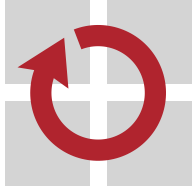
Ausgewählte Kapitel der Systemsoftware (AKSS)

Forschungsethik

02. Juni 2021

Phillip Raffeck, Tim Rheinfels, Simon Schuster, Peter Wägemann

Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg



Lehrstuhl für Verteilte Systeme
und Betriebssysteme

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

BibTeX-Einträge

```
@INPROCEEDINGS{9113112, author={Alcon, Miguel and Tabani, Hamid and Kosmidis,
    ↪Leonidas and Mezzetti, Enrico and Abella, Jaume and Cazorla, Francisco J.},
    ↪ booktitle={2020 IEEE Real-Time and Embedded Technology and Applications
    ↪Symposium (RTAS)}, title={Timing of Autonomous Driving Software: Problem
    ↪Analysis and Prospects for Future Solutions}, year={2020}, volume={},
    ↪number={}, pages={267-280}, doi={10.1109/RTAS48715.2020.000-1}}


@INPROCEEDINGS{alcon:2020:rtas,
  author={Miguel Alcon and Hamid Tabani and Leonidas Kosmidis and Enrico Mezzetti and
      ↪ Jaume Abella and Francisco J. Cazorla},
  booktitle={Proceedings of the 26th IEEE Real-Time and Embedded Technology and
      ↪Applications Symposium (RTAS '20)},
  title={Timing of Autonomous Driving Software: Problem Analysis and Prospects for
      ↪Future Solutions},
  year={2020},
  pages={267--280},
  doi={10.1109/RTAS48715.2020.000-1},
}
```

Agenda

Betrugsfälle

Gute Wissenschaftliche Praxis

Auswirkungen von Forschung

Problematik Dual-Use

Verantwortung in Ingenieursberufen

Betrugsfälle

Warum ist gute wissenschaftliches Praxis eigentlich wichtig?

Fall Friedhelm Hermann

- ehem. renommierter deutscher Krebsforscher
- **Fälschungsskandal** in 1997
  - systematische Fälschung von Labordaten
  - Diebstahl von Ideen und Ergebnissen anderer Forscher
  - ➔ DFG klagt auf Rückzahlung der Forschungsgelder
  - ➔ 2005: teilweise Rückzahlung

Warum ist gute wissenschaftliches Praxis eigentlich wichtig?

Fall Friedhelm Hermann

- ehem. renommierter deutscher Krebsforscher
- **Fälschungsskandal** in 1997
  - systematische Fälschung von Labordaten
  - Diebstahl von Ideen und Ergebnissen anderer Forscher
  - ➔ DFG klagt auf Rückzahlung der Forschungsgelder
  - ➔ 2005: teilweise Rückzahlung

Betrugsskandal in China (2017)

- Chinas Forschungsministerium deckt großflächigen *Peer-Review*-Betrugsring auf
- fast 500 Forscher schuldig gesprochen
- Zurücknahme von über 100 Papieren

Aktuelles Fallbeispiel

- *Peer-Review*-Betrugsring[1]
- „artificial intelligence and machine learning“

Vorgehen

1. Gruppe reicht Papiere ein
2. Papiertitel der Gruppe bekannt
3. Bieten für ebenjene Papiere
4. Erstellen positiver Gutachten
5. Bedrohung unbeteiligter Gutachter
6. Diskussionsteilnahme unter falschem Namen

[1] https://cacm.acm.org/magazines/2021/6/252840-collusion-rings-threaten-the-integrity-of-computer-science-research/fulltext

Vertrauen in Veröffentlichungen

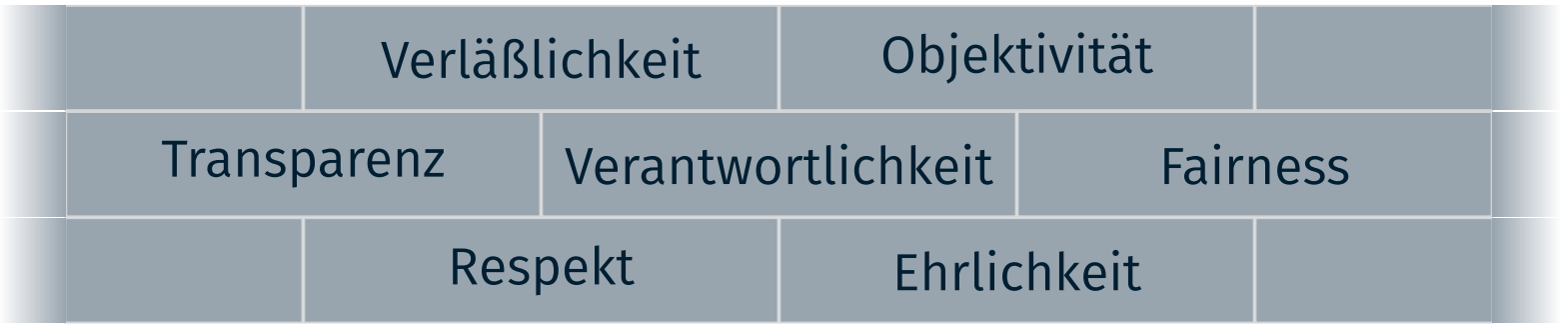
- Kenntnis und Verwendung des aktuellen Stand der Kunst
- kritische Betrachtung von Publikationen
- Grenzen des Begutachtungsprozesses
  - Zeitrahmen
  - Publikationsvolumen
  - Gutachter betreiben nicht die Forschung erneut!
- Grundlage des wissenschaftlichen Prozesses
  - Überprüfung
  - Neubewertung

Gute Wissenschaftliche Praxis

Gute Wissenschaftliche Praxis

Einhaltung allgemeiner Standards und Regeln

- Forschungspraktiken
- Ergebnisinterpretation, selbstkritischer Blick
- Teilnahme am wissenschaftlichen Diskurs
- Veröffentlichung
  - Autorenschaft
  - Zitierung
  - Ergebnisssicherung



Gute wissenschaftliche Praxis in der DFG

Deutsche Forschungsgemeinschaft (DFG) gibt Richtlinien und Leitfäden[2] heraus:

- „Leitlinien zur Sicherung guter wissenschaftlicher Praxis“
  - Kodex für alle Forschungseinrichtungen
  - Aktuelle Fassung vom 01. August 2019
  - Rechtsverbindliche Umsetzung als Voraussetzung für DFG-Fördermittel
  - Frist: 31.07.2022 (coronabedingt verlängert)
- Zusätzlich:
  - Denkschrift „Sicherung guter wissenschaftlicher Praxis“
  - Verfahrensleitfaden zur guten wissenschaftlichen Praxis

[2] https://www.dfg.de/foerderung/grundlagen_rahmenbedingungen/gwp/

Leitlinien zur Sicherung guter wissenschaftlicher Praxis

Leitlinien zur Sicherung guter wissenschaftlicher Praxis

Kodex

DFG

DFG-Prinzipien

- Selbstverpflichtung für Forschungseinrichtungen
  - Festlegung & Einhaltung von Regeln für gute wiss. Arbeit
  - Kommunikation an Angehörige
- Verantwortung liegt bei einzelnen WissenschaftlerInnen
  - Arbeit nach *Lege artis*
  - Ehrlichkeit
  - Hinterfragen eigener Ergebnisse
  - …

Übersicht Regelungen (I)

Akademische Werte

- Verantwortung für grundlegende Werte einzustehen
- Wertevermittlung in akademischer Lehre
- Forschungseinrichtungen schaffen Rahmenbedinungen

Übersicht Regelungen (I)

Akademische Werte

- Verantwortung für grundlegende Werte einzustehen
- Wertevermittlung in akademischer Lehre
- Forschungseinrichtungen schaffen Rahmenbedinungen

Forschungsprozess

- Forschung nach *lege artis*
- Berücksichtigung aktueller Forschungsstand
- Bewusstsein für Forschungsfolgen

Übersicht Regelungen (II)

Publikation und Ergebnissicherung

- Autorenschaft nur mit nachvollziehbarem Beitrag
- sorgfältige Auswahl des Publikationsorgans
- Veröffentlichung von Negativergebissen
- nachvollziehbare Dokumentation
- Archivierung von Publikation, Rohdaten, . . .
- „Quellcode von öffentlich zugänglicher Software muss persistent, zitierbar und dokumentiert sein“

Übersicht Regelungen (III)

Wissenschaftliches Fehlverhalten

- definierte Verfahren
- unabhängige Ombudspersonen
- Hinweisgeberschutz

An der FAU (I)

Satzung zur Sicherung guter wissenschaftlicher Praxis und zum Umgang mit wissenschaftlichem Fehlverhalten an der Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)

Vom 10. Oktober 2017

Aufgrund des Art. 13 Abs. 1 Satz 2 in Verbindung mit Art. 6 Abs. 1 Satz 3 Halbsatz 2 des Bayerischen Hochschulgesetzes (BayHSchG) erlässt die FAU folgende Satzung:

Inhaltsverzeichnis:

An der FAU (II)

Kommissionsmitglieder

Vorsitz

- Prof. Dr. Ferrari, Michele Camillo, +49 9131 85-22416, michele.ferrari@fau.de

Mitglied der Gruppe der Professorinnen und Professoren

- Prof. Dr. Ferrari, Michele Camillo, +49 9131 85-22416, michele.ferrari@fau.de
- Prof. Dr. Fischer, Dagmar, Apoth., +49 9131 85-29552, dagmar.fischer@fau.de
- Prof. Dr. Trollmann, Regina, +49 9131 85-33753, regina.trollmann@uk-erlangen.de

Ombudsmann

- Prof. Dr.-Ing. Fey, Dietmar, +49 9131 85 27003, dietmar.fey@informatik.uni-erlangen.de

Stellvertretender Ombudsmann

- Prof. Dr. Kudlich, Hans, +49 9131 85-22248, Hans.Kudlich@jura.uni-erlangen.de

30.07. Good Research Practice and Scientific Integrity – An Introduction (ONLINE)

Trainer:

Dr. Christian Schmitt-Engel is supporting young academics at FAU's Graduate Centre. He is a trained molecular biologist and did basic research as a doctoral researcher and postdoc at FAU and the University of Göttingen. Furthermore he was involved in teaching during these times and as a fulltime lecturer thereafter.

Relevanz im Seminar

~~Zitat~~ ~~Paraphrase~~ Plagiat

- mangelhafte oder fehlende Quellenangabe
- fälschlicher Eindruck der eigenen Urheberschaft
- Konsequenzen
  - Nichtbestehen
  - Exmatrikulation
  - Aberkennung

Auswirkungen von Forschung

Forschungsfrage

Vielaugenprinzip in Open Source – findet es (vorsätzlich eingebrachte) Sicherheitslücken zuverlässig?

Forschungsfrage

Vielaugenprinzip in Open Source – findet es (vorsätzlich eingebrachte) Sicherheitslücken zuverlässig?

Untersuchung im Linuxkern

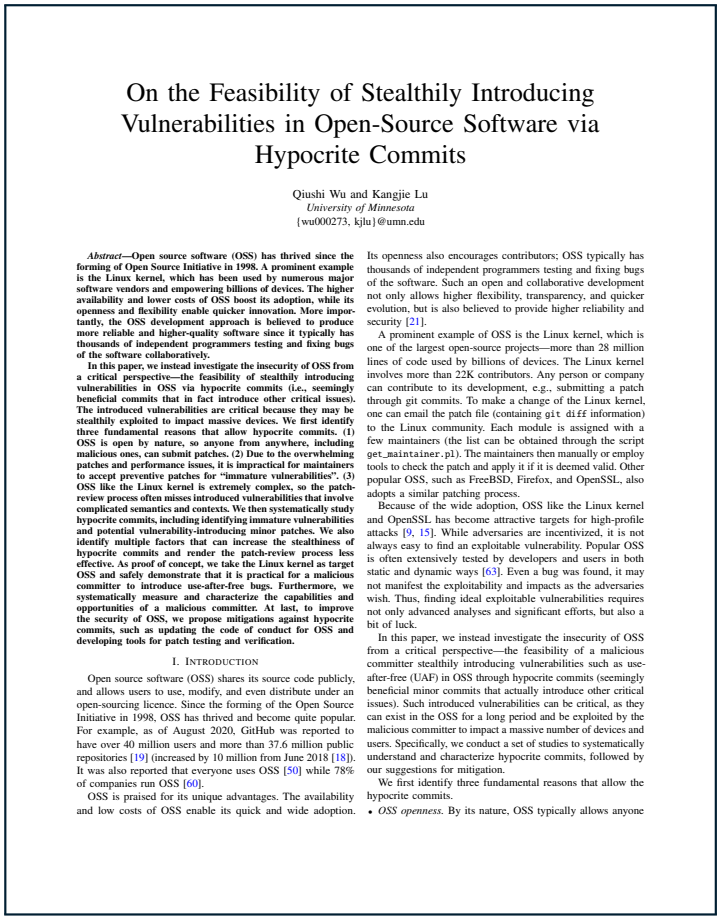
1. Analyse der Commitgeschichte
2. Einreichung eigener Commits[3]

```
Cc: Herbert Xu [...] linux-crypto@vger.kernel.org, linux-kernel@vger.kernel.org
Subject: [PATCH] crypto: cavium/nitrox: add an error message to explain [...]
Date: Thu, 20 Aug 2020 22:12:08 -0500

Provide an error message for users when pci_request_mem_regions failed. [...]

--- a/drivers/crypto/cavium/nitrox/nitrox_main.c
+++ b/drivers/crypto/cavium/nitrox/nitrox_main.c
@@ -451,6 +451,7 @@ static int nitrox_probe(struct pci_dev *pdev,
  if (err) {
    pci_disable_device(pdev);
+   dev_err(&pdev->dev, "Failed to request mem regions!\n");
    return err;
  }
```

[3] https://lore.kernel.org/lkml/20200821031209.21279-1-acostag.ubuntu@gmail.com/



On the Feasibility of Stealthily Introducing Vulnerabilities in Open-Source Software via Hypocrite Commits

Qiushi Wu and Kangjie Lu
University of Minnesota
{wu000273, kjlu}@umn.edu

***Abstract*—Open source software (OSS) has thrived since the forming of Open Source Initiative in 1998. A prominent example is the Linux kernel, which has been used by numerous major software vendors and empowering billions of devices. The higher availability and lower costs of OSS boost its adoption, while its openness and flexibility enable quicker innovation. More importantly, the OSS development approach is believed to produce more reliable and higher-quality software since it typically has thousands of independent programmers testing and fixing bugs of the software collaboratively.**

In this paper, we instead investigate the insecurity of OSS from a critical perspective—the feasibility of stealthily introducing vulnerabilities in OSS via hypocrite commits (i.e., seemingly beneficial commits that in fact introduce other critical issues). The introduced vulnerabilities are critical because they may be stealthily exploited to impact massive devices. We first identify three fundamental reasons that allow hypocrite commits. (1) OSS is open by nature, so anyone from anywhere, including malicious ones, can submit patches. (2) Due to the overwhelming patches and performance issues, it is impractical for maintainers to accept preventive patches for "immature vulnerabilities". (3) OSS like the Linux kernel is extremely complex, so the patch-review process often misses introduced vulnerabilities that involve complicated semantics and contexts. We then systematically study hypocrite commits, including identifying immature vulnerabilities and potential vulnerability-introducing minor patches. We also identify multiple factors that can increase the stealthiness of hypocrite commits and render the patch-review process less effective. As proof of concept, we take the Linux kernel as target OSS and safely demonstrate that it is practical for a malicious committer to introduce use-after-free bugs. Furthermore, we systematically measure and characterize the capabilities and opportunities of a malicious committer. At last, to improve the security of OSS, we propose mitigations against hypocrite commits, such as updating the code of conduct for OSS and developing tools for patch testing and verification.

I. Introduction

Open source software (OSS) shares its source code publicly, and allows users to use, modify, and even distribute under an open-sourcing licence. Since the forming of the Open Source Initiative in 1998, OSS has thrived and become quite popular. For example, as of August 2020, GitHub was reported to have over 40 million users and more than 37.6 million public repositories [19] (increased by 10 million from June 2018 [18]). It was also reported that everyone uses OSS [50] while 78% of companies run OSS [60].

OSS is praised for its unique advantages. The availability and low costs of OSS enable its quick and wide adoption. Its openness also encourages contributors; OSS typically has thousands of independent programmers testing and fixing bugs of the software. Such an open and collaborative development not only allows higher flexibility, transparency, and quicker evolution, but is also believed to provide higher reliability and security [21].

A prominent example of OSS is the Linux kernel, which is one of the largest open-source projects—more than 28 million lines of code used by billions of devices. The Linux kernel involves more than 22K contributors. Any person or company can contribute to its development, e.g., submitting a patch through git commits. To make a change of the Linux kernel, one can email the patch file (containing `git diff` information) to the Linux community. Each module is assigned with a few maintainers (the list can be obtained through the script `get_maintainer.pl`). The maintainers then manually or employ tools to check the patch and apply it if it is deemed valid. Other popular OSS, such as FreeBSD, Firefox, and OpenSSL, also adopts a similar patching process.

Because of the wide adoption, OSS like the Linux kernel and OpenSSL has become attractive targets for high-profile attacks [9, 15]. While adversaries are incentivized, it is not always easy to find an exploitable vulnerability. Popular OSS is often extensively tested by developers and users in both static and dynamic ways [63]. Even a bug was found, it may not manifest the exploitability and impacts as the adversaries wish. Thus, finding ideal exploitable vulnerabilities requires not only advanced analyses and significant efforts, but also a bit of luck.

In this paper, we instead investigate the insecurity of OSS from a critical perspective—the feasibility of a malicious committer stealthily introducing vulnerabilities such as use-after-free (UAF) in OSS through hypocrite commits (seemingly beneficial minor commits that actually introduce other critical issues). Such introduced vulnerabilities can be critical, as they can exist in the OSS for a long period and be exploited by the malicious committer to impact a massive number of devices and users. Specifically, we conduct a set of studies to systematically understand and characterize hypocrite commits, followed by our suggestions for mitigation.

We first identify three fundamental reasons that allow the hypocrite commits.

- *OSS openness.* By its nature, OSS typically allows anyone

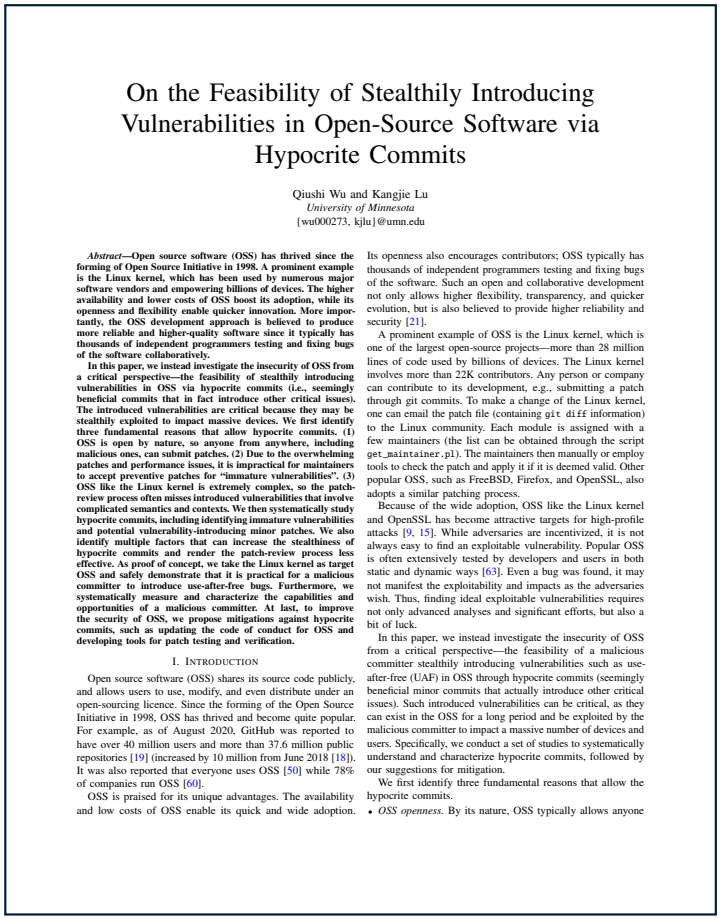
https://raw.githubusercontent.com/QiushiWu/qiushiwu.github.io/main/papers/OpenSourceInsecurity.pdf

- XX.11.2020: Veröffentlichung
- 21.11.2020: Annahme durch IEEESP
  - 2 Accepts
  - 2 Weak Accepts

$\Rightarrow$ In Top 5% der Einreichungen

Interaktion mit Personen/Umwelt

Hypocrite Commits II



On the Feasibility of Stealthily Introducing Vulnerabilities in Open-Source Software via Hypocrite Commits

Qiushi Wu and Kangjie Lu
University of Minnesota
{wu000273, kjlu}@umn.edu

***Abstract*—Open source software (OSS) has thrived since the forming of Open Source Initiative in 1998. A prominent example is the Linux kernel, which has been used by numerous major software vendors and empowering billions of devices. The higher availability and lower costs of OSS boost its adoption, while its openness and flexibility enable quicker innovation. More importantly, the OSS development approach is believed to produce more reliable and higher-quality software since it typically has thousands of independent programmers testing and fixing bugs of the software collaboratively.**

In this paper, we instead investigate the insecurity of OSS from a critical perspective—the feasibility of stealthily introducing vulnerabilities in OSS via hypocrite commits (i.e., seemingly beneficial commits that in fact introduce other critical issues). The introduced vulnerabilities are critical because they may be stealthily exploited to impact massive devices. We first identify three fundamental reasons that allow hypocrite commits. (1) OSS is open by nature, so anyone from anywhere, including malicious ones, can submit patches. (2) Due to the overwhelming patches and performance issues, it is impractical for maintainers to accept preventive patches for "immature vulnerabilities". (3) OSS like the Linux kernel is extremely complex, so the patch-review process often misses introduced vulnerabilities that involve complicated semantics and contexts. We then systematically study hypocrite commits, including identifying immature vulnerabilities and potential vulnerability-introducing minor patches. We also identify multiple factors that can increase the stealthiness of hypocrite commits and render the patch-review process less effective. As proof of concept, we take the Linux kernel as target OSS and safely demonstrate that it is practical for a malicious committer to introduce use-after-free bugs. Furthermore, we systematically measure and characterize the capabilities and opportunities of a malicious committer. At last, to improve the security of OSS, we propose mitigations against hypocrite commits, such as updating the code of conduct for OSS and developing tools for patch testing and verification.

I. Introduction

Open source software (OSS) shares its source code publicly, and allows users to use, modify, and even distribute under an open-sourcing licence. Since the forming of the Open Source Initiative in 1998, OSS has thrived and become quite popular. For example, as of August 2020, GitHub was reported to have over 40 million users and more than 37.6 million public repositories [19] (increased by 10 million from June 2018 [18]). It was also reported that everyone uses OSS [50] while 78% of companies run OSS [60].

OSS is praised for its unique advantages. The availability and low costs of OSS enable its quick and wide adoption. Its openness also encourages contributors; OSS typically has thousands of independent programmers testing and fixing bugs of the software. Such an open and collaborative development not only allows higher flexibility, transparency, and quicker evolution, but is also believed to provide higher reliability and security [21].

A prominent example of OSS is the Linux kernel, which is one of the largest open-source projects—more than 28 million lines of code used by billions of devices. The Linux kernel involves more than 22K contributors. Any person or company can contribute to its development, e.g., submitting a patch through git commits. To make a change of the Linux kernel, one can email the patch file (containing `git diff` information) to the Linux community. Each module is assigned with a few maintainers (the list can be obtained through the script `get_maintainer.pl`). The maintainers then manually or employ tools to check the patch and apply it if it is deemed valid. Other popular OSS, such as FreeBSD, Firefox, and OpenSSL, also adopts a similar patching process.

Because of the wide adoption, OSS like the Linux kernel and OpenSSL has become attractive targets for high-profile attacks [9, 15]. While adversaries are incentivized, it is not always easy to find an exploitable vulnerability. Popular OSS is often extensively tested by developers and users in both static and dynamic ways [63]. Even a bug was found, it may not manifest the exploitability and impacts as the adversaries wish. Thus, finding ideal exploitable vulnerabilities requires not only advanced analyses and significant efforts, but also a bit of luck.

In this paper, we instead investigate the insecurity of OSS from a critical perspective—the feasibility of a malicious committer stealthily introducing vulnerabilities such as use-after-free (UAF) in OSS through hypocrite commits (seemingly beneficial minor commits that actually introduce other critical issues). Such introduced vulnerabilities can be critical, as they can exist in the OSS for a long period and be exploited by the malicious committer to impact a massive number of devices and users. Specifically, we conduct a set of studies to systematically understand and characterize hypocrite commits, followed by our suggestions for mitigation.

We first identify three fundamental reasons that allow the hypocrite commits.

- *OSS openness.* By its nature, OSS typically allows anyone

https://raw.githubusercontent.com/QiushiWu/qiushiwu.github.io/main/papers/OpenSourceInsecurity.pdf

- XX.11.2020: Veröffentlichung
- 21.11.2020: Annahme durch IEEESP
  - 2 Accepts
  - 2 Weak Accepts

  $\Rightarrow$ In Top 5% der Einreichungen
- ...
- 21.04.2021: Greg KH „verbannt“ künftige Beiträge der UMN aus dem Linux Kernel
- 26.04.2021: Das Papier wird zurückgezogen

Kritikpunkte aus der Untersuchung durch das IEEESSP-Programmkomitee[4] basierend auf dem Menlo Report:

- **Einwilligung und Autonomie der Kernelentwickler**

  *The Menlo report says that "**Research** involving information and communication technology (ICT) also **raises the potential for harms to secondary stakeholders who**, while not the direct subjects of research, may also **have the right to autonomy**. When considering informed consent, we suggest researchers and research ethics boards (REBs) carefully explore the complex interconnected relationships between users and the myriad of organizations which provide ICT services." Whether this research constituted direct human-subjects research remains subject to considerable debate among the PC, but **there is no doubt that the autonomy of secondary stakeholders was violated**.*

- **Risiko durch eingebrachte Fehler**

  *The Menlo report requires "**appropriately balancing probable harm and likelihood of enhanced welfare** ... diligent analysis of how harms are minimized and benefits are maximized ... and implementing these evaluations into the research methodology." After extensive discussion, the PC believes that there were alternative research methods (for example, a controlled experiment on a simulated open-source project) that would have produced equivalent or better scientific value with much less potential for harm.*

Hervorhebung durch uns

[4] https://www.ieee-security.org/TC/SP2021/downloads/2021_PC_Statement.pdf

Folgen

- UMN: Verplichtende Ethikschulung für Doktoranden
- Konferenz: Einrichtung einer Ethikkommission
- Kernel: Revision aller Patches der UMN
- ...
- Flurschaden? Chance?

Mehr Informationen:

- Stellungnahme des Linux Technical Advisory Board
  https://lore.kernel.org/lkml/202105051005.49BFABCE@keescook/
- Stellungnahme der Autoren
  https://www-users.cs.umn.edu/~kjlu/papers/clarifications-hc.pdf
- Stellungnahme der Konferenz/IEEE
  https://www.ieee-security.org/TC/SP2021/downloads/2021_PC_Statement.pdf

Problematik Dual-Use

Dual-Use Problematik

Doppelverwendungsfähigkeit

- Technologieverwendung für zivile & militärische Zwecke
- Problem: Wie verhindern, dass Technologie *in die falschen Hände* gerät?
- Anwendungsbeispiele
  - Drohne für Transport von Medikamenten
  - Drohne für Transport von Sprengkörpern

Mehr Informationen für zivile Einsatzszenarien von Drohnen:

- Ärzte ohne Grenzen, Medikamententransport https://www.aerzte-ohne-grenzen.de/sites/germany/files/attachments/aerz_967.1_akut-2-2016_web.pdf
- Drones in Humanitarian Action (FSD) https://europa.eu/capacity4dev/innov-aid/documents/drones-humanitarian-action-survey-perceptions-and-applications

Auszug Leitbild der FAU

Die FAU ist sich als öffentliche Einrichtung der gesellschaftlichen Folgenverantwortung ihrer Forschung bewusst. Durch ihren Beitrag zu transparenter, öffentlicher und interdisziplinärer Diskussion kommt sie der Einhaltung von anerkannten ethischen und moralischen Standards auf nationaler und internationaler Ebene nach. Verantwortungsbewusstes Handeln wird von ihr gefördert und resultiert im gerechten und friedlichen Zusammenleben zwischen Menschen, Kulturen und Nationen.

Mehr Informationen:

- Arbeitskreis Zivilklausel https://stuve.fau.de/friedlich
- Vollständiges Leitbild https://www.fau.de/fau/willkommen-an-der-fau/leitbild/
- Positionspapier Stuve https://stuve.fau.de/blog/wp-content/uploads/2011/04/stuve-positionspapier-zivilklausel-1.pdf

Verantwortung in Ingenieursberufen

Definition “Ingenieur”

Was bedeutet Ingenieur/in?

Definition “Ingenieur”

Duden Definition[5] — Was bedeutet Ingenieur/in?

*auf einer Hoch- oder Fachschule ausgebildeter **Techniker** (**Berufsbezeichnung**)*

Hervorhebung durch uns

[5] https://www.duden.de/rechtschreibung/Ingenieur, abgerufen am 01.06.2021

Definition “Ingenieur”

Duden Definition[5] — Was bedeutet Ingenieur/in?

*auf einer Hoch- oder Fachschule ausgebildeter **Techniker** (**Berufsbezeichnung**)*

Wer ist Ingenieur/in?

Hervorhebung durch uns

[5] https://www.duden.de/rechtschreibung/Ingenieur, abgerufen am 01.06.2021

Definition “Ingenieur”

Duden Definition[5] — Was bedeutet Ingenieur/in?

*auf einer Hoch- oder Fachschule ausgebildeter **Techniker** (**Berufsbezeichnung**)*

Art. 2 Abs. 1 BayIngG[6] — Wer ist Ingenieur/in?

***Die Berufsbezeichnung Ingenieurin oder Ingenieur** allein oder in einer Wortverbindung **darf führen**,*

*1. wer ein **grundständiges Studium** an einer **staatlichen oder staatlich anerkannten deutschen Hochschule** mit Erfolg abgeschlossen hat*
*a) in einer **technisch-naturwissenschaftlichen Fachrichtung**,*
*b) das eine **Regelstudienzeit von mindestens sechs Semestern in Vollzeit** aufweist und mit dem bei Anwendung des ECTS-Systems **mindestens 180 Punkte** erworben werden können und*
*c) in dem **die Bereiche Mathematik, Informatik, Naturwissenschaften und Technik überwiegen**; diese Voraussetzung gilt nicht für das Führen der Berufsbezeichnung ausschließlich in der Wortverbindung Wirtschaftsingenieurin oder Wirtschaftsingenieur durch Personen, die ein grundständiges Studium des Wirtschaftsingenieurwesens absolviert haben,*

2. wer nach Ausbildung im Ausland die Genehmigung hierzu erhalten hat,

3. wer nach dem Recht eines anderen Landes der Bundesrepublik Deutschland hierzu berechtigt ist oder

4. wer bis zum Inkrafttreten dieses Gesetzes hierzu berechtigt war.

Hervorhebung durch uns

[5] https://www.duden.de/rechtschreibung/Ingenieur, abgerufen am 01.06.2021

[6] https://www.gesetze-bayern.de/Content/Document/BayIngG2016/true, abgerufen am 01.06.2021

Verantwortung in Ingenieursberufen

Verantwortung in Ingenieursberufen

VDI: Ethische Grundsätze des Ingenieurberufs[7], 1.1

> ***Igenieurinnen und Ingenieure*** *sind alleine oder – bei arbeitsteiliger Zusammenarbeit –* ***mitverantwortlich*** *für die* ***Folgen*** *ihrer beruflichen Arbeit sowie für die sorgfältige Wahrnehmung ihrer spezifischen Pflichten, die ihnen aufgrund ihrer Kompetenz und ihres Sachverstandes zukommen.*

Hervorhebung durch uns

[7] https://www.vdi.de/ueber-uns/presse/publikationen/details/ethische-grundsaetze-des-ingenieurberufs, idF. 03/2002, S. 4

Verantwortung in Ingenieursberufen

VDI: Ethische Grundsätze des Ingenieurberufs[7], 1.1

> ***Igenieurinnen und Ingenieure*** *sind alleine oder – bei arbeitsteiliger Zusammenarbeit –* ***mitverantwortlich*** *für die* ***Folgen*** *ihrer beruflichen Arbeit sowie für die sorgfältige Wahrnehmung ihrer spezifischen Pflichten, die ihnen aufgrund ihrer Kompetenz und ihres Sachverstandes zukommen.*

Viele offene Fragen

- Was bedeutet Verantwortung? (Literatur: [8])
- Wann spricht man von einer Technikfolge und worauf wirken sich diese aus? (Literatur: [9])

$\Rightarrow$ Zu tiefgreifend, wir verwenden heute die “landläufigen” Begriffe

Hervorhebung durch uns

[7] https://www.vdi.de/ueber-uns/presse/publikationen/details/ethische-grundsaetze-des-ingenieurberufs, idF. 03/2002, S. 4

[8] Werner, Micha H.: “Verantwortung”. In: Handbuch Technikethik. Grunwald, Armin (Hg.). J.B. Metzler. Stuttgart 2013. S. 38–43. https://doi.org/10.1007/978-3-476-05333-6_7

[9] Decker, Michael: “Technikfolgen”. In: Handbuch Technikethik. Grunwald, Armin (Hg.). J.B. Metzler. Stuttgart 2013. S. 33–38. https://doi.org/10.1007/978-3-476-05333-6_6

Diskussion: Technikfolgen in Vergangenheit und Gegenwart

Zitat

I have always wished for my computer to be as easy to use as my telephone; my wish has come true because I can no longer figure out how to use my telephone

Diskussion: Technikfolgen in Vergangenheit und Gegenwart

Zitat

> *I have always wished for my computer to be as easy to use as my telephone; my wish has come true because I can no longer figure out how to use my telephone*

Bjarne Stroustrup, Begründer von C++, um 1990[10]

[10] https://www.stroustrup.com/quotes.html, abgerufen am 01.06.2021

Diskussion: Technikfolgen in Vergangenheit und Gegenwart

Zitat

> *I have always wished for my computer to be as easy to use as my telephone; my wish has come true because I can no longer figure out how to use my telephone*

Bjarne Stroustrup, Begründer von C++, um 1990[10]

IBM Simon von 1992[11]

[10] https://www.stroustrup.com/quotes.html, abgerufen am 01.06.2021

[11] Bcos47, Public domain, via Wikimedia Commons

Verantwortung in Ingenieursberufen – Nochmal

VDI: Ethische Grundsätze des Ingenieurberufs[12], 1.1

> ***Igenieurinnen und Ingenieure*** *sind alleine oder – bei arbeitsteiliger Zusammenarbeit –* ***mitverantwortlich*** *für die* ***Folgen*** *ihrer beruflichen Arbeit sowie für die sorgfältige Wahrnehmung ihrer spezifischen Pflichten, die ihnen aufgrund ihrer Kompetenz und ihres Sachverstandes zukommen.*

Hervorhebung durch uns

Verantwortung in Ingenieursberufen – Nochmal

VDI: Ethische Grundsätze des Ingenieurberufs[12], 1.1

> ***Igenieurinnen und Ingenieure*** *sind alleine oder – bei arbeitsteiliger Zusammenarbeit –* ***mitverantwortlich*** *für die Folgen ihrer beruflichen Arbeit sowie für die sorgfältige* ***Wahrnehmung ihrer spezifischen Pflichten****, die ihnen* ***aufgrund ihrer Kompetenz und ihres Sachverstandes*** *zukommen.*

Hervorhebung durch uns

[12] https://www.vdi.de/ueber-uns/presse/publikationen/details/ethische-grundsaetze-des-ingenieurberufs, idF. 03/2002, S. 4

Verantwortung in Ingenieursberufen – Nochmal

VDI: Ethische Grundsätze des Ingenieurberufs[12], 1.1

> ***Igenieurinnen und Ingenieure*** *sind alleine oder – bei arbeitsteiliger Zusammenarbeit –* ***mitverantwortlich*** *für die Folgen ihrer beruflichen Arbeit sowie für die sorgfältige* ***Wahrnehmung ihrer spezifischen Pflichten****, die ihnen* ***aufgrund ihrer Kompetenz und ihres Sachverstandes*** *zukommen.*

Interpretation (nicht abschließend...)

- Prospektiv Verantwortung übernehmen
- Technikfolgen interdisziplinär betrachten
- Kritisch und reflektiert Handeln
- Informiert bleiben

Hervorhebung durch uns

[12] https://www.vdi.de/ueber-uns/presse/publikationen/details/ethische-grundsaetze-des-ingenieurberufs, idF. 03/2002, S. 4