“Hypocrite Commits” Legal and Ethical Limits to Cybersecurity Research

Ivo Emanuilov

KU Leuven Centre for IT & IP Law

ivo.emanuilov@kuleuven.be

October 7, 2021

IAAL but IANYL and TINLA

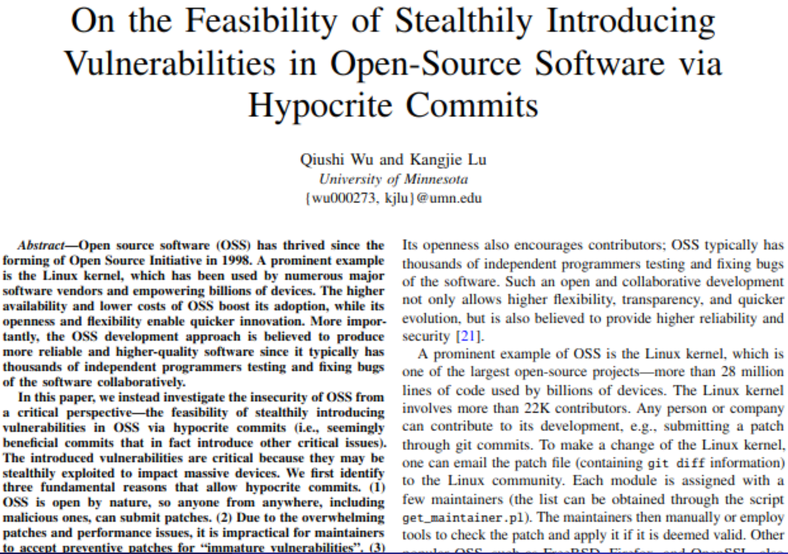
Outline

1. Introduction
    - What are “hypocrite commits”?
    - Linux kernel’s development model
2. Chain of events: the Hyprocrite Commits paper and the community backlash
3. The ethical side of “hypocrite commits”
4. The legal side of “hypocrite commits”
5. Discussion

Introduction

"Research is what I am doing when I do not know what I am doing."
Wernher von Braun

What are “hypocrite commits”?



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

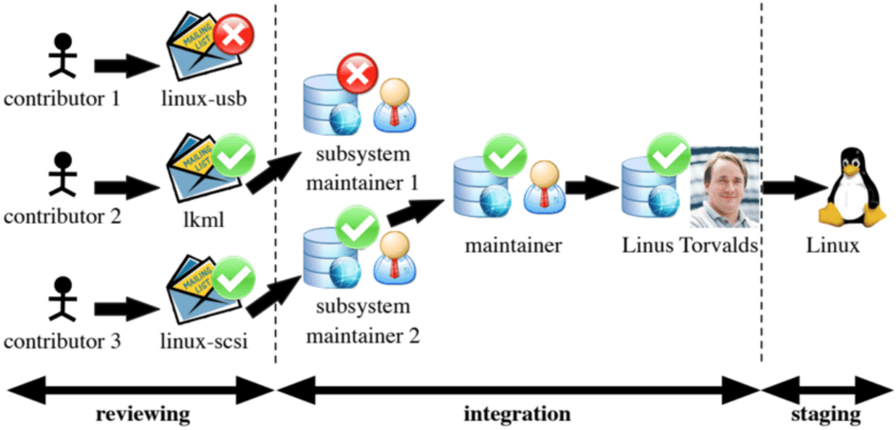
***Abstract*—Open source software (OSS) has thrived since the forming of Open Source Initiative in 1998. A prominent example is the Linux kernel, which has been used by numerous major software vendors and empowering billions of devices. The higher availability and lower costs of OSS boost its adoption, 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)

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

Linux kernel's development model



Jiang et al., 2013

What’s in a hypocrite commit?

```
/*Introducing: CVE-2019-12819*/

int __mdiobus_register(...) {
    ...
    err = device_register(&bus->dev);
    if (err) {
        pr_err("mii_bus %s failed to register\n",
                            bus->id);
+       put_device(&bus->dev);
        return -EINVAL;
    }
}
```

Chain of events

Chain of events

2018

UMN bug-fix research on Linux kernel starts, and roughly 400 bug-fix patches are contributed over the next two years, mainly centered around specific research papers

August 2020

"Hypocrite Commits" patches from UMN researchers sent to kernel developers under false identities.

November 2020

(1) "Hypocrite Commits" paper is published + accepted by IEEE Symposium on Security and Privacy

(2) Sarah Jamie Lewis calls attention to paper's ethics

Chain of events

December 2020

(1) Sarah Jamie Lewis & others send a letter to IEEESSP

(2) UMN IRB appears to give an exemption to the research

April 2021

(1) Poor quality patches sent by UMN after 7 months of silence

(2) Greg Kroah-Hartman asks submitters to stop sending poor quality patches under the guise of "research on maintainers"

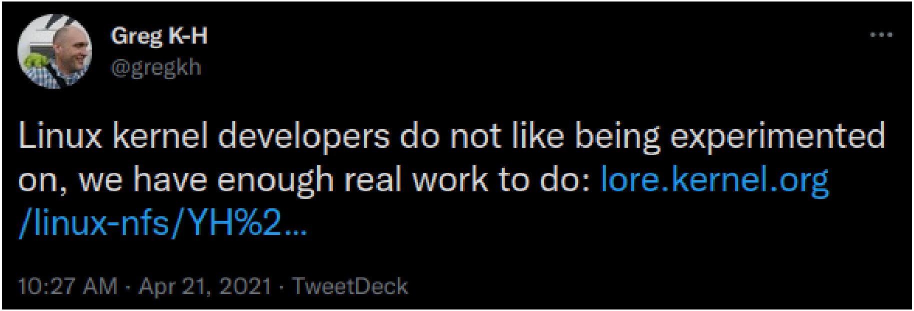
(3) Linux Foundation sends letter to UMN...

May 2021

Linux Foundation's TAB publishes a technical report

The Backlash and the TAB Report

The community backlash

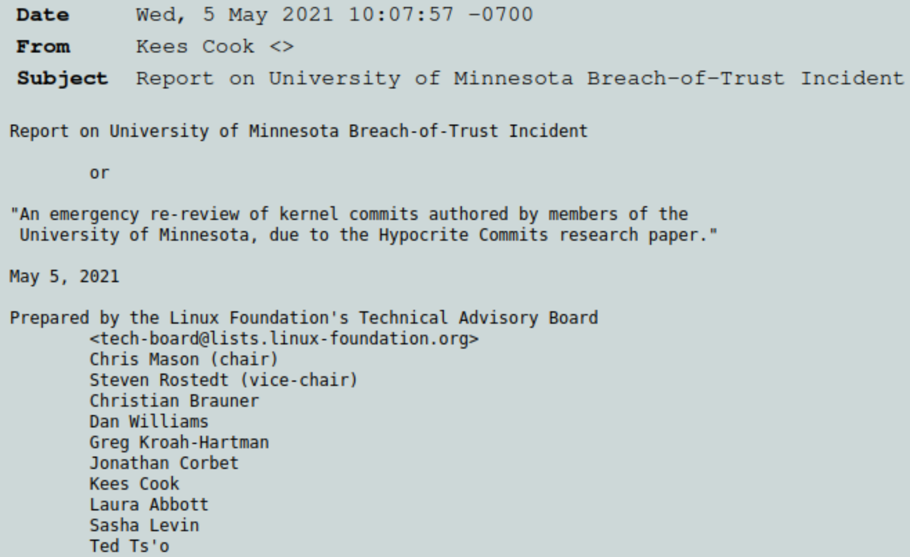


Greg K-H
@gregkh

Linux kernel developers do not like being experimented on, we have enough real work to do: lore.kernel.org/linux-nfs/YH%2...

10:27 AM · Apr 21, 2021 · TweetDeck

The TAB report



Date Wed, 5 May 2021 10:07:57 -0700

From Kees Cook <>

Subject Report on University of Minnesota Breach-of-Trust Incident

```
Report on University of Minnesota Breach-of-Trust Incident

        or

"An emergency re-review of kernel commits authored by members of the
 University of Minnesota, due to the Hypocrite Commits research paper."

May 5, 2021

Prepared by the Linux Foundation's Technical Advisory Board
        <tech-board@lists.linux-foundation.org>
        Chris Mason (chair)
        Steven Rostedt (vice-chair)
        Christian Brauner
        Dan Williams
        Greg Kroah-Hartman
        Jonathan Corbet
        Kees Cook
        Laura Abbott
        Sasha Levin
        Ted Ts'o
```

The Ethical Side

The Ethical Side

POC: Ethical Considerations

1. "**Ensuring the safety of the experiment**"

   "Our goal is not to introduce vulnerabilities to harm OSS (...) We also prepare the correct patches for fixing the minor issues (...) Once a maintainer confirmed our patches, e.g., an email indicating "looks good", we immediately notify the maintainers of the introduced UAF and request them to not go ahead to apply the patch"

2. ***Potential* "human research concerns"**

   "...studies issues with the patching process instead of individual behaviors, and we do not collect any personal information (...) Bug-introducing patch is a known problem in the Linux community [28, 67]. We also informed the community that malicious committers could intentionally introduce bugs, and they acknowledged that they knew patches could introduce further bugs, and they will do their best to review them"

TAB Report: Recommendations

- UMN must improve the quality of the changes that are proposed for inclusion into the kernel
- TAB will be working with researchers to develop a document describing a set of best practices for researchers to follow when working with the kernel (and open-source projects in general) community (...) living document, maintained in the kernel documentation tree and evolved over time as needed.

TAB Report: Recommendations

"This incident has highlighted the differences in the motivations behind the kernel development and research communities. While both groups are interested in making a better kernel for the massive user community, **kernel developers** *tended to be focused on engineering processes, reviewing contributions, and mentoring new contributors, while* **researchers** *tend to be focused on exploring new ideas and methodologies, developing new tools, and furthering their understanding of how development communities interact. There is a lot of value created by both groups, but they can also occasionally lead to conflict."*

Was the paper’s methodology sound?

Methodology

New vulnerability-introducing method

1. Is the threat model of hypocrite commits correct?
2. Was this really a "safe" proof-of-concept?
3. (When) are these methods for academic research sound?

Methodology

Full Disclosure on the Case Study, 27 April 2021

The vulnerability-introducing method proposed in the paper is about how to introduce a vulnerability condition (not a vulnerability by itself) through a minor patch in such a way that it turns an “immature vulnerability” into a real one without being detected during the patch review process. To show that this was a real problem, we carried out a proof-of-concept case study by starting the process of proposing 3 “hypocrite patches” (patches introducing a vulnerability condition, each has 1-4 lines of code change, thus is called “minor patch”). The patches by themselves do not contain vulnerabilities. These patches were proposed in a way that would prevent them from ever being introduced into the kernel. All of the ”hypocrite patches” were aborted before they could move forward, as was the plan and as can be seen in the following details.

The Legal Side

The Legal Side

POC: Legal Considerations

1. **Committer Liability**

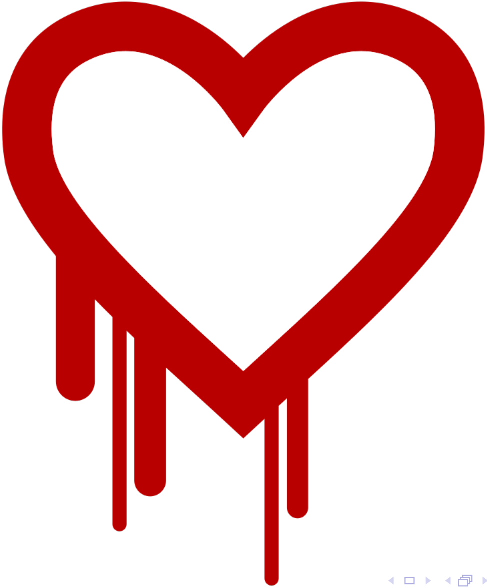
   "OSS openly encourages contributors. Committers can freely submit patches without liability. We believe that an effective and immediate action would be to update the code of conduct of OSS, such as adding a term like "by submitting the patch, I agree to not intend to introduce bugs." Only committers who agreed to it would be allowed to go ahead to submit the patches. By introducing the liability, the OSS would not only discourage malicious committers but also raise the awareness of potential introduced bugs for benign committers"

2. **Committer Accountability**

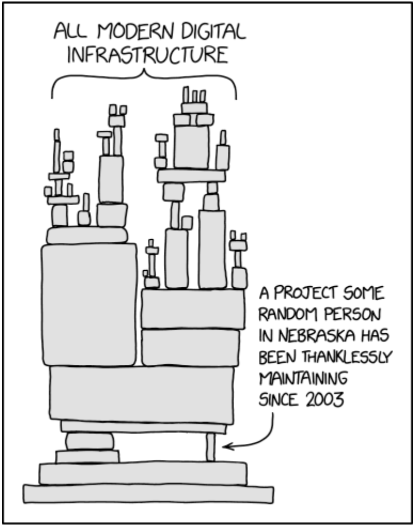
   "Verifying the identities of committers, i.e., introducing the accountability, is also an effective mitigation against hypocrite commits [but] checking the identity over online social networks is a challenging problem."

What if this was Heartbleed?

Heartbleed: ‘open source’s worst hour’



The problem with choosing your subject of research carefully...


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

Is there a Safe Harbor for OSS developers?

OSSH Study (2021): false sense of security for developers

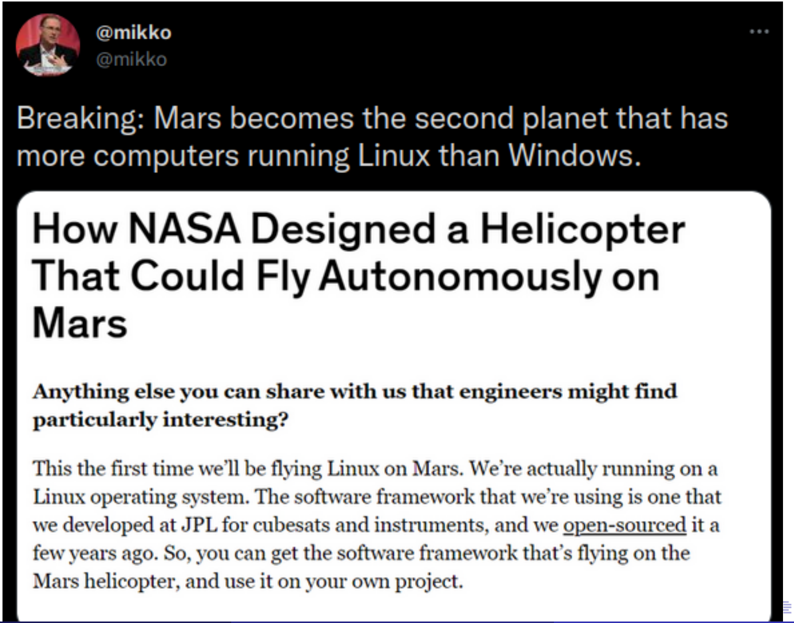
License clauses

1. Existing clauses seeking to exclude liability which can be found in almost all OSS and OSH licenses provide a false sense of security to developers.
2. Questionable enforceability, bearing in mind both domestic legislation in Member States, and EU legislation such as the General Product Safety Directive
3. No direct nexus between ownership of intellectual property and liability for its content, so it is possible that there is no opportunity for a developer to apply an exclusion clause in any event.
4. Consideration of the extent to which it is appropriate to extend liability to individual developers contributing to designs or code which are then incorporated into products and placed on the market.

Discussion

1. What role for research ethics boards, esp. if lacking in technical expertise?
2. When does a research project cross the boundaries? When is ‘research’ no longer research? Do we need a new understanding for research implicating ‘human subjects’?
3. How is the Hypocrite Commits research different from ‘traditional’ cybersecurity research?
4. Should there be ‘no-go’ ethical boundaries for safety-critical projects, such as the Linux kernel?
5. Is or should the researcher or their employer be potentially liable in such cases?

Linux on Mars



@mikko
@mikko

Breaking: Mars becomes the second planet that has more computers running Linux than Windows.

How NASA Designed a Helicopter That Could Fly Autonomously on Mars

Anything else you can share with us that engineers might find particularly interesting?

This the first time we'll be flying Linux on Mars. We're actually running on a Linux operating system. The software framework that we're using is one that we developed at JPL for cubesats and instruments, and we open-sourced it a few years ago. So, you can get the software framework that's flying on the Mars helicopter, and use it on your own project.

Your questions?

Contact details

Ivo Emanuilov
Centre for IT & IP Law
KU Leuven
ivo.emanuilov@kuleuven.be
@IvoEmanuilov

Important Notice

The views and opinions expressed in this presentation are those of the presenter unless identified as those of other parties. The information contained herein is of a general nature and is intended for educational and instructional purposes only. Although the presenter has strived to provide accurate and timely information, there can be no guarantee that such information is accurate as of the date it is received or that it will continue to be accurate in the future. No one should act on such information without appropriate professional and scholarly advice after a thorough examination of the particular situation.