THE LINUX FOUNDATION PROJECTS

CIVIL INFRASTRUCTURE PLATFORM

Summary

The Tip of the Iceberg: On the Merits of Finding Security Bugs

Dinesh Kumar, Toshiba Software India

23/Nov/2020

About the Research paper

- About the paper
  - In this paper researchers investigated a fundamental question regarding FOSS security
  - ***Is the security(w.r.t. security bugs) of software releases increasing over time?***
  - This investigation is based on Debian Linux distribution covering multiple versions
  - This paper was published in ACM journal Sept-20 https://dl.acm.org/doi/10.1145/3406112

- Authors

  - NIKOLAOS ALEXOPOULOS, Technical University of Darmstadt, Germany
  - SHEIKH MAHBUB HABIB, Continental AG, Germany
  - STEFFEN SCHULZ, Intel Labs, Germany
  - MAX MÜHLHÄUSER, Technical University of Darmstadt, Germany

Reference https://fileserver.tk.informatik.tu-darmstadt.de/Publications/2020/alexopoulos2020TOPS.pdf

CIVIL INFRASTRUCTURE PLATFORM

Introduction

- For analyzing whether the security of software increasing or are we are introducing more bugs than getting fixed
- Authors developed DVAF(Debian Vulnerability Analysis Framework)
- Vulnerabilities such as *shellshock(CVE-2014-6271), Heartbleed(CVE-2014-0160)* did great damage. However zero-days vulnerabilities are even more damaging
- Even though security community has come up with various defense mechanism, still manual verification requires significant manual effort
- Debian has been chosen for this study because of
  - It is one of the biggest and most popular collection of FOSS
  - Policy of only adopting critical patches in stable releases
  - Security is handled by a dedicated team with transparent workflows, status reports and public reports
- While the dataset is limited to Debian, it is likely that the results can be generalized to all general purpose Linux distributions

Debian Vulnerability Analysis Framework(DVAF)

- System for automatically collecting relevant data from publicly available sources NVD(National Vulnerability Database) & DSA(Debian Security Advisory)
- The dataset considered for the analysis is for LTS versions of Debian
- While the dataset is limited to Debian, it is likely that the results can be generalized to all general purpose Linux distributions

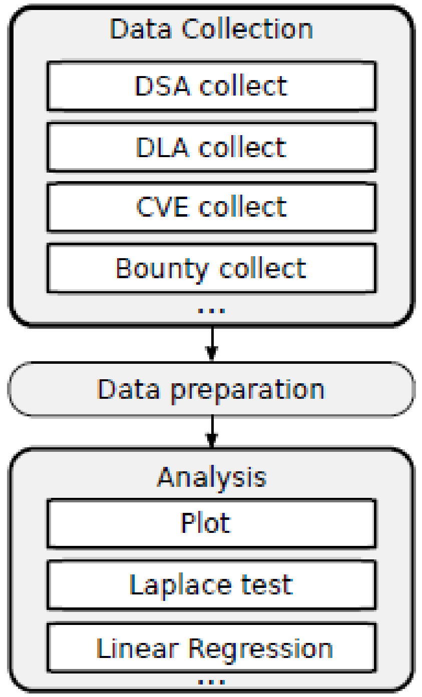
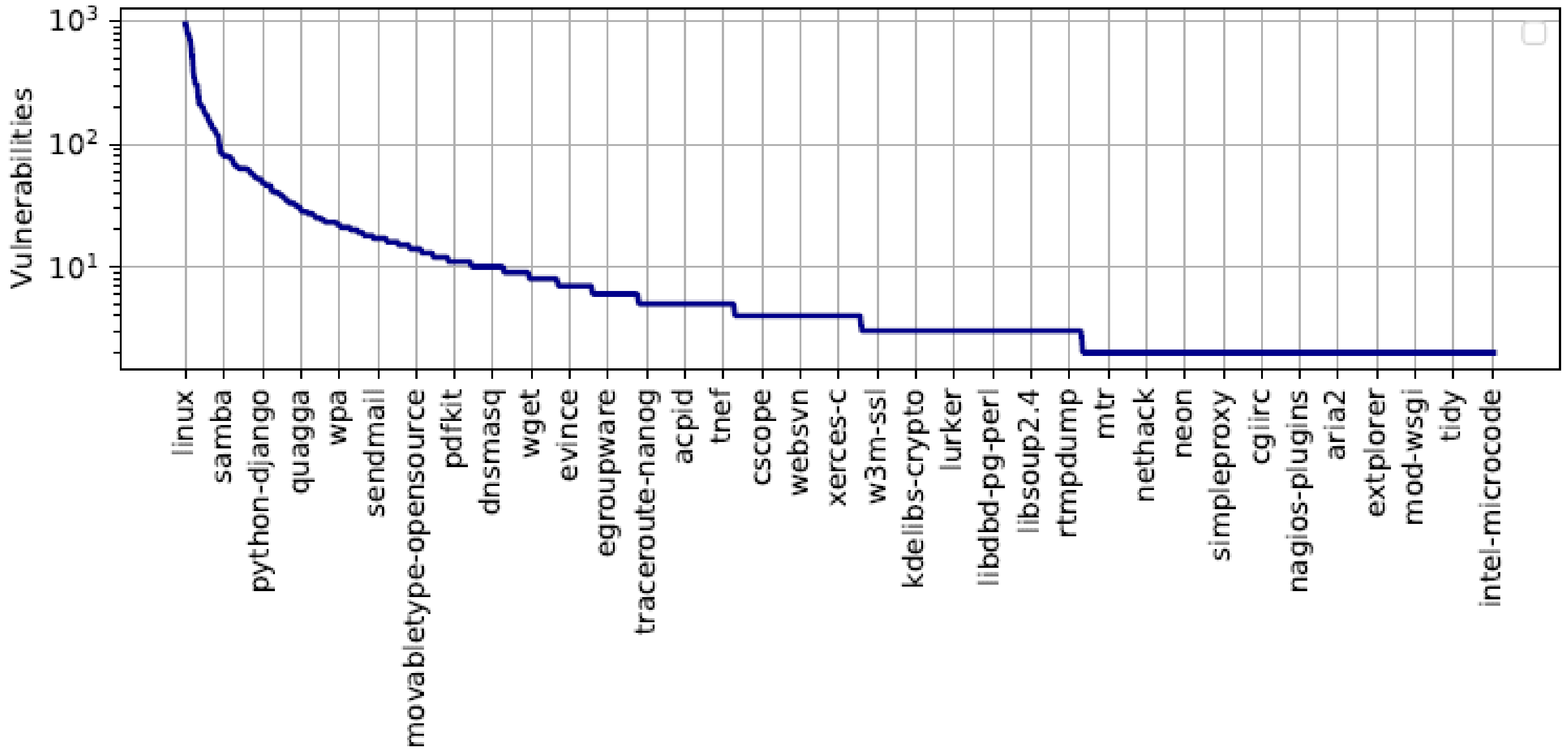


Fig. 1. The DVAF's extendable architecture and workflow

DLA->Debian Long term advisory
DSA->Debian Security advisory

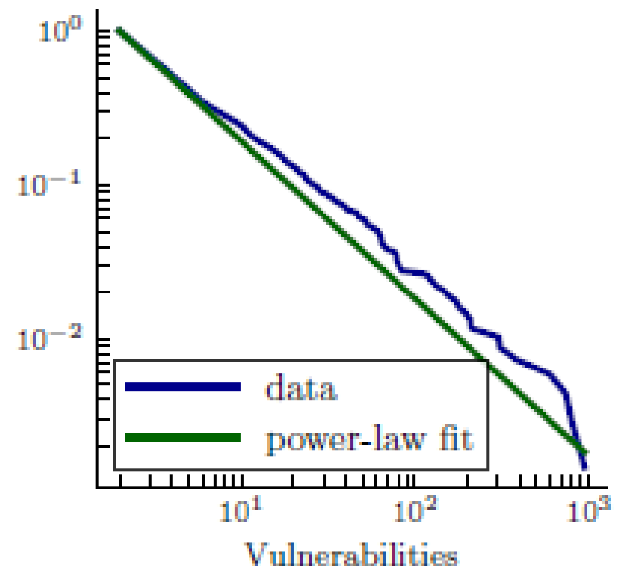
Vulnerabilities in Debian(2001~2018)



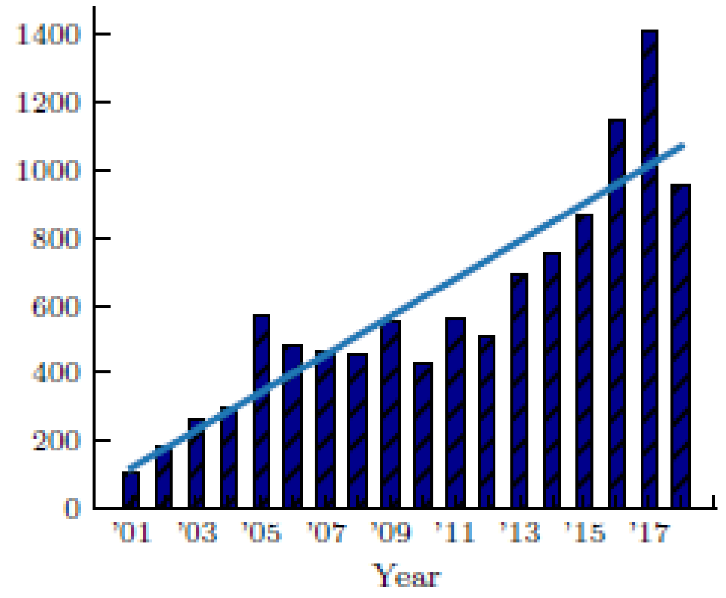
The distribution of vulnerabilities per package (years 2001-2018). Every twentieth package name appears on the x axis for space reasons. The y axis is logarithmic. Packages with at least two vulnerabilities are taken into account.

CIVIL INFRASTRUCTURE PLATFORM

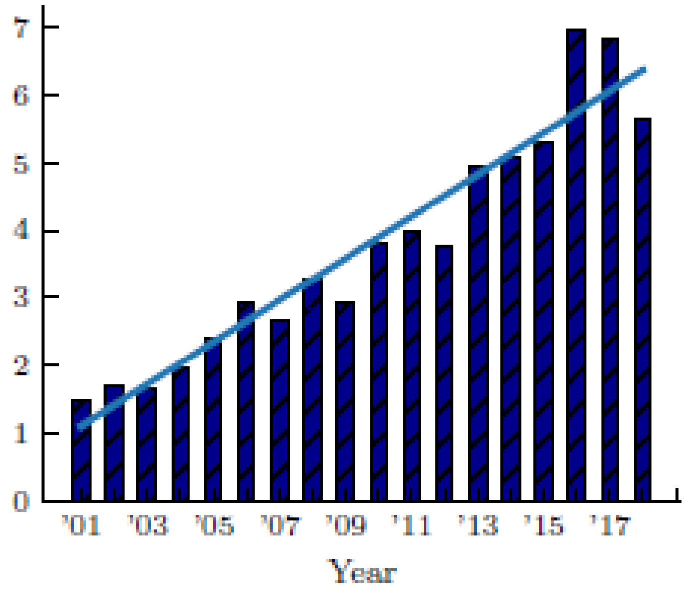
Vulnerabilities distribution and trends in Debian(2001~2018)



(a) A log-log plot (complementary cumulative distribution function) of the distribution of



(b) Total vulnerabilities reported per calendar year (2001-2018).



(c) Average vulnerabilities per package (that had at least one security incident in that year) per calendar year (2001-2018).

CIVIL INFRASTRUCTURE PLATFORM

Summary of results and contributions

- Authors approached central question in three steps by forming three main hypothesis
  - **H1:The vulnerability rate of widely used FOSS is showing signs of maturity**
    - No clear evidence of maturing behaviour found
  - **H2:There are less severe bugs and certain types are decreasing**
    - Although the ratio of high-severity vulnerabilities compared to the total is dropping, their absolute number does not show a sign of decrease
  - **H3:Vulnerability prices for FOSS in bug bounty programs are rising**
    - Investigation of a community-driven bug bounty program showed that there is no increase in the prices paid, even when considering only high severity vulnerabilities of popular FOSS

Discussions & Recommendations cont..

- Need for improved procedures
  - The maintenance of longer term branches required for studying about security
  - Stricter application of coding guidelines and testing strategy, as well as developer education on memory issues
  - Commit based static analysis methods such as VCCFinder should be pursued
  - Organizations should share threat information with each other, therefore common platform to share this information is needed
- New Detection Tools and Improvements
  - More and better ways of finding bugs during all phases of software life cycle are needed
  - Tools like the kernel fuzzer syzcaller paired with automatic continuous fuzzing of kernel branches (syzbot) are steps in the right direction
  - Google's recently launched OSS-Fuzz project27 is an interesting positive initiative and may produce measurable positive results in the near future
  - Tools like Asan(Address Sanitizer) & UBSan should be pursued

Discussions & Recommendations

- Need for more attractive bug-bounty programs for FOSS
- Need for more expressive security metrics and continuous measurement
  - Measuring the difficulty to find vulnerabilities instead of their number, would better express software quality, this is currently open problem
  - Given that the security landscape is changing at a high rate, we need studies that are aimed at continuous measurement and plug-and-play reproducibility over time
- Need for more effective mitigation measures
  - Software countermeasures, like control-flow integrity or sandboxing of components/libraries, coupled with hardware (CPU) features like NX/XD bits help to limit the effectiveness of range of vulnerabilities

Conclusion

- There are no compelling evidence suggesting that FOSS products are becoming more secure
- Overall we are not finding vulnerabilities fast enough and therefore we are not making the iceberg any smaller
- The effect of automatic analyzers and tools seems rather limited , while bug bounty programs for FOSS lack in long-term attractiveness
- However, The community is making impressive effort in producing new fuzzing and static analysis tools, in addition to hardware security features, while vulnerability reporting practices are showing signs of improvement.

Learning from this paper

- Any vulnerabilities found by any CIP members should be reported in upstream, it will help researchers to come up with more accurate results
- DVAF can be further investigated and if we can re-use it for improving CIP security, it would be a great contribution to upstream as well
- Tools recommended in this paper such as static code analysis tool, VCCFinder, syzcaller, Asan, UBSan etc. can be explored to understand if these tools can be integrated in CIP

Thanks you!

CIVIL
INFRASTRUCTURE
PLATFORM