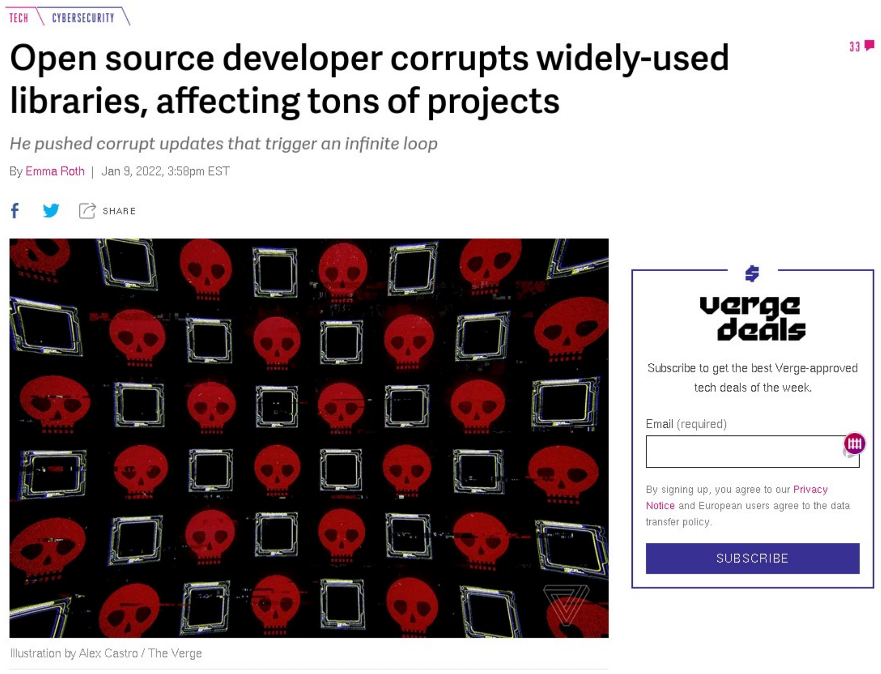
Fran Van Esch
ma 13 dec 2021 11:20

Onrust in de cyberwereld: beveiligingslek gevonden in veelgebruikte software

Er is een beveiligingslek gevonden in veelgebruikte software in binnen- en buitenland, en dat kan op korte termijn tot ernstige schade leiden. Het lek zou al aangetroffen zijn bij Twitter, Amazon en Apple, en ook het spel Minecraft zou gehackt kunnen worden.

De Nederlandse cyberwaakhond NCSC deelde ondertussen al een lijst met de software die kwetsbaar is voor hackers. De software op de

Lack of recognition - Faker.js



TECH \ CYBERSECURITY

Open source developer corrupts widely-used libraries, affecting tons of projects

33

He pushed corrupt updates that trigger an infinite loop

By Emma Roth | Jan 9, 2022, 3:58pm EST

SHARE

verge deals

Subscribe to get the best Verge-approved tech deals of the week.

Email (required)

By signing up, you agree to our Privacy Notice and European users agree to the data transfer policy.

SUBSCRIBE

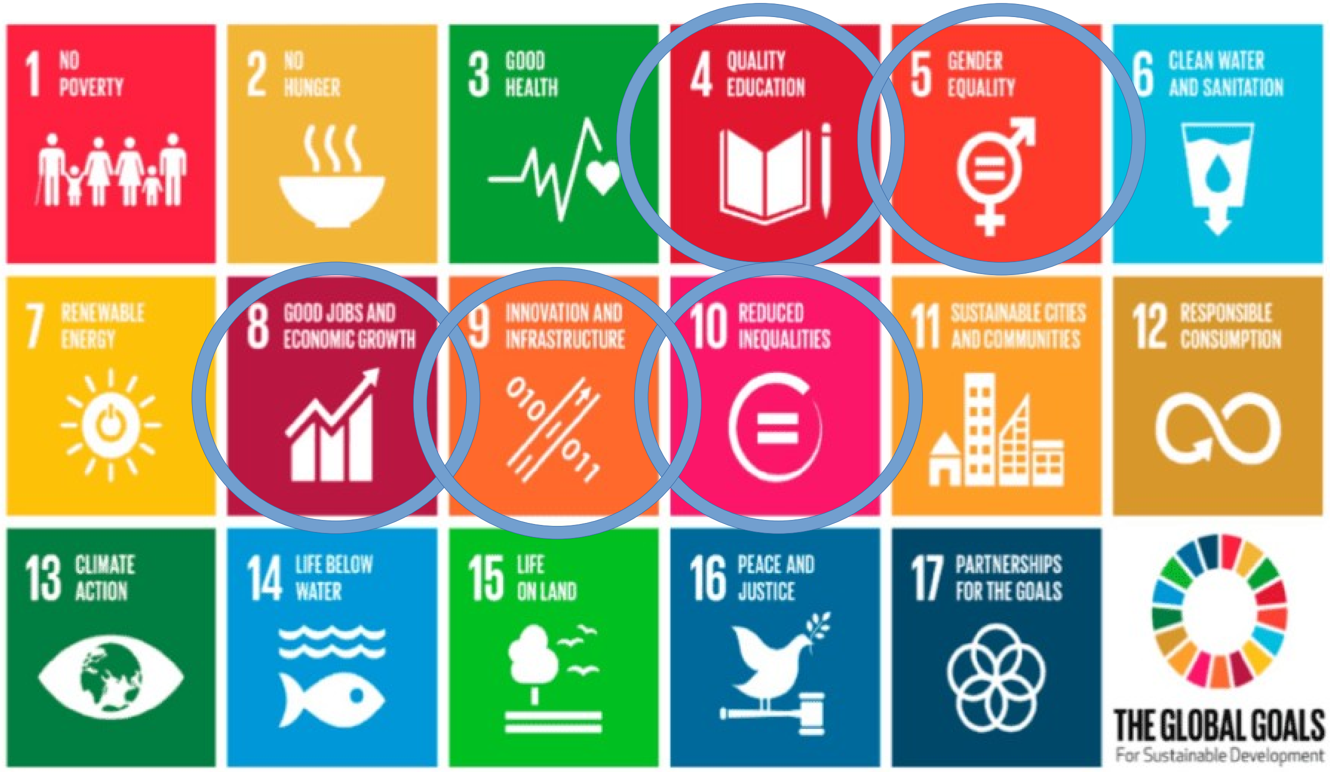
Illustration by Alex Castro / The Verge

- *The sabotaged versions cause applications to infinitely output strange letters and symbols, beginning with three lines of text that read "LIBERTY LIBERTY LIBERTY."*
- *Squires' bold move draws attention to the moral — and financial — dilemma of open-source development, which was likely the goal of his actions.*
- *"It's hard to get your boss to pay for software; it's extra hard when you get no additional value."*

3. Stakeholders

- Governments
  - As software users, commissioners and producers
  - As law makers
- Research and Higher Education
  - Open science/research and reproducibility
  - Software development literacy
  - The transfer of knowledge and practices
- Education
- Industry and Commerce
- **Memory Institutions**
- NGOs / Other **standard-settings bodies**
- **Individuals**
  - As target audience but also as creator

Lever for sustainable development?


1 NO POVERTY
2 NO HUNGER
3 GOOD HEALTH
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 RENEWABLE ENERGY
8 GOOD JOBS AND ECONOMIC GROWTH
9 INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE AND JUSTICE
17 PARTNERSHIPS FOR THE GOALS
THE GLOBAL GOALS
For Sustainable Development

My view?