SOLAR

БЕЗОПАСНОСТЬ SUPPLY CHAIN: ПРОТИВОСТОИМ ОПАСНЫМ ЗАВИСИМОСТЯМ

ТАТЬЯНА КУЦОВОЛ
ВЕДУЩИЙ АНАЛИТИК-ИССЛЕДОВАТЕЛЬ ИБ
ГК «СОЛАР»

t.kutsovol@rt-solar.ru
@luttatiana

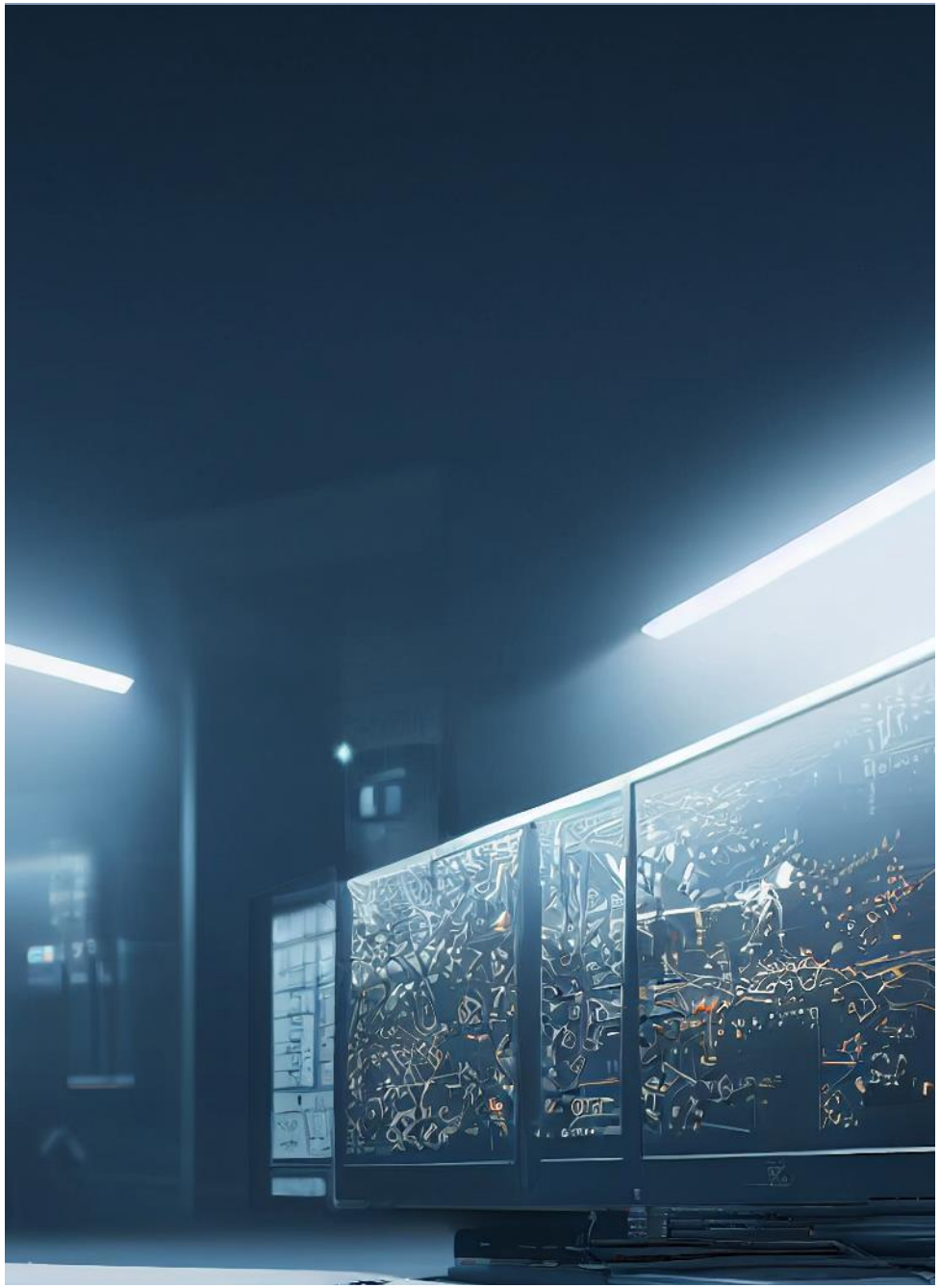
План-капкан

1. Общее понятие безопасности цепочки поставок (Supply Chain) в контексте разработки

2. Обзор мировых стандартов SLSA, OWASP SCVS, CIS SSCSG

3. Разбор атак на цепочку поставок (Supply Chain). Примеры техник злоумышленников и подходы детектирования

4. Оценка риска компрометации через Supply Chain зависимостей. Детальный обзор метрик для сторонних компонент, которые можно и нужно смотреть



Вводные

- Open Source-зависимости везде
- НЕ использовать Open Source-зависимости невозможно
- Использовать Open Source-зависимости важно безопасно

SECURE SUPPLY CHAIN

Вводные

“You can't trust code that you did not totally create yourself. Especially code from companies that employ people like me”.

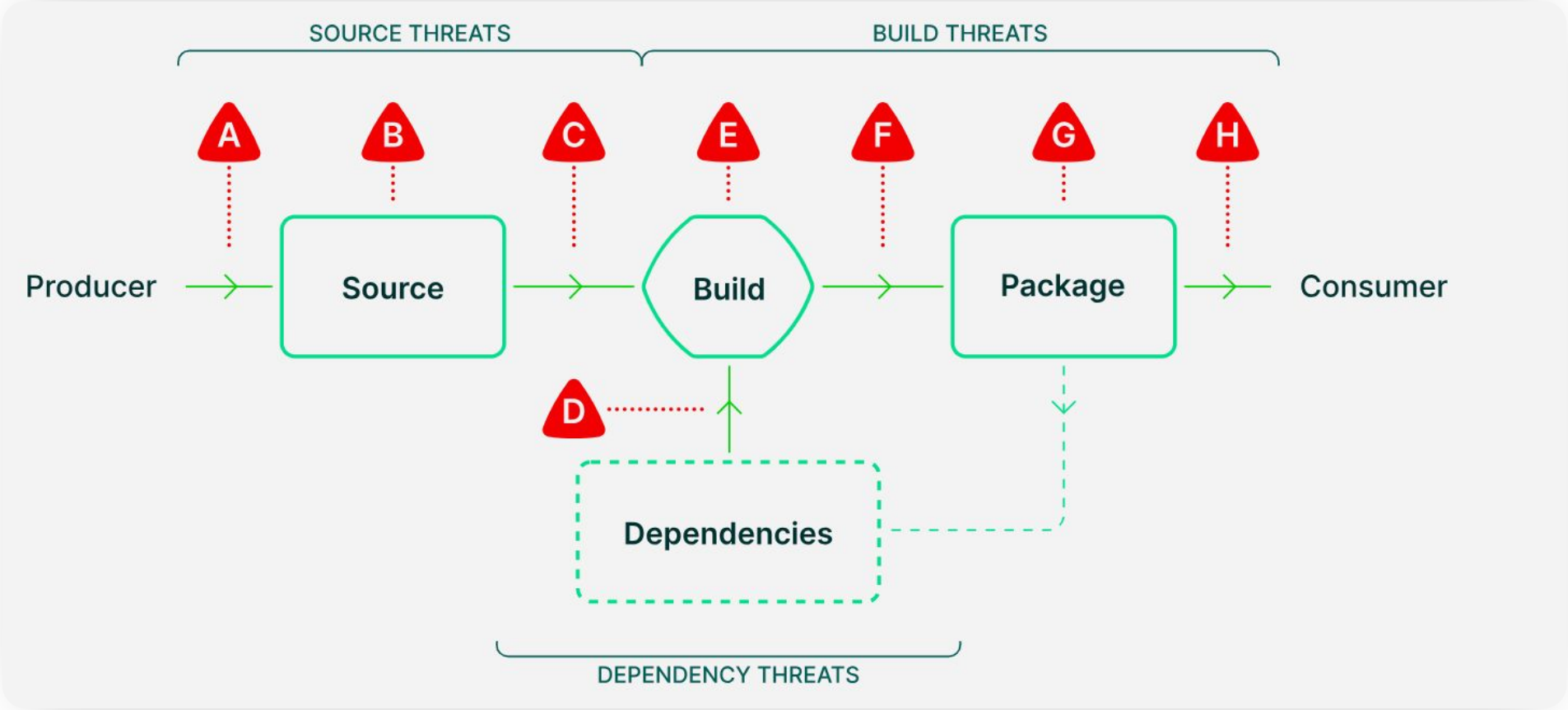
Вы можете доверять коду, только если вы написали его сами. Особенно нельзя доверять коду от компаний, которые берут на работу таких как я”.

Ken Thompson, 1984
“Reflections on Trusting Trust”

Безопасность цепочки поставок (Supply Chain)

SOLAR

Обеспечение безопасности на всех этапах пути, по которому ПО попадает в организацию, от момента его создания или покупки до этапа использования.



SOLAR

ПОЧЕМУ БЕЗОПАСНОСТЬ SUPPLY CHAIN – ПРОБЛЕМА?

Почему безопасность Supply Chain – проблема?

SOLAR

Event-stream в 2018 г. - завладели популярной библиотекой в NPM

Packt Hub

Malicious code in npm ‘event-stream’ package targets a bitcoin wallet and causes 8 million download...

Last week Ayrton Sparling, a Computer Science major at CSUF, California disclosed that the popular npm package, event-stream, contains a...

28 нояб. 2018 г.

SolarWinds в 2020 г. - поставка уязвимых обновлений для клиентов

Ведомости

Reuters: при атаке на SolarWinds хакеры украли данные контрразведки США

Хакеры, взломавшие IT-компанию SolarWinds в прошлом году, похитили данные о контрразведывательных операциях США и санкционной политике...

7 окт. 2021 г.

VMConnect в 2023 г. - вредоносный пакет в PyPI, похожий на vConnector

Bleeping Computer

Lazarus hackers deploy fake VMware PyPI packages in VMConnect attacks

North Korean state-sponsored hackers have uploaded malicious packages to the PyPI (Python Package Index) repository, camouflaging one of...

31 авг. 2023 г.

ТАК ПОЧЕМУ БЕЗОПАСНОСТЬ SUPPLY CHAIN – ПРОБЛЕМА?

Почему безопасность Supply Chain – проблема?

на 742%

в год в среднем растет количество атак
через цепочку поставок

НО ПОЧЕМУ ВСЕ ЖЕ БЕЗОПАСНОСТЬ SUPPLY CHAIN – ПРОБЛЕМА?

Почему безопасность Supply Chain – проблема?

SOLAR

THE WHITE HOUSE

Administration Priorities The Record Briefing Room Español

MAY 12, 2021

Executive Order on Improving the Nation's Cybersecurity

BRIEFING ROOM › PRESIDENTIAL ACTIONS

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered as follows:

Section 1. Policy. The United States faces persistent and increasingly sophisticated malicious cyber campaigns that threaten the public sector, the private sector, and ultimately the American people's security and privacy. The Federal Government must improve its efforts to identify, deter, protect against, detect, and respond to these actions and actors. The Federal Government must also carefully examine what occurred during any major cyber incident and apply lessons learned. But cybersecurity requires more

Почему безопасность Supply Chain – проблема?

vendor's current stage, next steps, and points of contact for questions;

(iii) incorporating automation throughout the lifecycle of FedRAMP, including assessment, authorization, continuous monitoring, and compliance;

(iv) digitizing and streamlining documentation that vendors are required to complete, including through online accessibility and pre-populated forms; and

(v) identifying relevant compliance frameworks, mapping those frameworks onto requirements in the FedRAMP authorization process, and allowing those frameworks to be used as a substitute for the relevant portion of the authorization process, as appropriate.

Sec. 4. Enhancing Software Supply Chain Security.

(a) The security of software used by the Federal Government is vital to the Federal Government's ability to perform its criti functions. The development of commercial software often lacks cy, sufficient focus on the ability of the software to resist att ate controls to prevent tampering by malicious actors. There is a eed to implement more rigorous and predictable mechanisms for ensu that products function securely, and as intended. The security and integrity of "critical software" — software that performs functions critical to trust (such as affording or requiring elevated system privileges or direct access to networking and computing resources) — is a particular concern. Accordingly, the Federal Government must take action to rapidly improve the security and integrity of the software supply chain, with a priority on addressing critical software.

Почему безопасность Supply Chain – проблема?

THE WHITE HOUSE

Administration Priorities The Record Briefing Room Español MENU

vendor's current stage, next steps, and points of contact for questions;

(iii) incorporating automation throughout the lifecycle of FedRAMP, including assessment, authorization, continuous monitoring, and compliance;

Sec. 4. Enhancing Software Supply Chain Security.

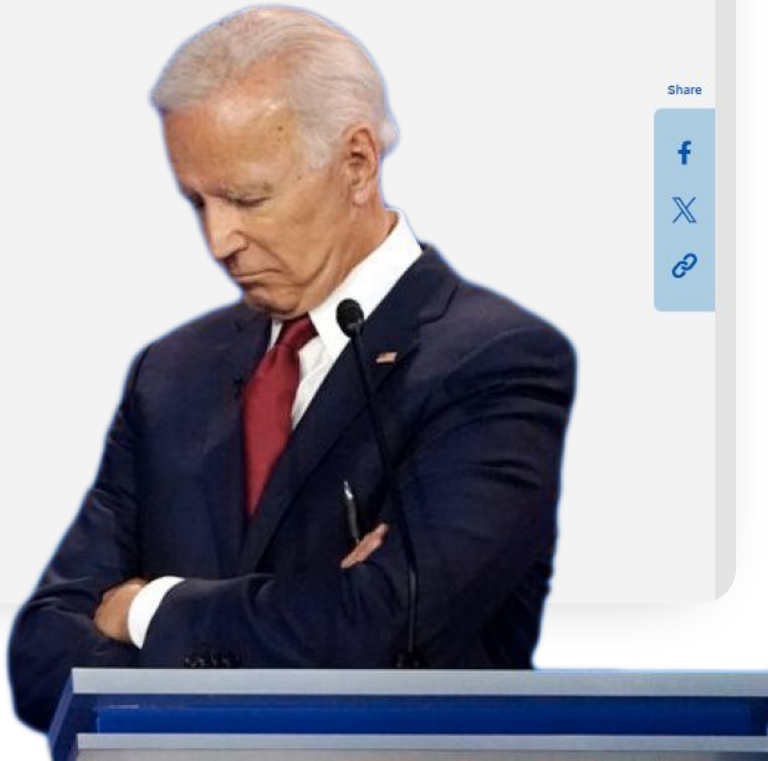
(a) The security of software used by the Federal Government is

of the authorization process, as appropriate.

Sec. 4. Enhancing Software Supply Chain Security.

(a) The security of software used by the Federal Government is vital to the Federal Government's ability to perform its critic functions. The development of commercial software often lacks cy, sufficient focus on the ability of the software to resist att ate controls to prevent tampering by malicious actors. There is a eed to implement more rigorous and predictable mechanisms for ensuring that products function securely, and as intended. The security and integrity of "critical software" — software that performs functions critical to trust (such as affording or requiring elevated system privileges or direct access to networking and computing resources) — is a particular concern. Accordingly, the Federal Government must take action to rapidly improve the security and integrity of the software supply chain, with a priority on addressing critical software.

Share

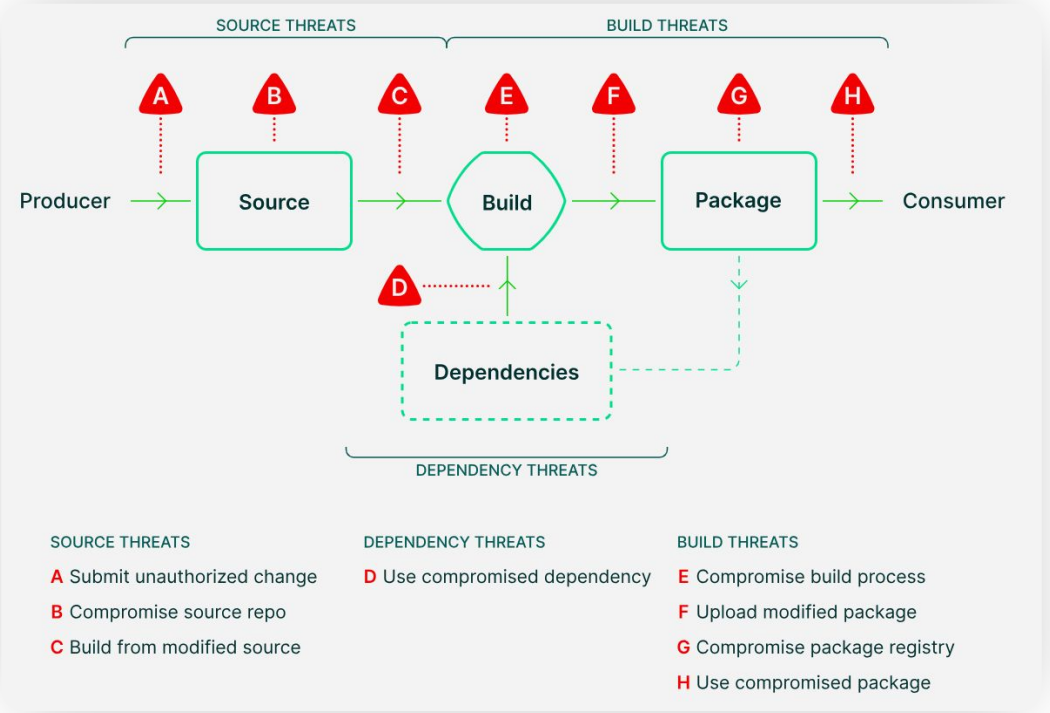


Обзор стандартов

1. SLSA (Supply-chain Levels for Software Artifacts)
2. OWASP Software Component Verification Standard (SCVS)
3. CIS Software Supply Chain Security Guide (SSCSG)

SLSA (Supply-chain Levels for Software Artifacts)

Фреймворк, чек-лист требований, которые должны быть учтены в безопасной цепочке поставок (Supply Chain)

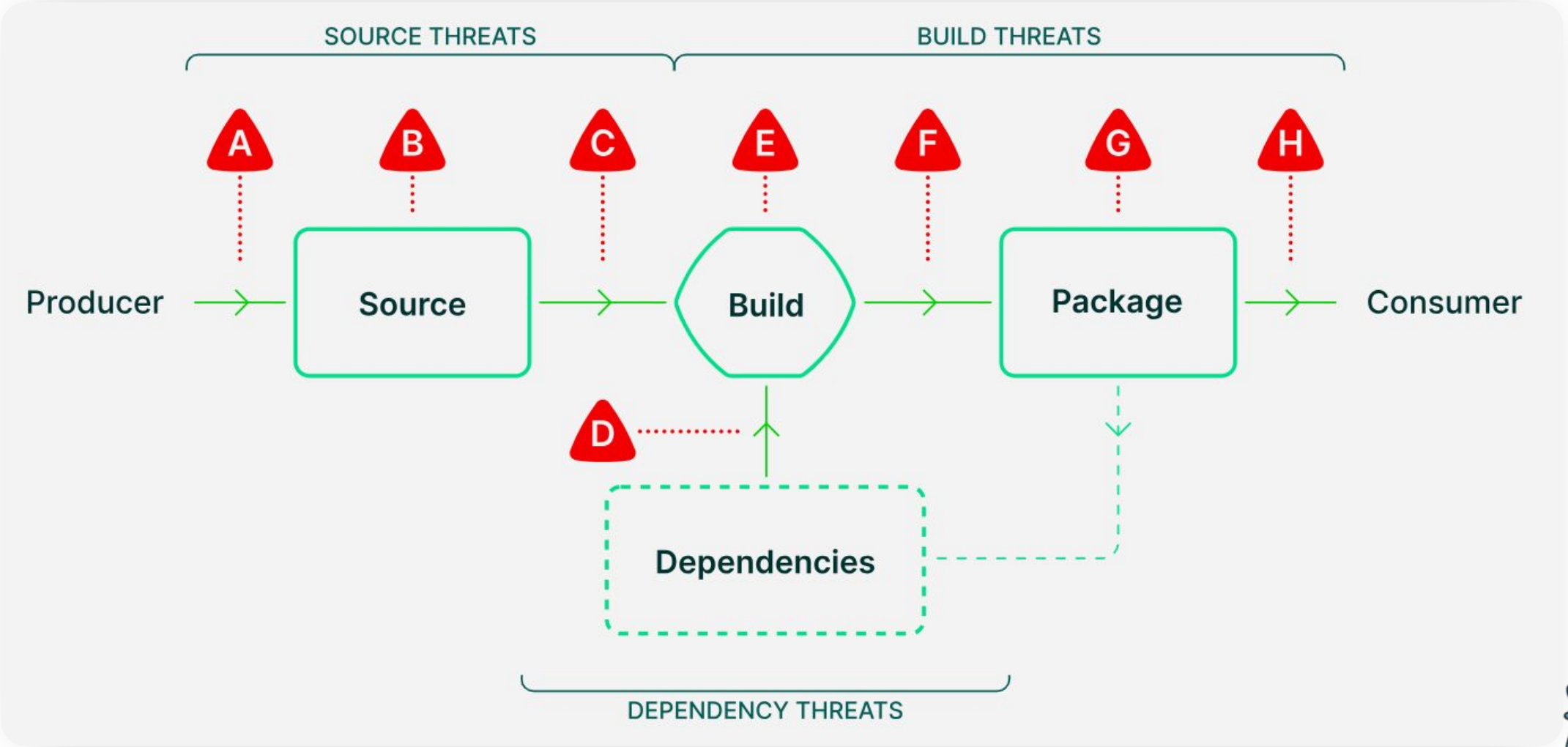


SLSA (Supply-chain Levels for Software Artifacts)

Интересный доклад по теме цепочек поставок:

SLSA (Supply-chain Levels for Software Artifacts)

SOLAR



SLSA (Supply-chain Levels for Software Artifacts)

> 50%

популярных библиотек имеют хотя бы один коммит, который был создан и влит только одним разработчиком

SLSA (Supply-chain Levels for Software Artifacts)

Provenance – информация о том, кто создал один или несколько программных артефактов и какие этапы и материалы использовались для создания этих артефактов. Формат in-toto (https://github.com/in-toto/attestation).

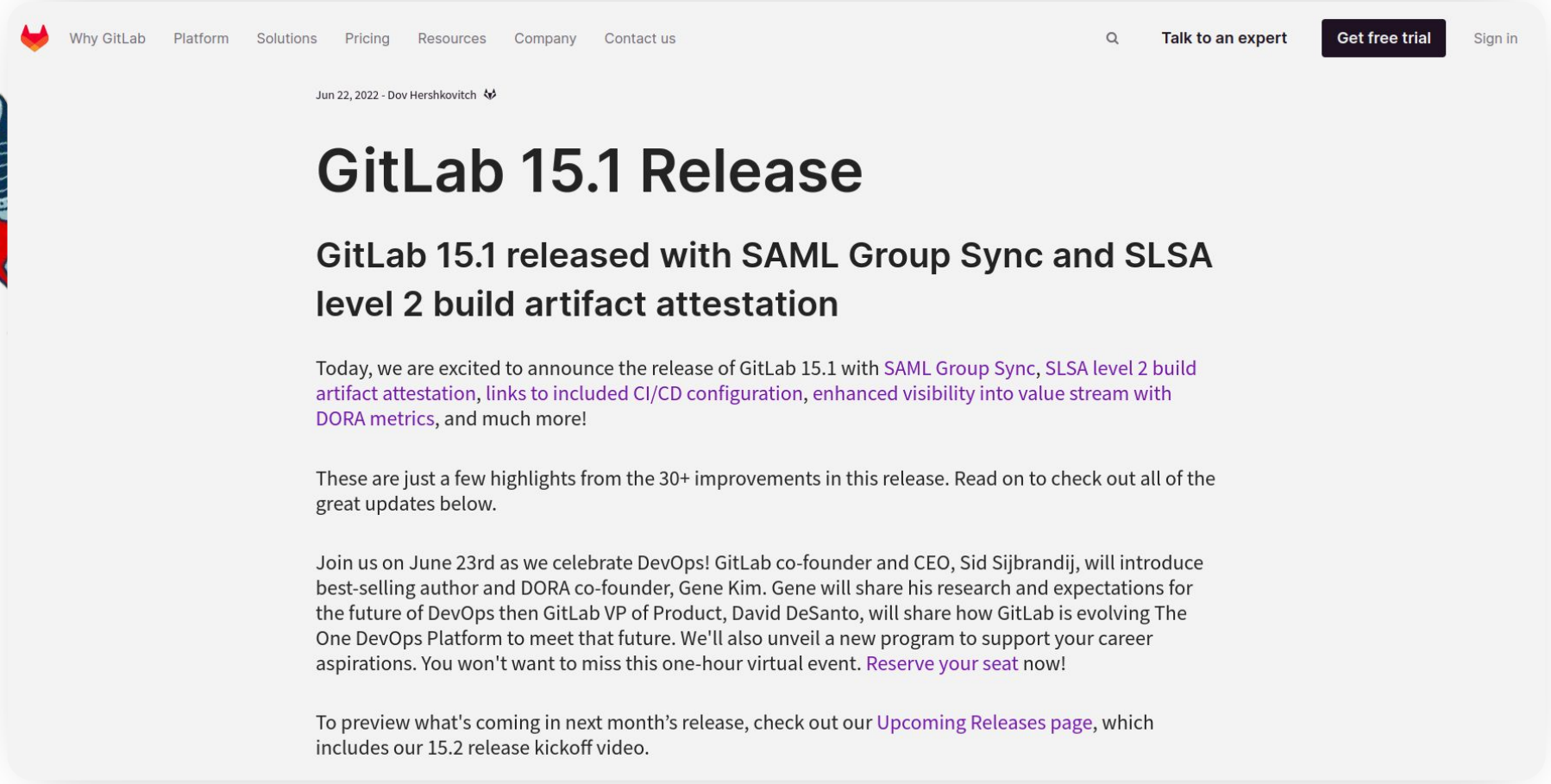
SLSA (Supply-chain Levels for Software Artifacts)

Provenance – информация о том, кто создал один или несколько программных артефактов и какие этапы и материалы использовались для создания этих артефактов. Формат in-toto (https://github.com/in-toto/attestation).


```
{
    "_type": "https://in-toto.io/Statement/v0.1",
    "subject": [
        {
            "name": "output.txt",
            "digest": {
                "sha256": "a948904f2f0f479b8f8197694b30184b0d2ed1c1cd2a1ec0fb85d299a192a447"
            }
        }
    ],
    "predicateType": "https://slsa.dev/provenance/v0.2",
    "predicate": {
        "buildtype": "https://www.jenkins.io/Pipeline",
        "builder": {
            "id": "http://jenkisDashboard/job/public_test/122/"
        },
        "invocation": {
            "configSource": {
                "uri": "git://github.com/Samsung/slsa-jenkins-generator.git@refs/heads/main",
```

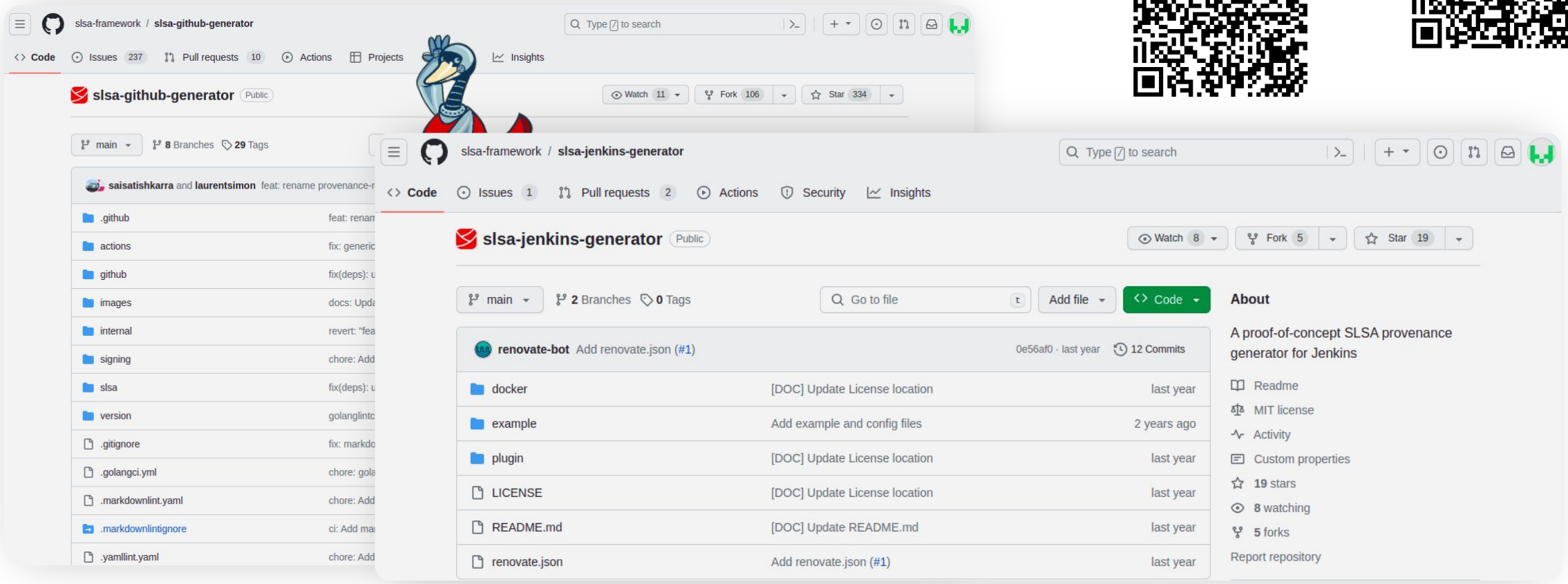
SLSA (Supply-chain Levels for Software Artifacts)

Gitlab платная версия - Опция RUNNER_GENERATE_ARTIFACTS_METADATA = "true" в вашем .gitlab-ci.yml файле



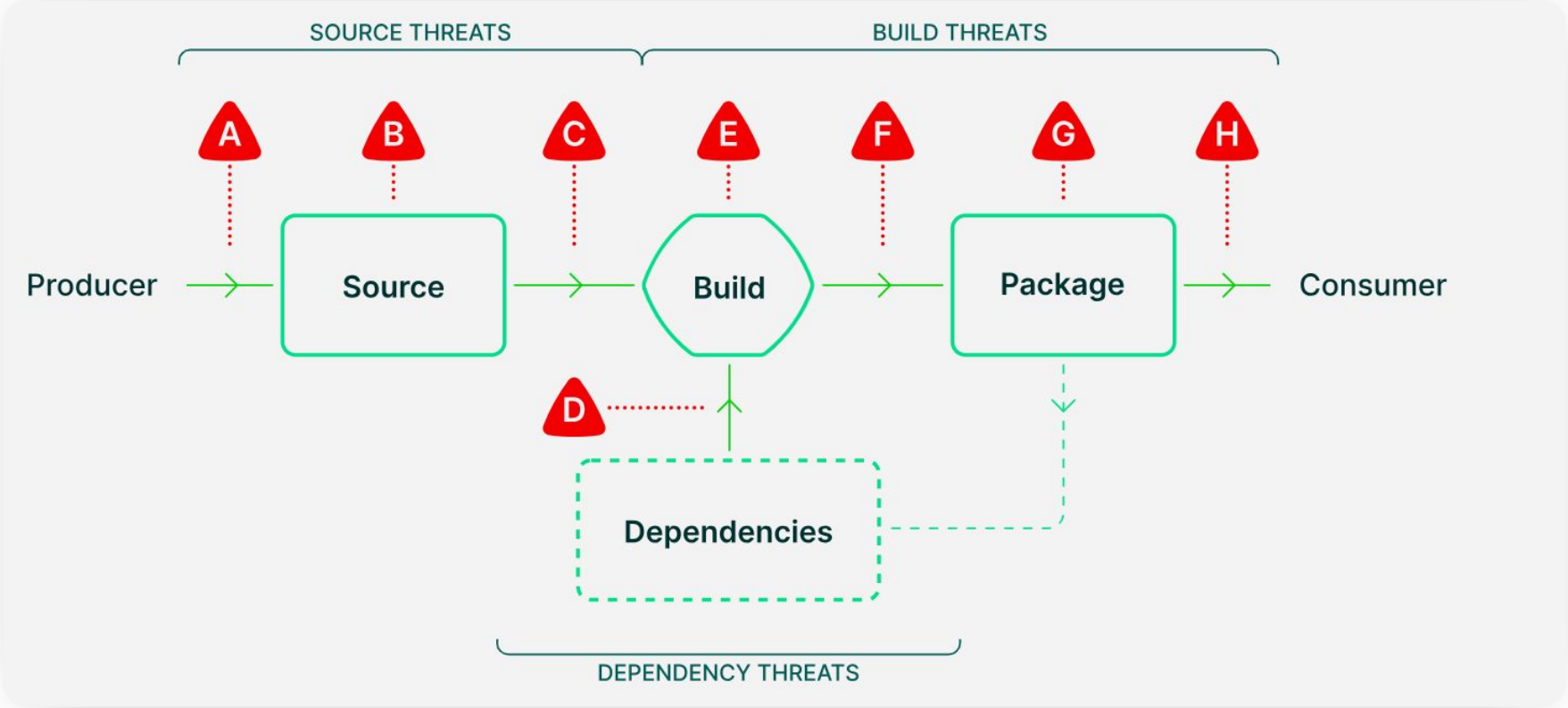
SLSA (Supply-chain Levels for Software Artifacts)

GitHub Actions с плагином - https://github.com/slsa-framework/slsa-github-generator
Для Jenkins - https://github.com/slsa-framework/slsa-jenkins-generator



SLSA (Supply-chain Levels for Software Artifacts)

Нужен в первую очередь для производителя ПО, чтобы можно было сказать: «А вот у меня SLSA3, и вот мой provenance». Это дает больше доверия для пользователя и заставляет производителя ПО относиться к выбору сборки более тщательно.



SLSA (Supply-chain Levels for Software Artifacts)

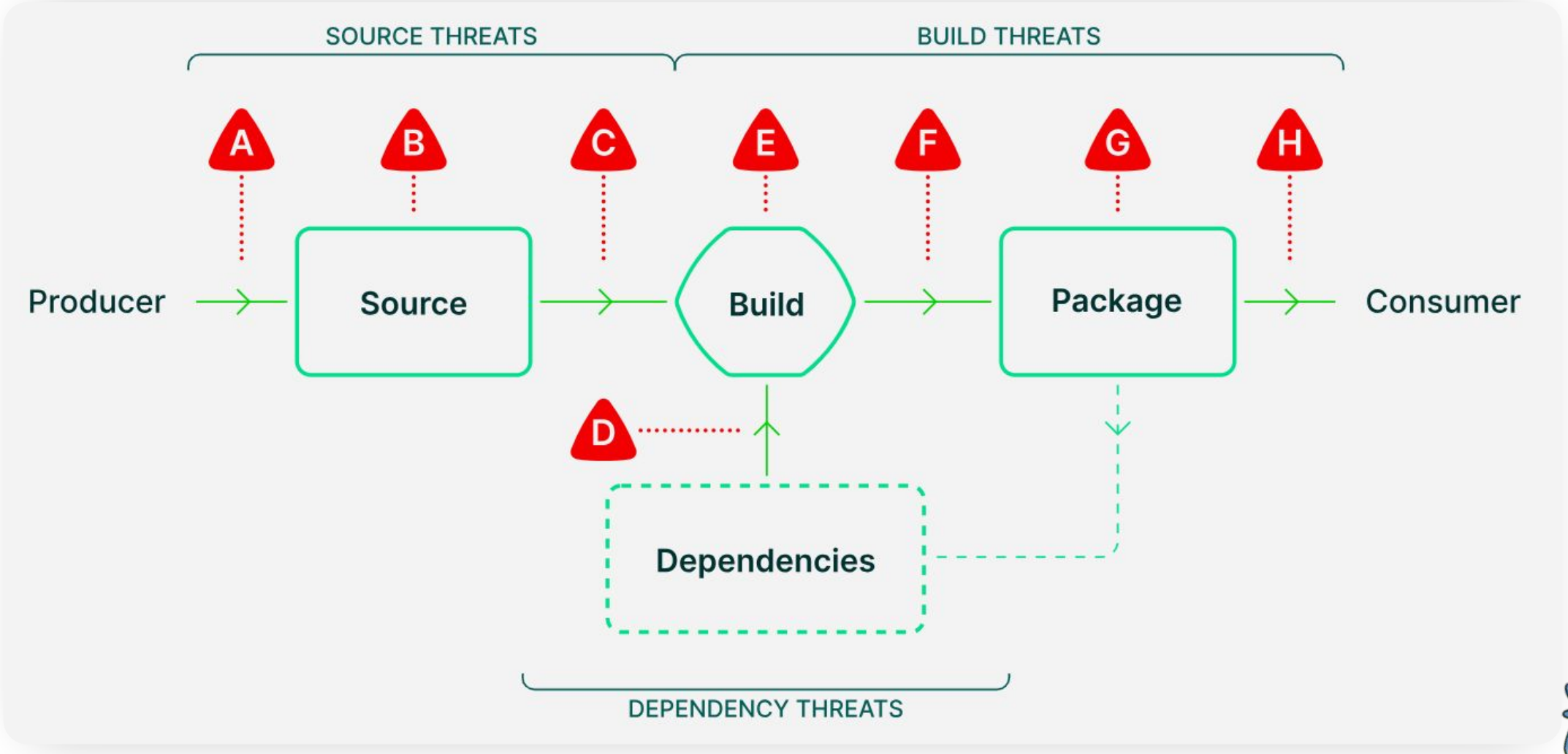
Kyverno прошел third-party аудит, проводимый компанией Ada Logics, по итогу которого были также оценены риски цепочки поставок в соответствии с SLSA.

target one attack surface identified during the threat modelling: Policy bypasses, i.e. where an internal attacker attempts to submit a request that bypasses a policy deployed by the Kyverno admin.

The SLSA review found that Kyverno complies at the highest level (SLSA Level 3). Kyverno builds its releases on GitHub Actions and includes verifiable provenance with releases, which makes Kyverno hardened against a series of well-known attack vectors in Kyverno's software supply-chain.

3 Ada Logics Ltd

SLSA (Supply-chain Levels for Software Artifacts)



OWASP Software Component Verification Standard (SCVS)

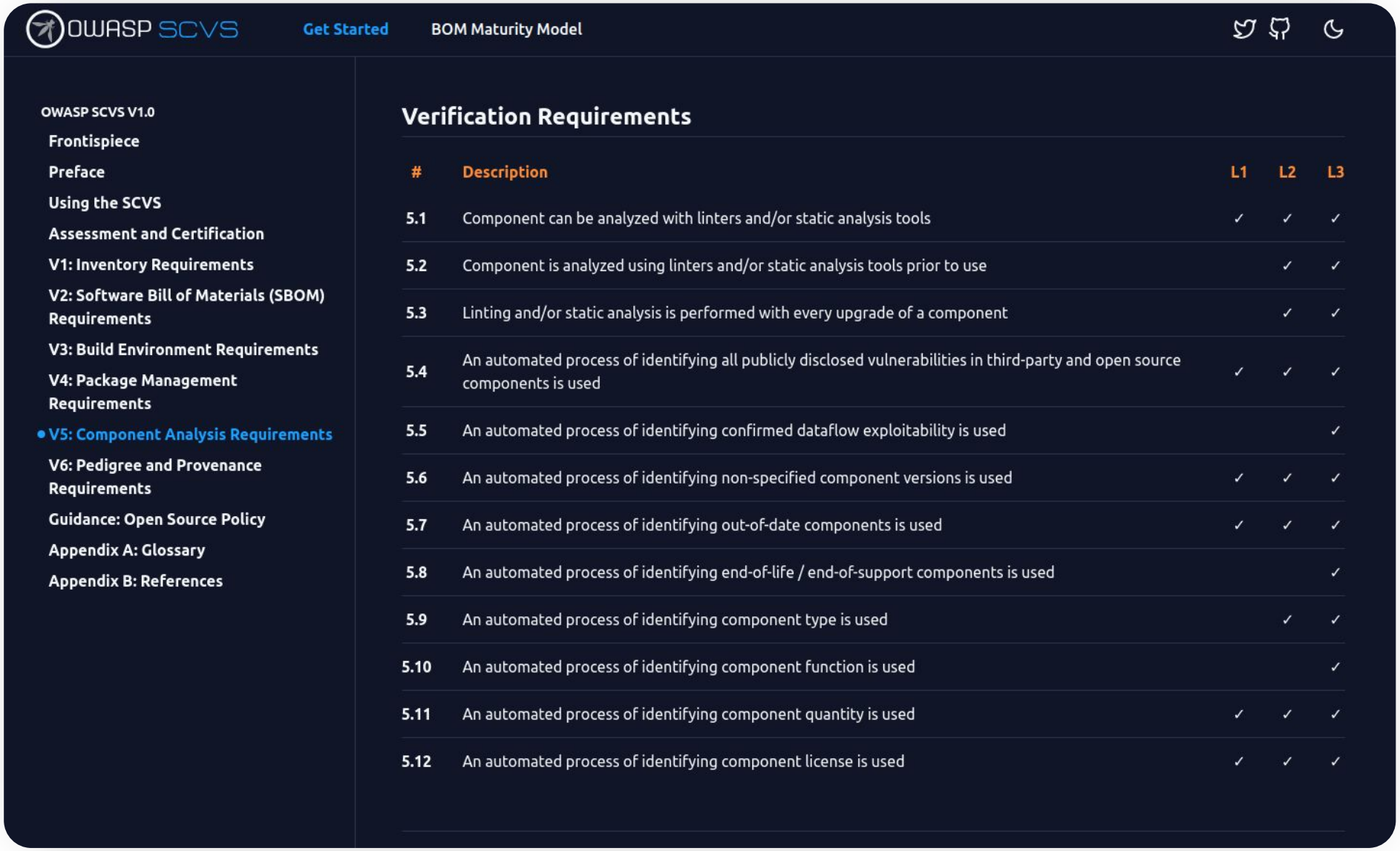
Стандарт OWASP SCVS:

- направлен на улучшение безопасности и качества цепочек поставок ПО
- сформулирован достаточно кратко
- состоит из **шести контролей**

OWASP Software Component Verification Standard (SCVS)

- V1 Необходимость инвентаризации
- V2 Требования к содержимому спецификации программного обеспечения (SBOM)
- V3 Требования к правильной настройке среды сборки
- V4 Требования к управлению пакетами
- V5 Анализ компонентов на уязвимости, качество и наличие опасных лицензий
- V6 Сбор информации о происхождении пакетов и их модификаций

OWASP Software Component Verification Standard (SCVS)

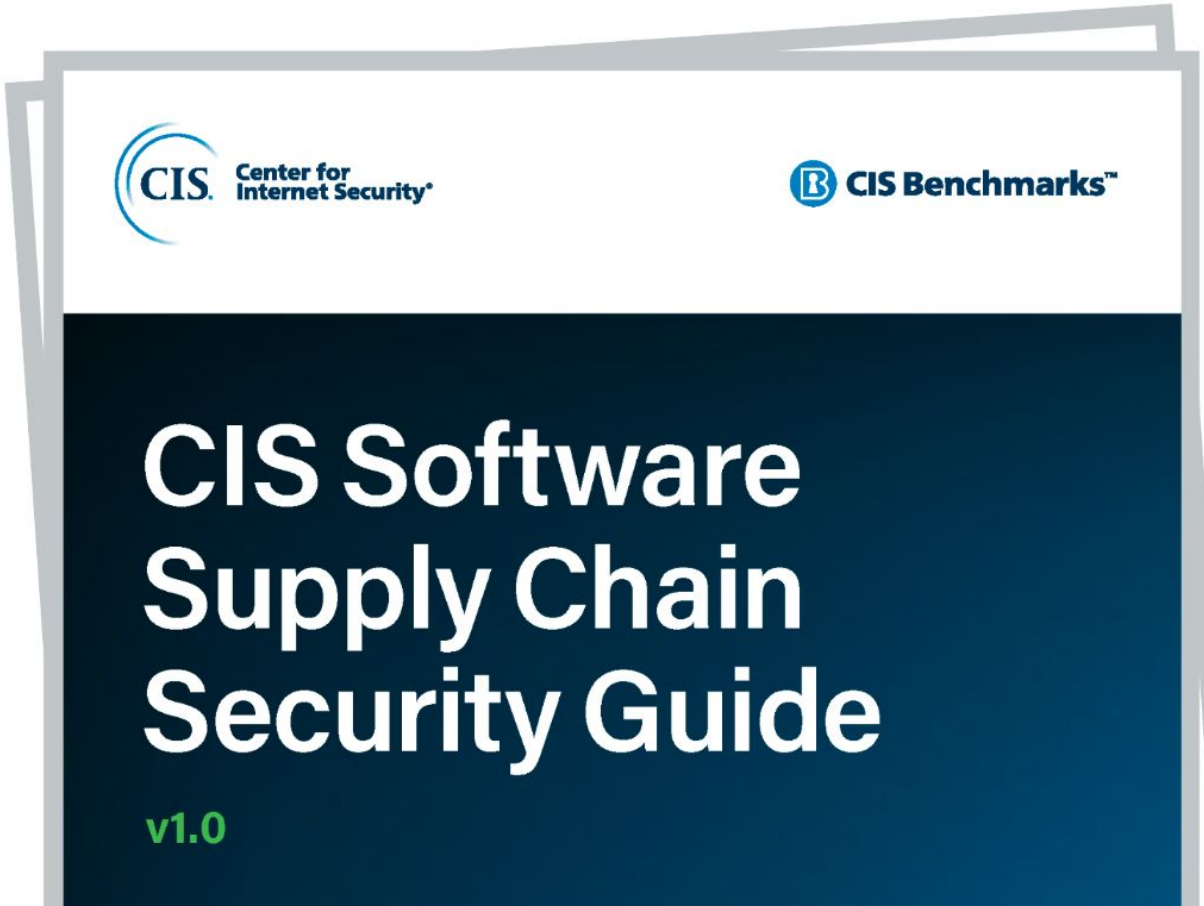


#	Description	L1	L2	L3
5.1	Component can be analyzed with linters and/or static analysis tools	✓	✓	✓
5.2	Component is analyzed using linters and/or static analysis tools prior to use		✓	✓
5.3	Linting and/or static analysis is performed with every upgrade of a component		✓	✓
5.4	An automated process of identifying all publicly disclosed vulnerabilities in third-party and open source components is used	✓	✓	✓
5.5	An automated process of identifying confirmed dataflow exploitability is used			✓
5.6	An automated process of identifying non-specified component versions is used	✓	✓	✓
5.7	An automated process of identifying out-of-date components is used	✓	✓	✓
5.8	An automated process of identifying end-of-life / end-of-support components is used			✓
5.9	An automated process of identifying component type is used		✓	✓
5.10	An automated process of identifying component function is used			✓
5.11	An automated process of identifying component quantity is used	✓	✓	✓
5.12	An automated process of identifying component license is used	✓	✓	✓

CIS Software Supply Chain Security Guide (SSCSG)

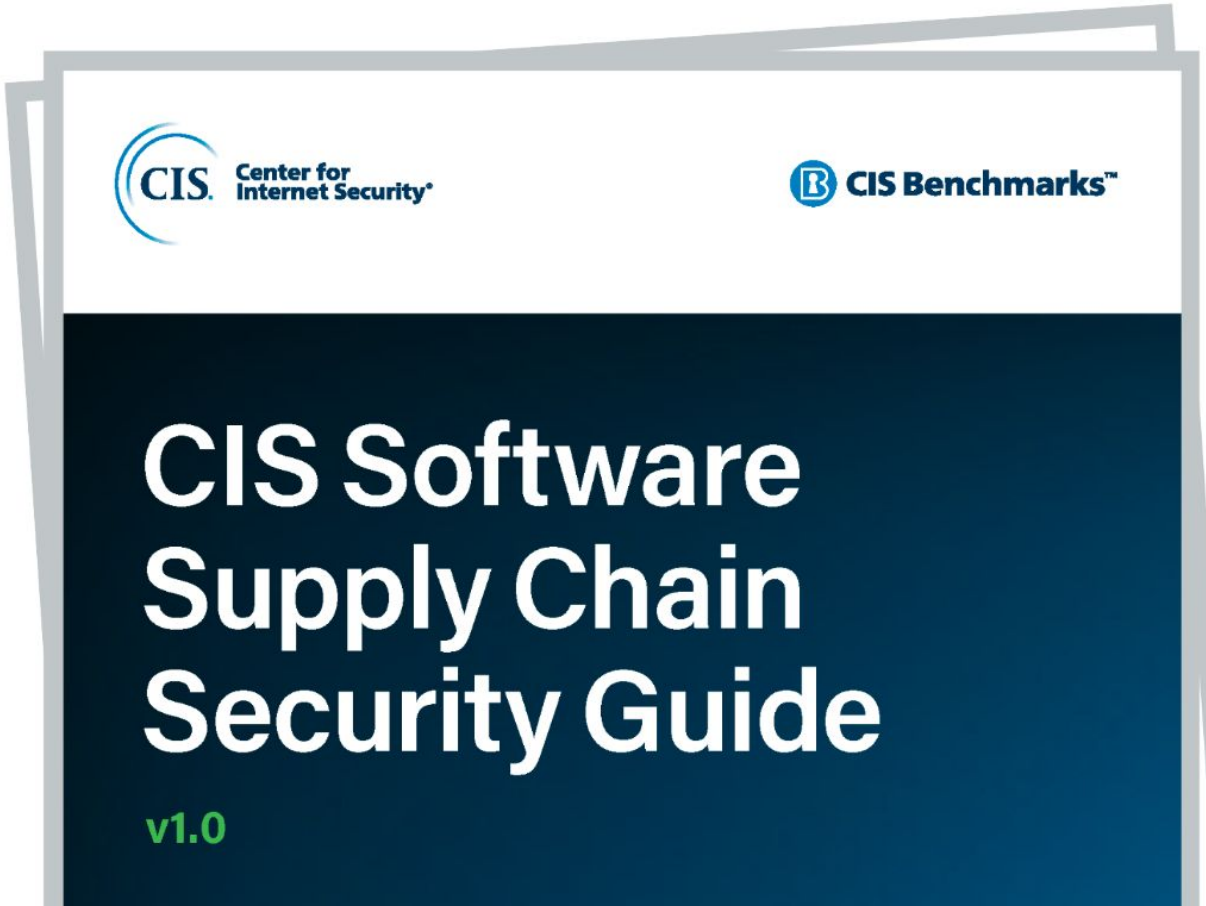
Гайд описывает этапы цепочки поставок программного обеспечения, начиная с добавления кода разработчиком и заканчивая доставкой ПО клиенту.

Руководство согласуется со стандартами безопасности, такими как SLSA, и включает более **100 рекомендаций в пяти основных категориях**.



CIS Software Supply Chain Security Guide (SSCSG)

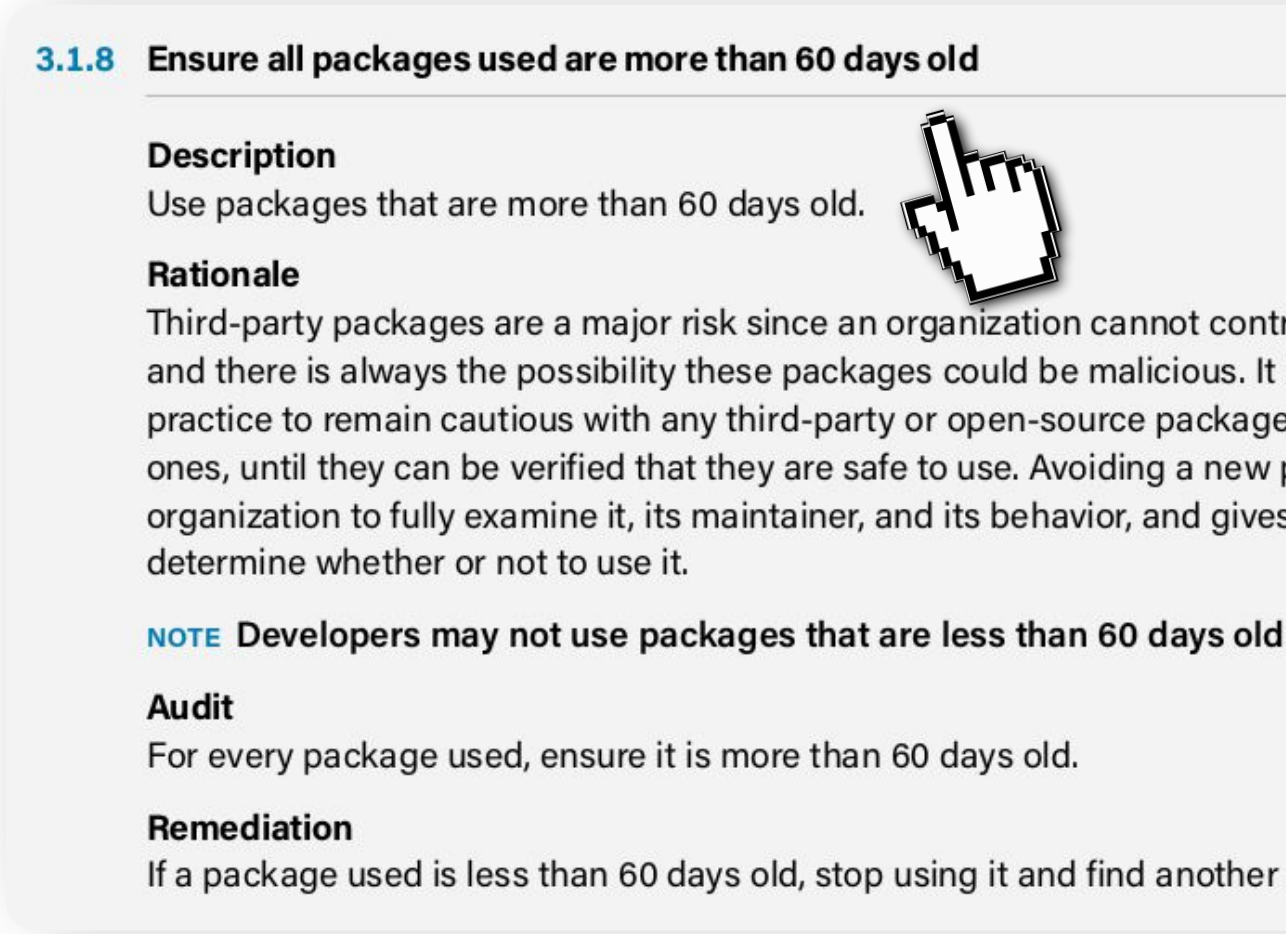
1. **Source Code:** управление исходным кодом приложения
2. **Build Pipelines:** безопасность сборки
3. **Dependencies:** управление зависимостями
4. **Artifacts:** управление артефактами сборки
5. **Deployment:** защита процесса развертывания приложения и связанных файлов и конфигураций



CIS Software Supply Chain Security Guide (SSCSG)

ПРИМЕР

Согласно пункту **3.1.8 CIS Software Supply Chain Security Guidelines**, возраст пакета должен быть больше 60 дней: эта мера предосторожности поможет избежать внедрения потенциально вредоносных сторонних пакетов и обеспечить достаточное время на их проверку.



3.1.8 Ensure all packages used are more than 60 days old

Description
Use packages that are more than 60 days old.

Rationale
Third-party packages are a major risk since an organization cannot contr
and there is always the possibility these packages could be malicious. It
practice to remain cautious with any third-party or open-source package
ones, until they can be verified that they are safe to use. Avoiding a new p
organization to fully examine it, its maintainer, and its behavior, and gives
determine whether or not to use it.

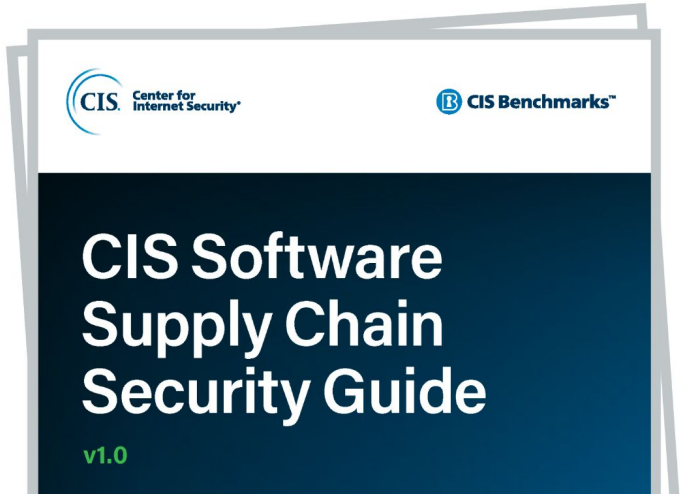
NOTE **Developers may not use packages that are less than 60 days old**

Audit
For every package used, ensure it is more than 60 days old.

Remediation
If a package used is less than 60 days old, stop using it and find another

Обзор стандартов

1. SLSA (Supply-chain Levels for Software Artifacts)
2. OWASP Software Component Verification Standard (SCVS)
3. CIS Software Supply Chain Security Guide (SSCSG)



САМОЕ УЯЗВИМОЕ ЗВЕНО ЦЕПОЧКИ ПОСТАВОК – ЭТО

...

САМОЕ УЯЗВИМОЕ ЗВЕНО ЦЕПОЧКИ ПОСТАВОК – ЭТО ЗАВИСИМОСТИ

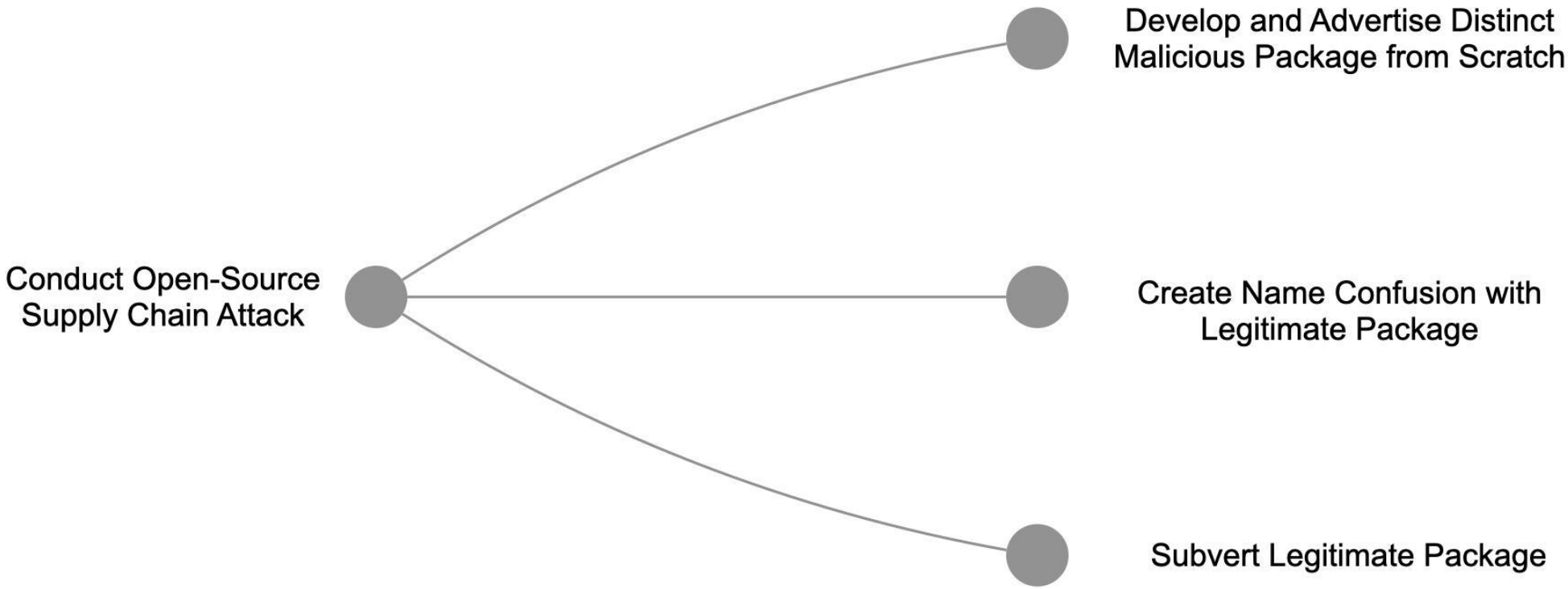
Вспомним

“You can't trust code that you did not totally create yourself. Especially code from companies that employ people like me”.

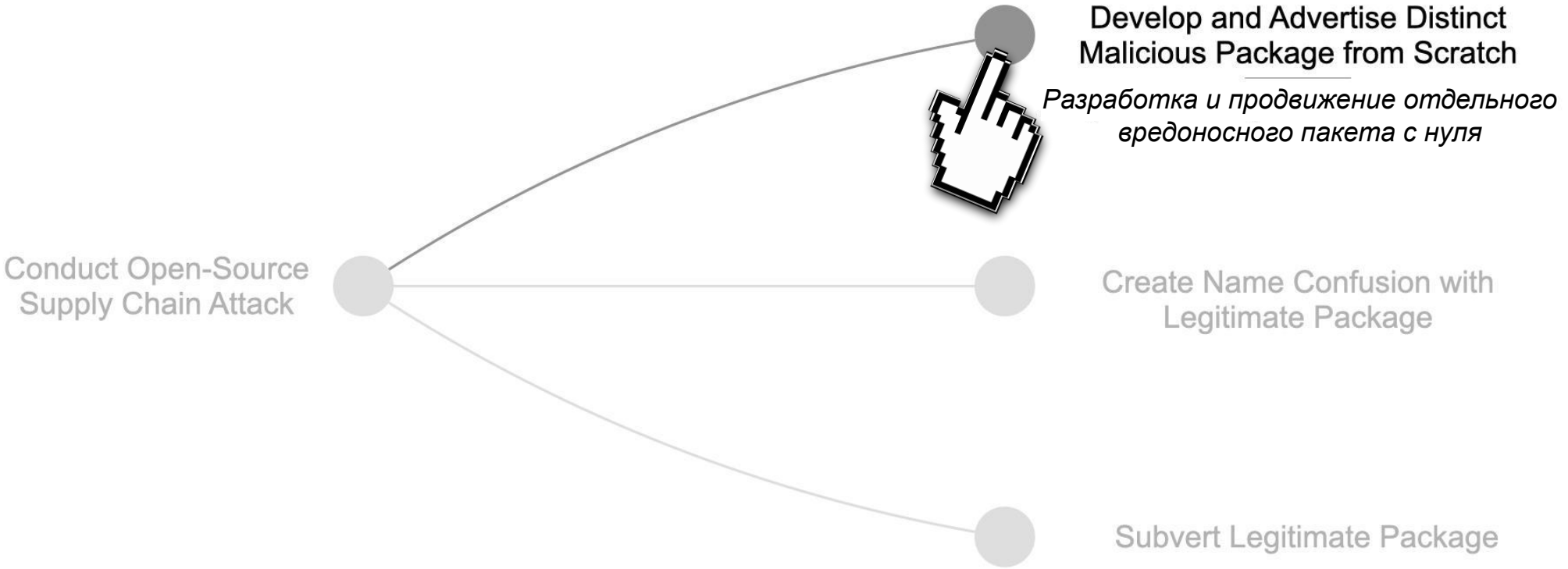
Вы можете доверять коду, только если вы написали его сами. Особенно нельзя доверять коду от компаний, которые берут на работу таких как я”.

Ken Thompson, 1984
“Reflections on Trusting Trust”

Разбор атак на цепочку поставок (Supply Chain)



Разбор атак на цепочку поставок (Supply Chain)



Разбор атак на цепочку поставок (Supply Chain)

Разработчик RIAEvangelist

Разбор атак на цепочку поставок (Supply Chain)

Разработчик RIAEvangelist

dev.by
https://devby.io › Новости

Популярный npm-пакет удаляет и перезаписывает ...

18 мар. 2022 г. — Однако теперь обнаружилось, что некоторые версии известной библиотеки node-ipc, тоже поддерживаемой **RIAEvangelist**, содержат гораздо более ...

SC Media
https://www.scmagazine.com › ...

What happens when 'protestware' sabotages open source ...

17 мар. 2022 г. — As **RIAEvangelist** updated Node-ipc, he updated the version numbers as well, triggering automatic updating of code for many downstream users. " ...

PortSwigger
https://portswigger.net › npm...

NPM maintainer targets Russian users with data-wiping ' ...

21 мар. 2022 г. — '**RIAEvangelist**' (aka Brandon Nozaki Miller) embedded malware – or 'protestware', as he dubbed it – into Node.JS module node-ipc's latest ...

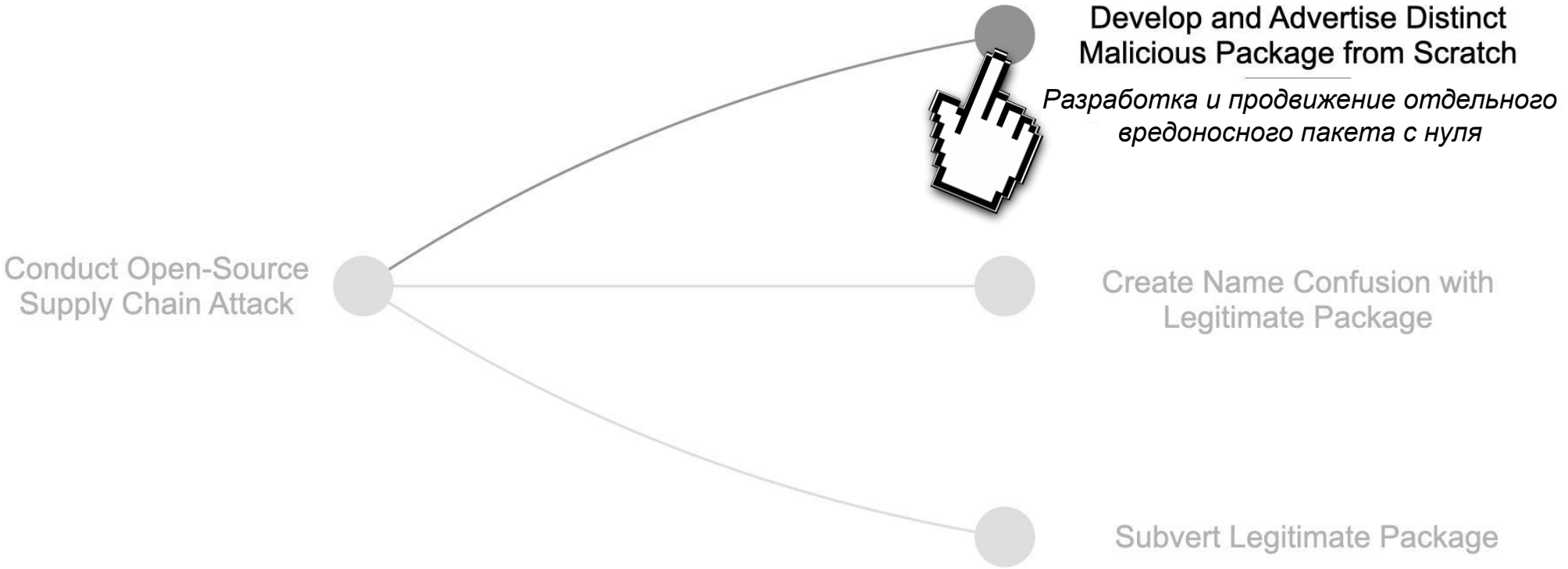
Student Pocket Guide
https://www.thestudentpocketguide.com › ...

RIAEvangelist's anti-war "protestware" bashed by FOSS ...

Hacktivist **RIAEvangelist** sparked outrage beyond the FOSS (Free and Open Source Software) community after uploading “protestware” (read malware) to his own ...

Подобных хактивистов можно отслеживать в репозитории
https://github.com/toxic-repos/toxic-repos

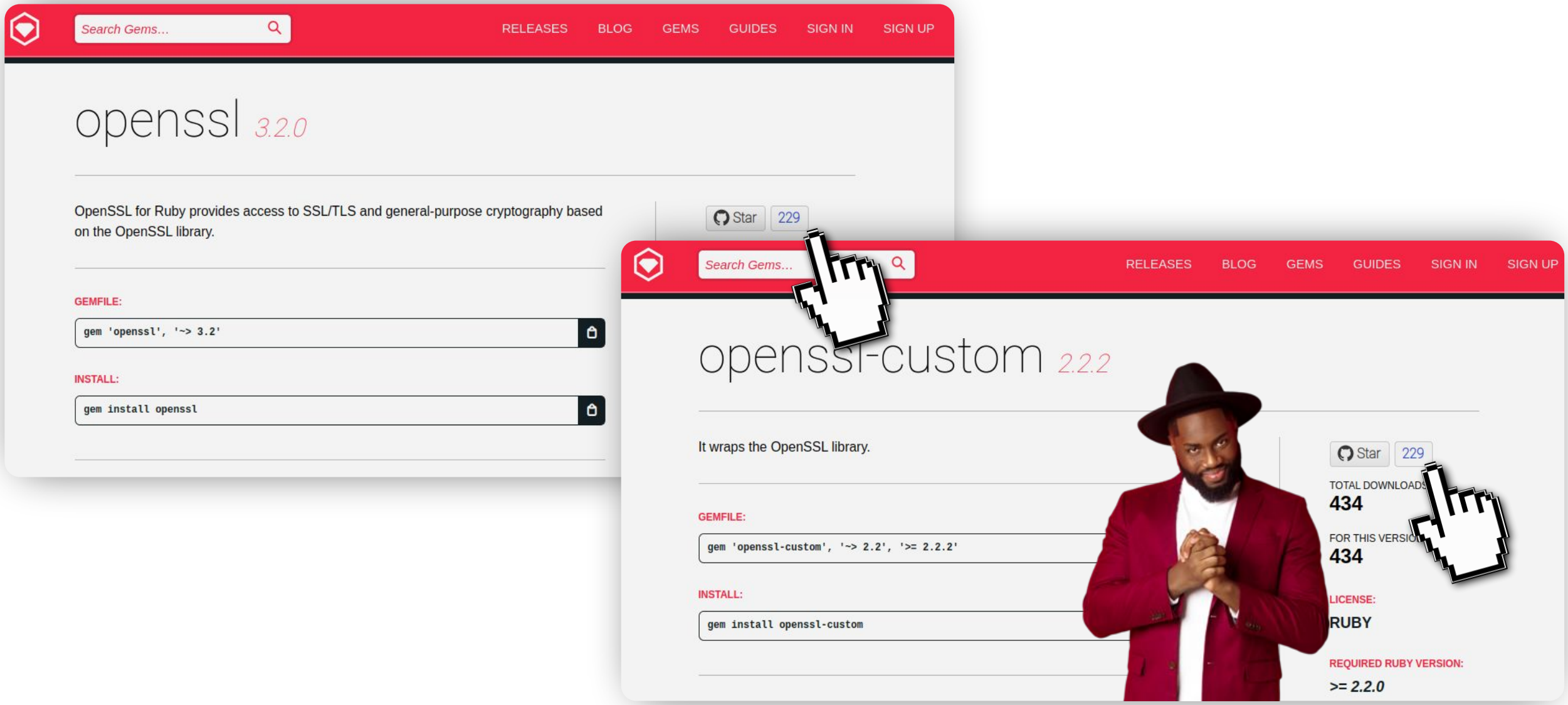
Разбор атак на цепочку поставок (Supply Chain)



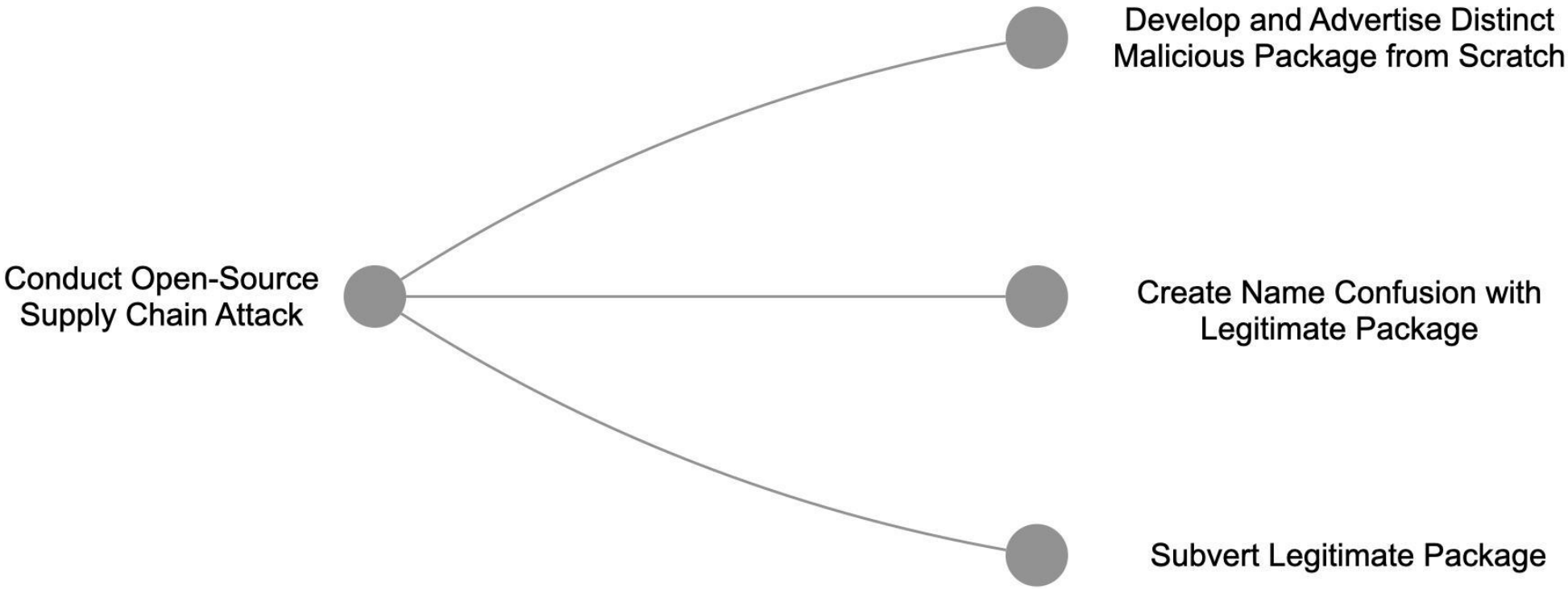
Разбор атак на цепочку поставок (Supply Chain)

Starjacking - техника актуальна для пакетных менеджеров NPM/Yarn, PyPi и RubyGems. Заключается в краже популярности («звездочек») у чужого известного пакета.

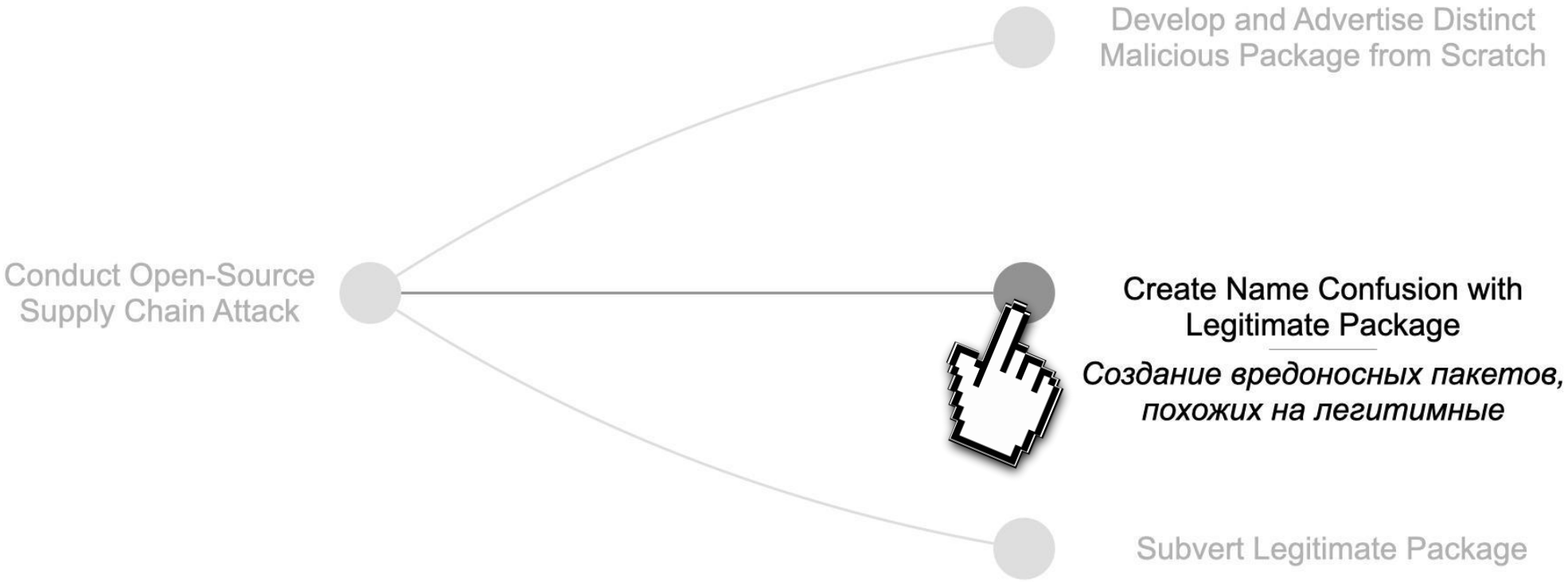
Разбор атак на цепочку поставок (Supply Chain)



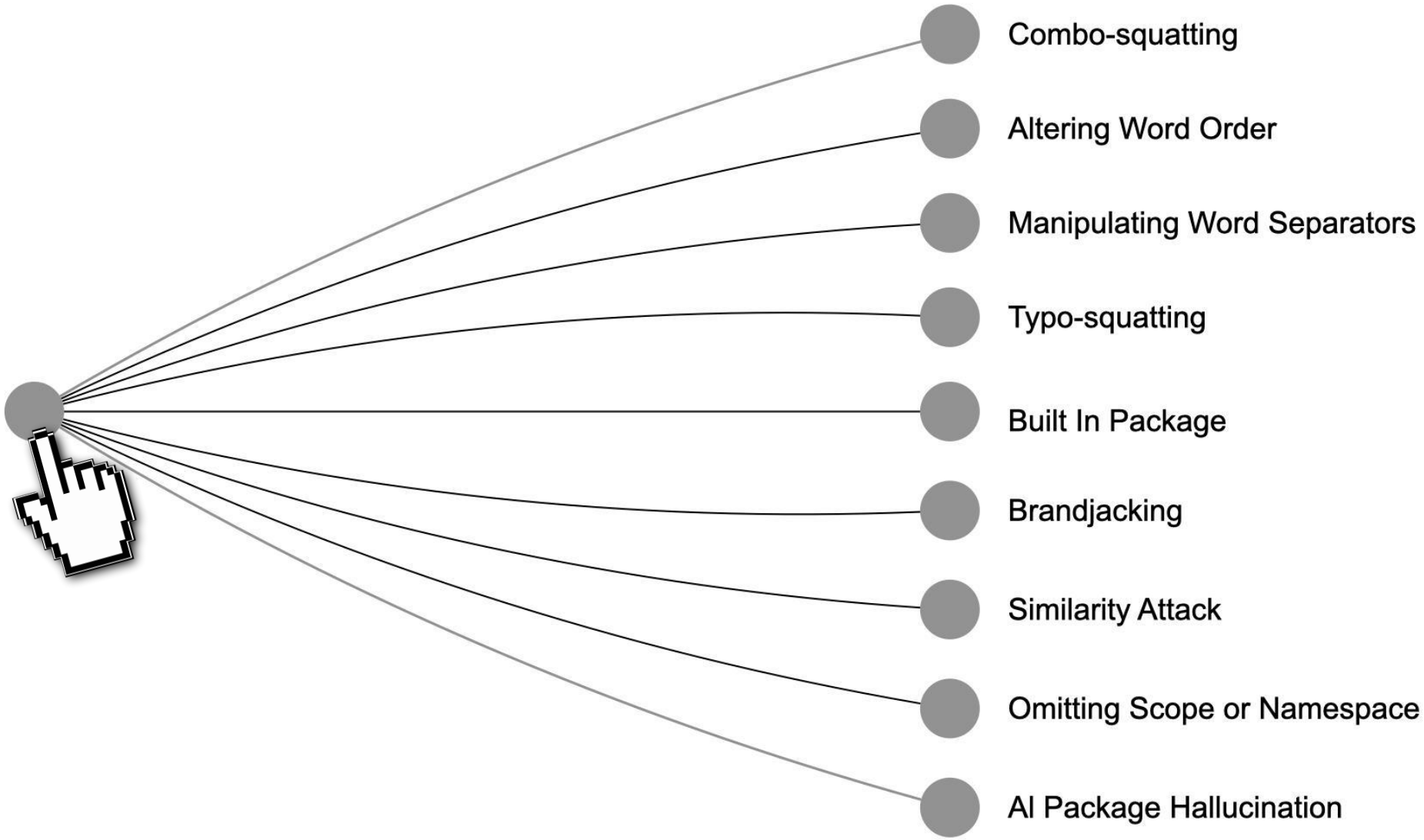
Разбор атак на цепочку поставок (Supply Chain)



Разбор атак на цепочку поставок (Supply Chain)



Разбор атак на цепочку поставок (Supply Chain)



Разбор атак на цепочку поставок (Supply Chain)

colors DT

1.4.0 • Public • Published 4 years ago

Code Beta | 0 Dependencies | 21 436 Dependents | 26 Versions

colors.js

build unknown | npm v1.4.0 | dependencies | devDependencies

Please check out the roadmap for upcoming features and releases. Please open Issues to provide feedback, and check the `develop` branch for the latest bleeding-edge upd

get color and style in your node.js console

```
→ node examples/normal-usage.js
First some yellow text
Underline that text
Make it bold and red
Double Raindows All Day Long
dŘØP ŦĤË ȚΔ§ϟ
dR◉P ŦĤě ЯΛ¡ ŋßØω βΛϟȣ
Chains are also cool.
So are inverse styles!
```

Install

```
> npm i colors
```

colorsss

0.0.2 • Public • Published 5 days ago

Explore BETA | 7 Dependencies | 0 Dependents | 2 Versions

This package does not have a README. Add a README to your package so t rs know how to get started.

Install

```
> npm i colorsss
```

Keywords

ansi terminal colors

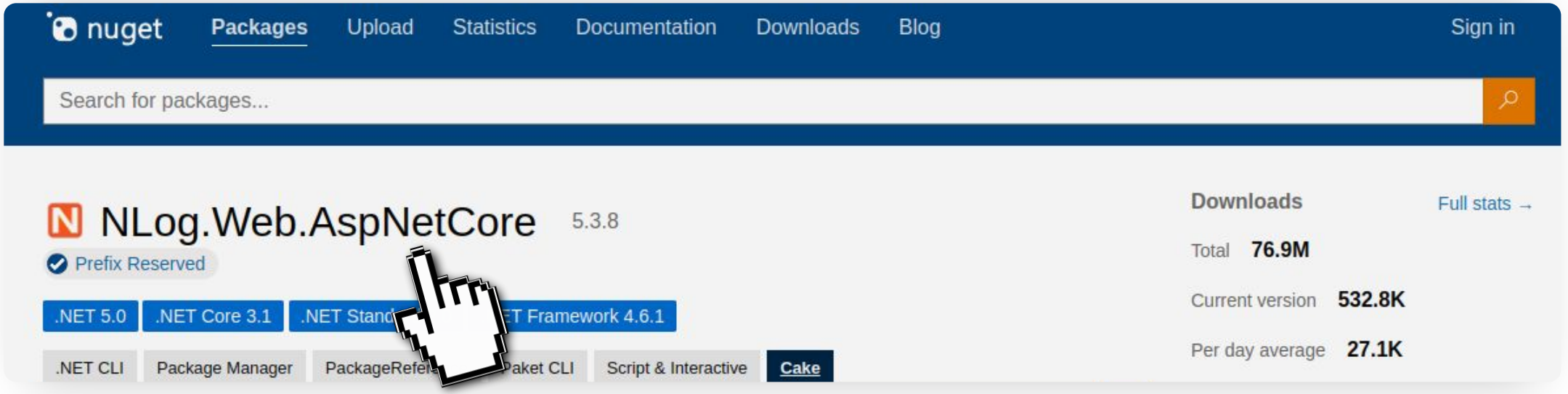
Weekly Downloads

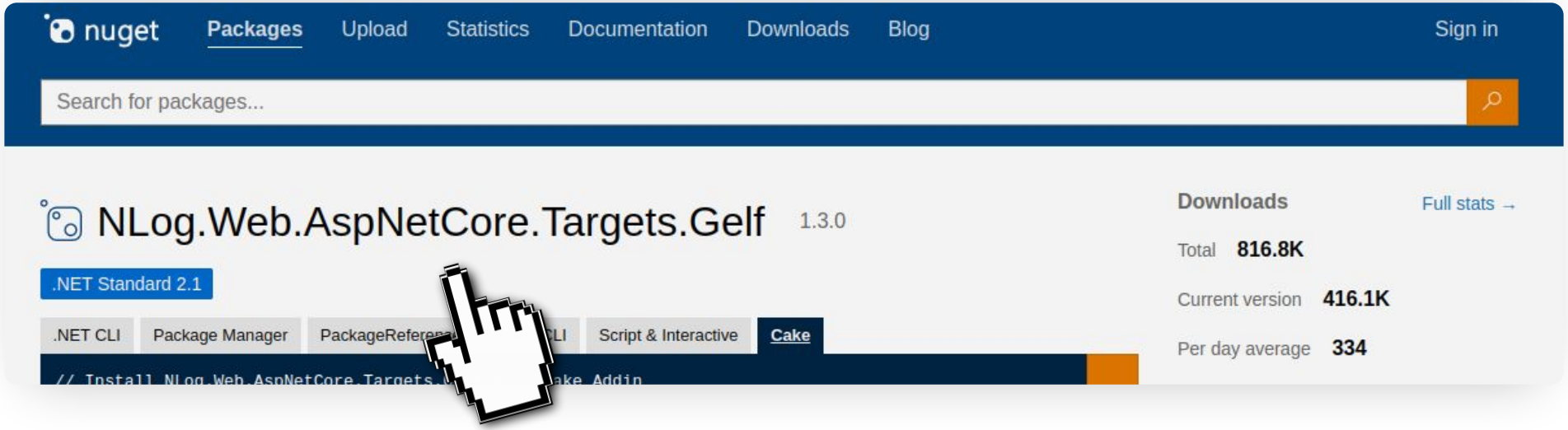
100

Version	License
0.0.2	MIT

Unpacked Size	Total Files
50.9 kB	20

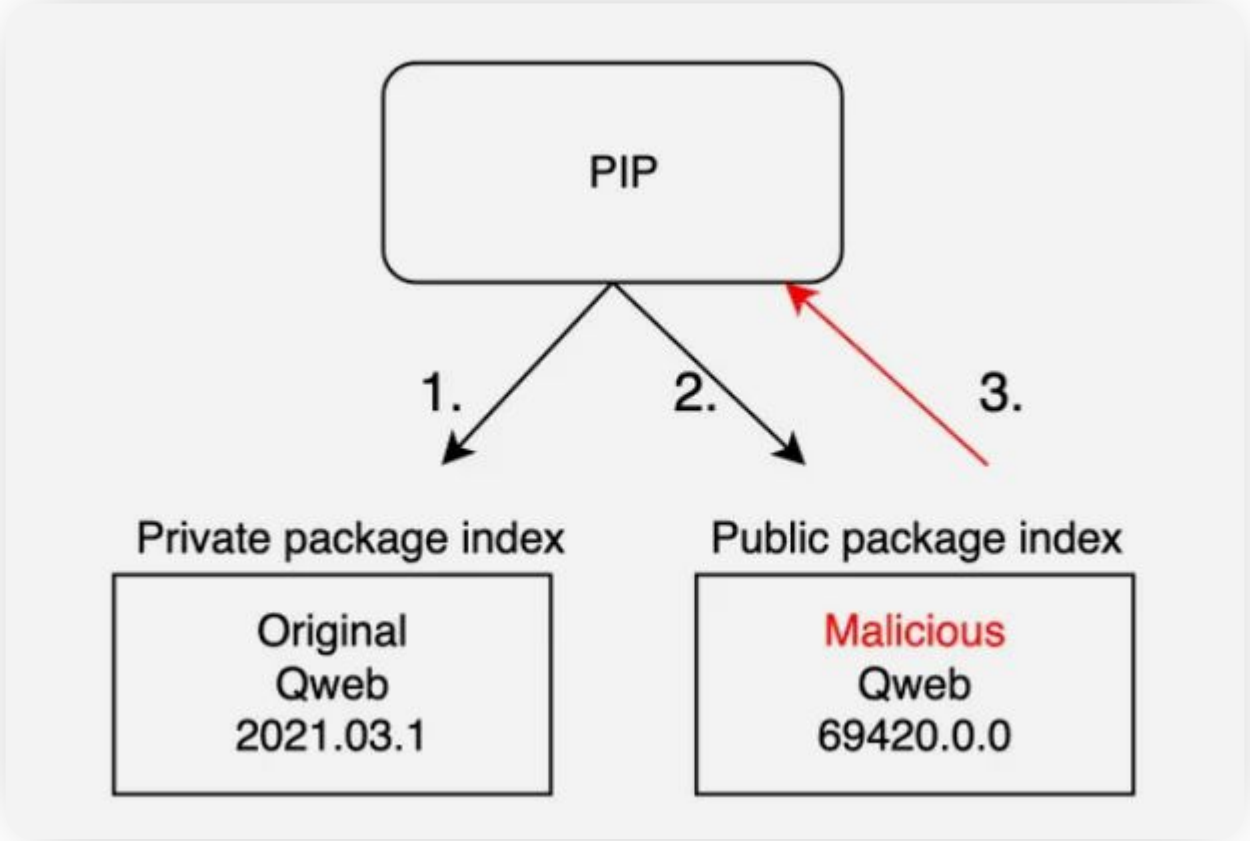
Разбор атак на цепочку поставок (Supply Chain)





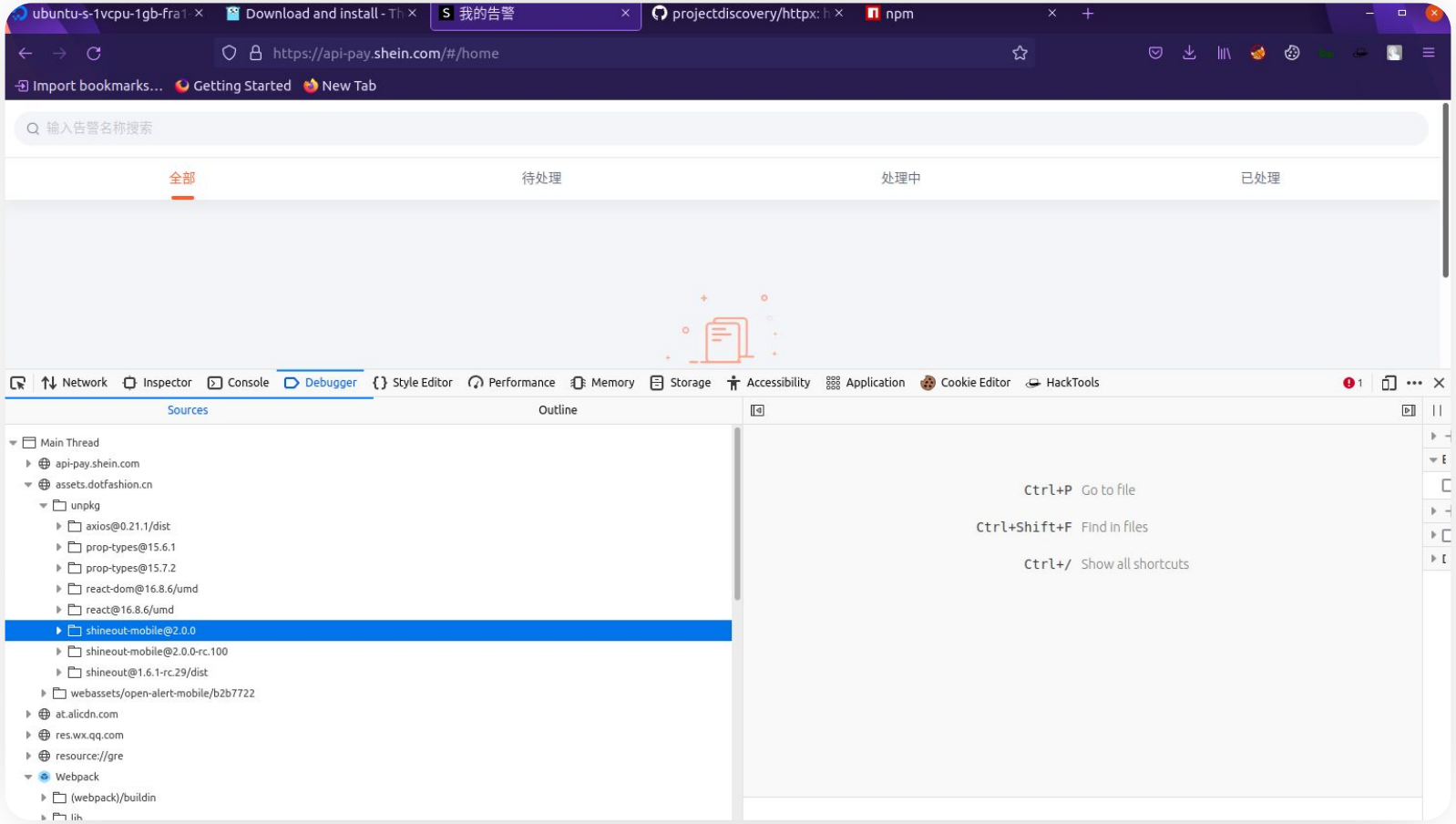
Разбор атак на цепочку поставок (Supply Chain)

Dependency Confusion - запутывание менеджера пакета во время скачивания локальных зависимостей. Для этого злоумышленники могут загружать в общедоступный репозиторий вредоносный пакет с тем же именем, что и внутренний.



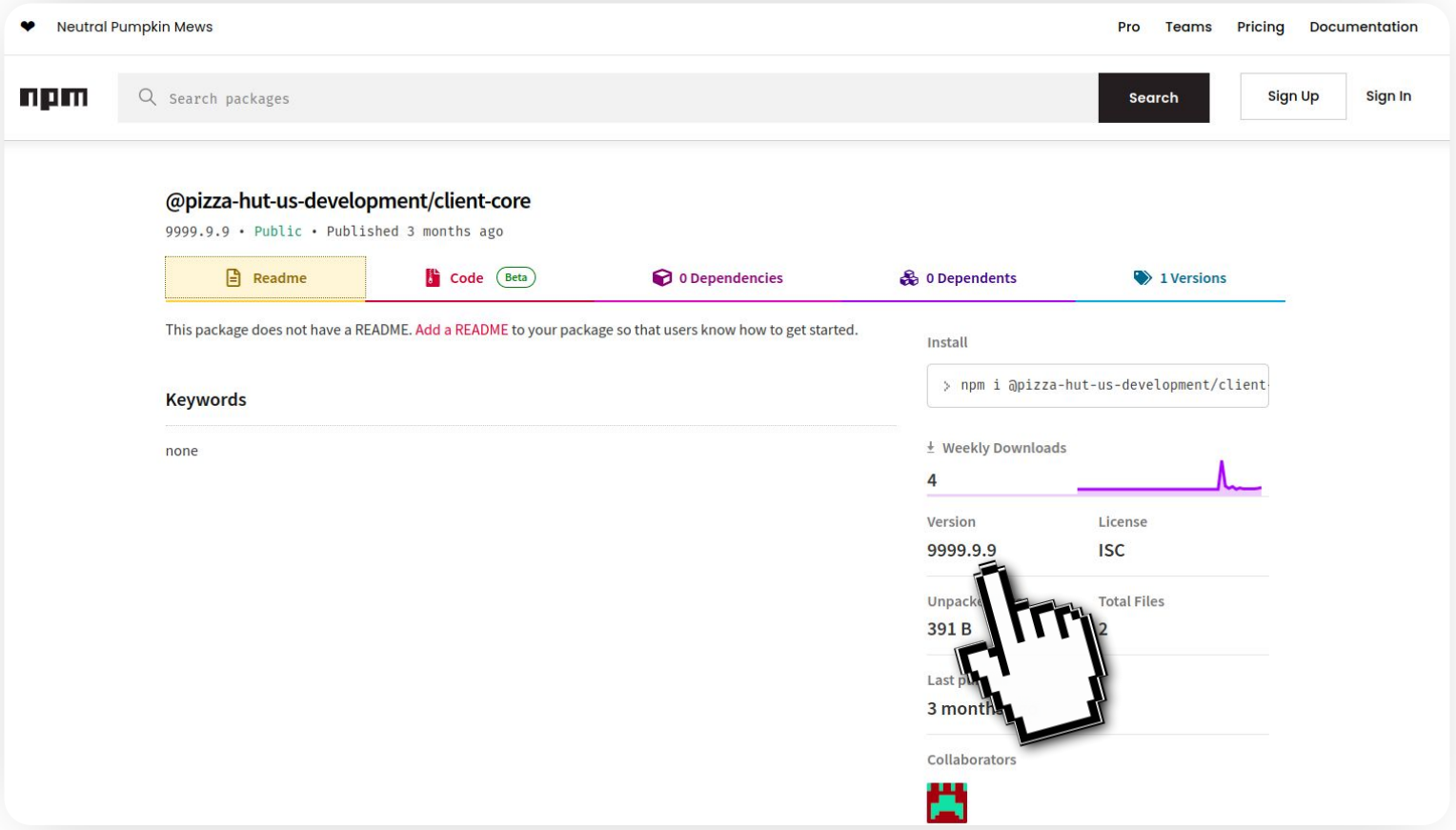
Разбор атак на цепочку поставок (Supply Chain)

Dependency Confusion - запутывание менеджера пакета во время скачивания локальных зависимостей. Для этого злоумышленники могут загружать в общедоступный репозиторий вредоносный пакет с тем же именем, что и внутренний.



Разбор атак на цепочку поставок (Supply Chain)

Dependency Confusion - запутывание менеджера пакета во время скачивания локальных зависимостей. Для этого злоумышленники могут загружать в общедоступный репозиторий вредоносный пакет с тем же именем, что и внутренний.



Как избежать Dependency Confusion?

Проверить, нет ли уже в общедоступном пакетном менеджере ваших библиотек

Примеры для NPM: https://www.npmjs.com/package/YOUR-PACKAGE-NAME
https://npmjs.com/org/SCOPE-NAMES

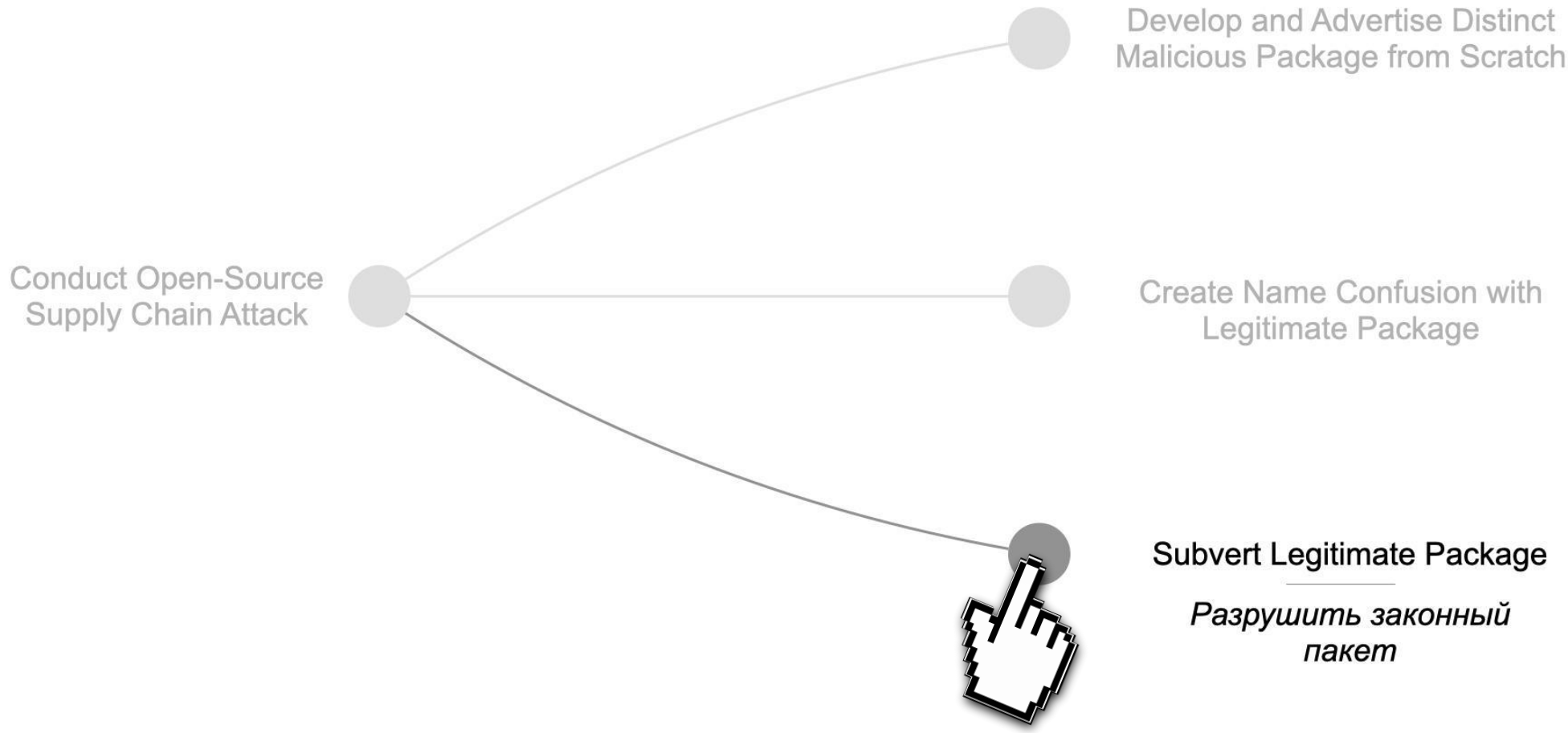
Использовать возможности файлов блокировки для обеспечения безопасного управления зависимостями

Пример: package-lock.json для NPM или Gemfile.lock для Ruby

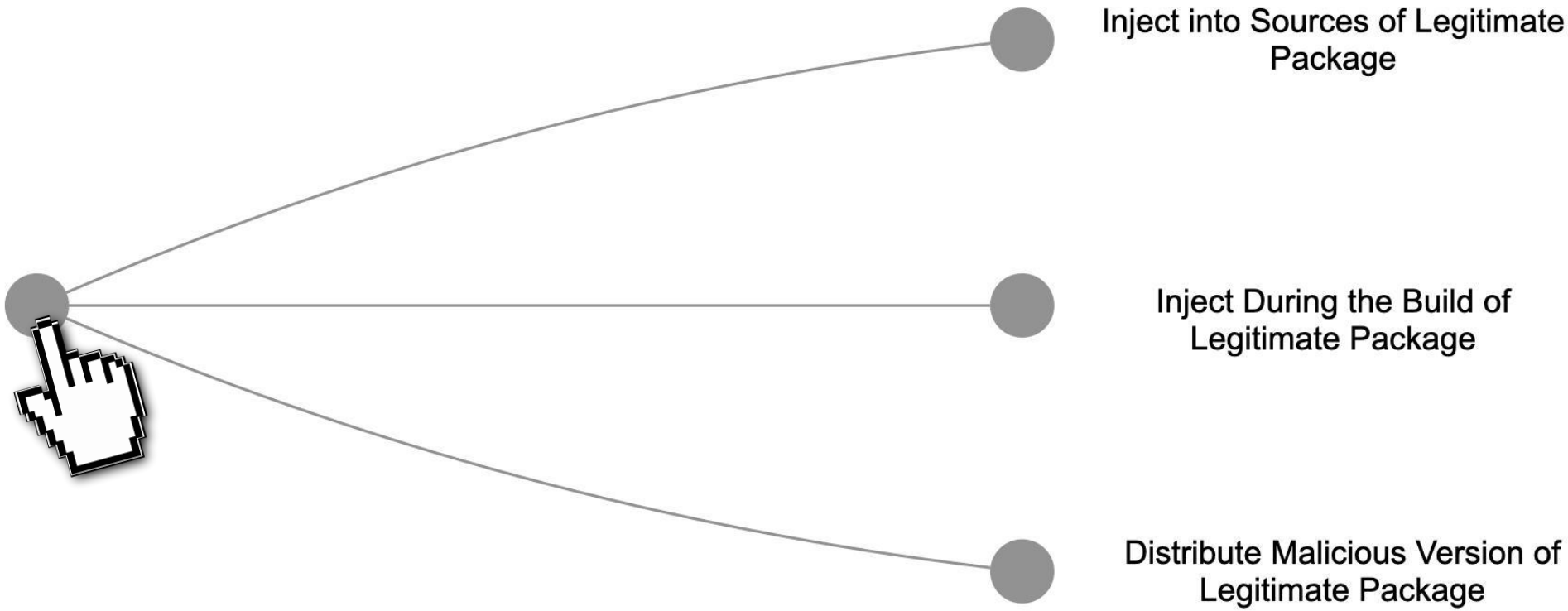
Использовать следующие инструменты:

- Утилита https://github.com/visma-prodsec/confused проверяет, заняты ли имена приватных пакетов в публичных репозиториях
- Утилита https://github.com/sonatype-nexus-community/repo-diff проверяет, нет ли в проксируемых репозиториях Nexus пакетов с такими же именами, что и в приватных

Разбор атак на цепочку поставок (Supply Chain)



Разбор атак на цепочку поставок (Supply Chain)



Разбор атак на цепочку поставок (Supply Chain)

[AV-800] Become Maintainer - злоумышленник заставляет авторов заброшенных пакетов передать себе права или регистрирует на себя репозиторий под именем другого, который был ранее удален или сменил название.

Разбор атак на цепочку поставок (Supply Chain)

[AV-800] Become Maintainer - злоумышленник заставляет авторов заброшенных пакетов передать себе права или регистрирует на себя репозиторий под именем другого, который был ранее удален или сменил название.

Packt Hub

Malicious code in npm ‘event-stream’ package targets a bitcoin wallet and causes 8 million download...

Last week Ayrton Sparling, a Computer Science major at CSUF, California disclosed that the popular npm package, event-stream, contains a...

28 нояб. 2018 г.

event-stream в 2018 г. - завладели популярной библиотекой в NPM

Оценка риска компрометации через Supply Chain зависимостей

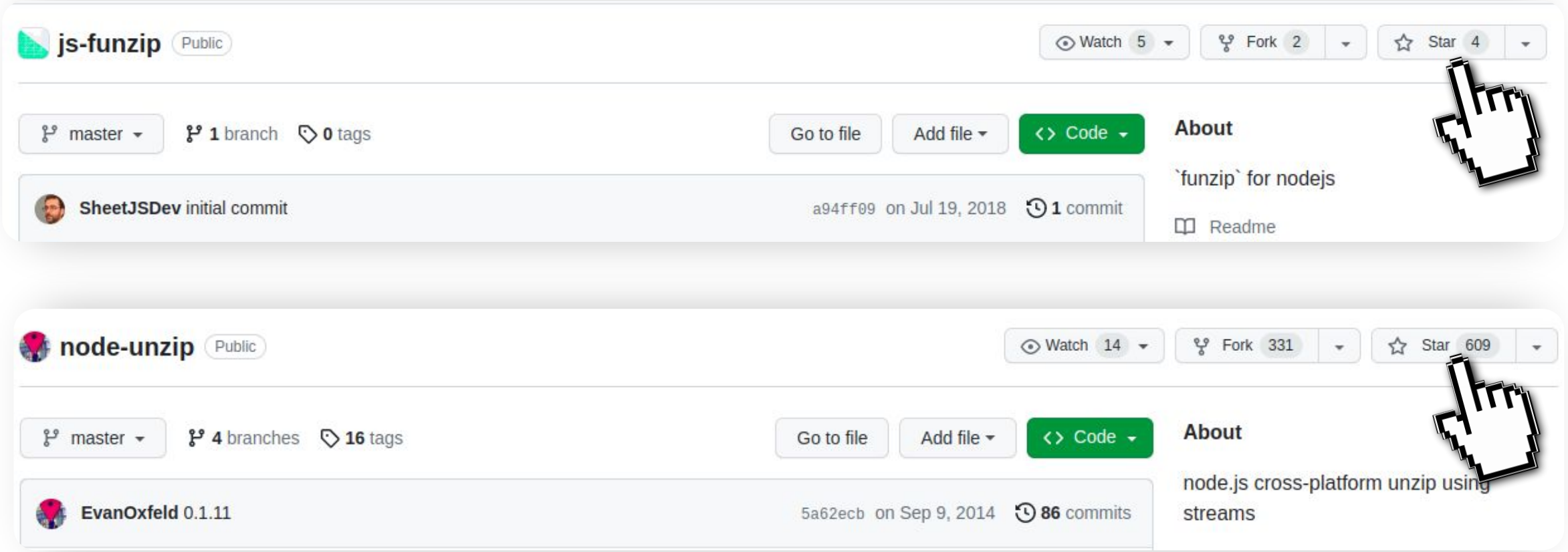
Метрики

1. Популярность
2. Авторский состав
3. Активность сообщества
4. Заинтересованность в безопасности
5. Библиотека создана недавно
6. Первая версия подозрительно высокая

Оценка риска компрометации через Supply Chain зависимостей

Метрика

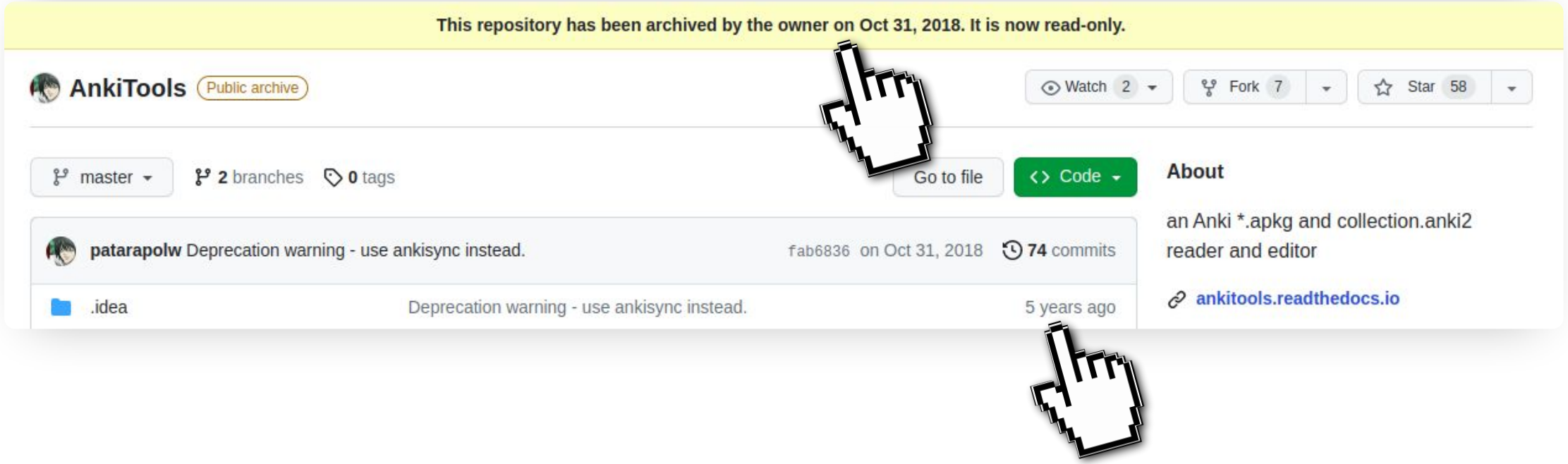
1 Популярность



Оценка риска компрометации через Supply Chain зависимостей

Метрика

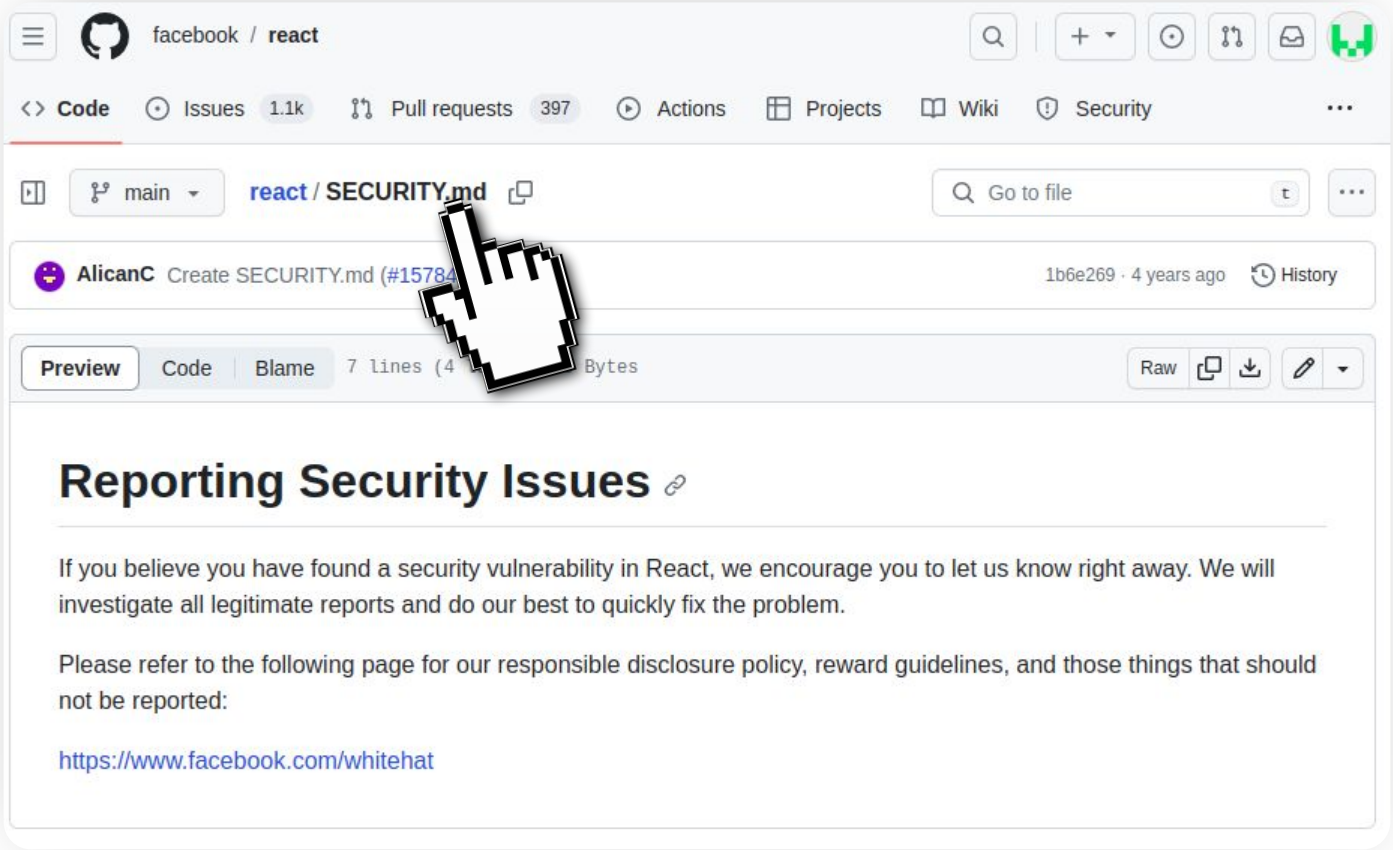
3 Активность сообщества



Оценка риска компрометации через Supply Chain зависимостей

Метрика

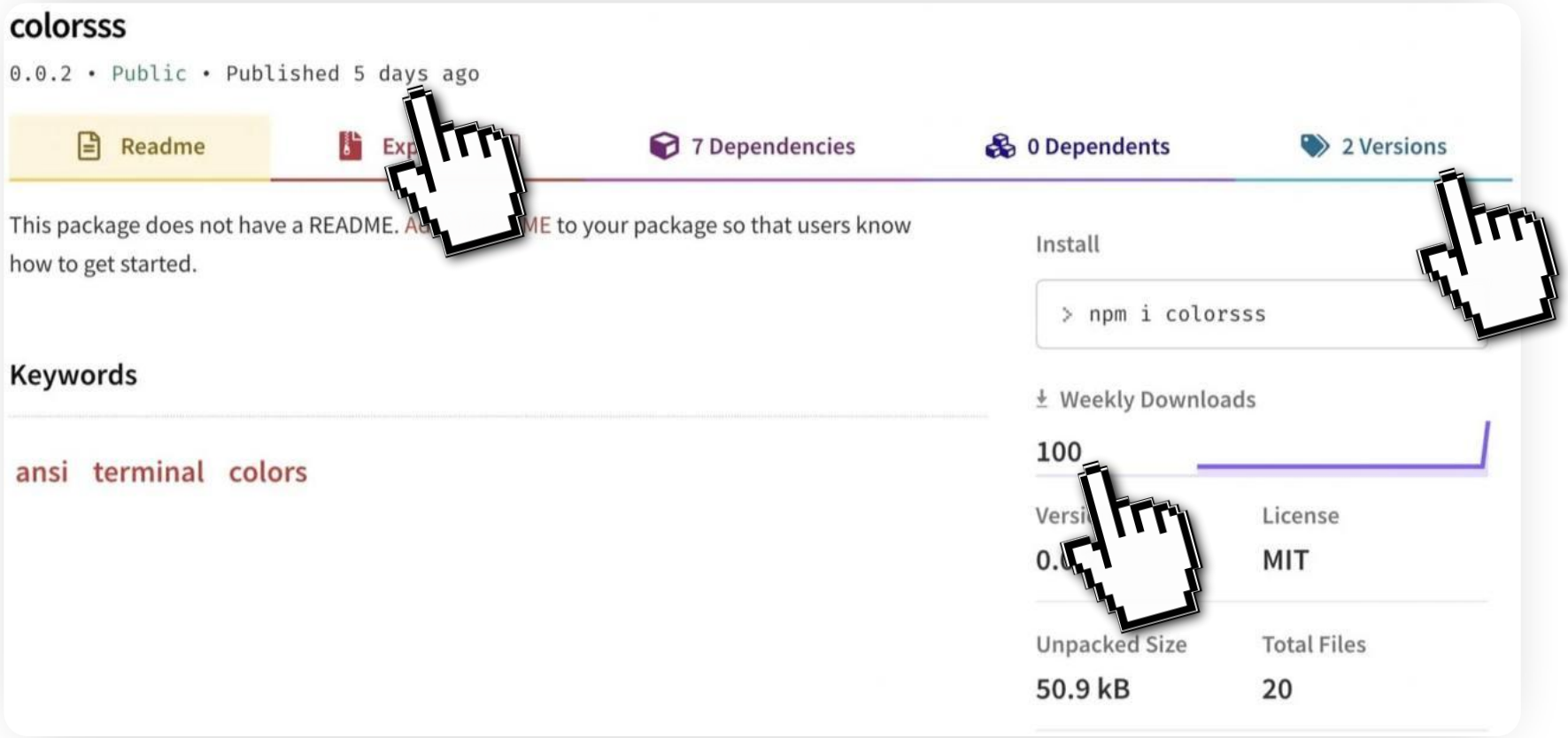
4 Заинтересованность в безопасности



Оценка риска компрометации через Supply Chain зависимостей

Метрика

5 Библиотека создана недавно



Выводы

Безопасность Supply Chain – это обеспечение безопасности **на всех этапах пути**, по которому ПО попадает в организацию, от момента создания или покупки до использования

Инвентаризация: **собирать SBOM’ы**.
Вместе с этим проще искать то самое уязвимое звено

Самое уязвимое звено цепочки поставок – **зависимости**

Превентивные меры: смотреть, что мы устанавливаем, **до установки**

Q&A

СПАСИБО ЗА ВНИМАНИЕ!

ТАТЬЯНА КУЦОВОЛ
ВЕДУЩИЙ АНАЛИТИК-ИССЛЕДОВАТЕЛЬ ИБ
ГК «СОЛАР»

t.kutsovol@rt-solar.ru
@luttatiana