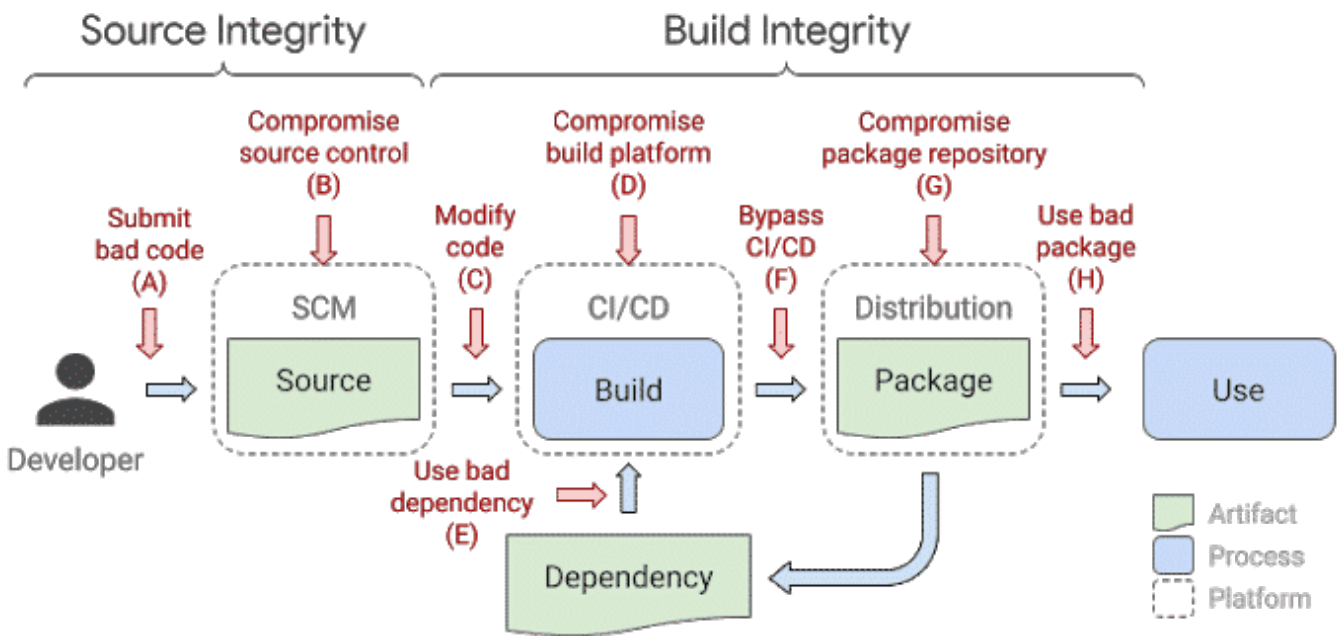
# Développement logiciel : SLSA veut sécuriser la chaîne

La « méthode Google » deviendra-t-elle la norme pour assurer l'intégrité des chaînes de développement logiciel ? Le groupe américain a en tout cas amorcé, voilà quelques semaines, un projet communautaire dans ce sens : SLSA (prononcer « salsa »).

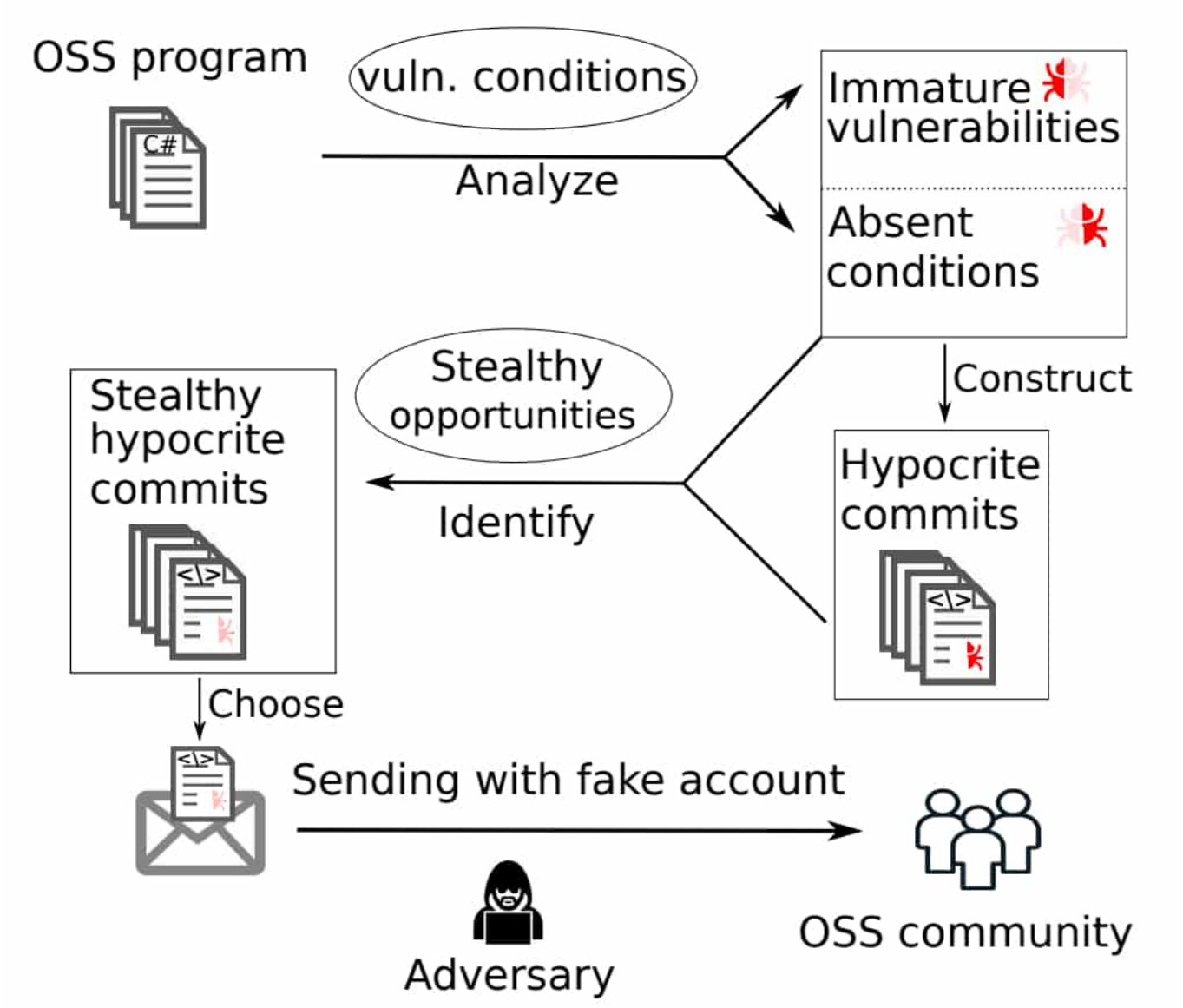
Le *framework* qu'il compte élaborer dans ce cadre se fonde sur un processus utilisé en interne depuis bientôt dix ans. Son nom : autorisation binaire pour Borg. Son principe : vérifier la provenance du code et implémenter son identité.

En l'état, SLSA se compose d'un ensemble de définitions et de lignes directrices fruits d'un « consensus industriel ». Dans sa forme finale, il devra permettre la certification automatisée des différentes composantes de la *supply chain* logicielle.

Sur cette dernière, Google distingue huit points sensibles, illustrés dans le schéma ci-dessous.




Tout au début, il y a le risque d'introduction de code malveillant dans le dépôt source. Pour exemple, une expérience qu'un chercheur de l'université du Minnesota avait menée à l'été 2020. Elle avait impliqué l'injection de vulnérabilités dans le noyau Linux.




Autre point potentiellement vulnérable : les gestionnaires de code source. Illustration avec l'attaque survenue fin mars contre le dépôt principal du PHP. La compromission du serveur Git s'était ensuivie de l'injection de code malveillant, à deux reprises. Il en avait résulté une porte dérobée.

Au niveau du serveur de *build*, il y a le risque d'usage d'un mauvais code source. C'est ce qui est arrivé en 2018 à Webmin. La modification d'un script de changement de mot de passe avait permis la mise en place de *backdoors*, en jouant notamment sur l'horodatage et l'usage d'un dossier local.

La plate-forme de *build* elle-même peut faire l'objet d'une compromission. Parmi les cas récents qui l'illustrent, il y a SolarWinds. Un implant a vérolé plusieurs mises à jour d'Orion, la plate-forme de supervision informatique de l'éditeur.

## SLSA : une certification à quatre niveaux

Autre levier : l'usage de dépendances malveillantes. En 2018, la bibliothèque event-stream en avait fait les frais. Le créateur du projet n'ayant plus le temps de le maintenir, il en avait confié les clés à un autre développeur. Lequel avait d'abord effectué des mises à jour « innocentes »… avant

d'introduire un *commit* malveillant.

L'épisode Codecov, survenu récemment, a symbolisé un autre risque : le contournement du CI/CD. La PME américaine, éditrice d'outils de gestion de la couverture de code, avait vu l'un de ses scripts détourné grâce à des identifiants qui avaient filtré. Ce script servait à cartographier les environnements de développement et à transmettre des rapports de couverture. Il a permis d'envoyer des charges malveillantes.

En aval, les gestionnaires de paquets ont eux aussi leurs fragilités. Google prend en référence le rapport d'un chercheur qui en a exploité plusieurs. En particulier au niveau des miroirs.

Tout au bout de la chaîne, reste la possibilité de pousser l'utilisateur à exploiter un paquet malveillant. Le cas de Browserify l'illustre.

SLSA comprendrait quatre niveaux de certification, résumés dans le tableau suivant.

| | | Required at | | | |
|---|---|---|---|---|---|
| Requirement | | SLSA 1 | SLSA 2 | SLSA 3 | SLSA 4 |
| Source | Version Controlled | | ✓ | ✓ | ✓ |
| | Verified History | | | ✓ | ✓ |
| | Retained Indefinitely | | | 18 mo. | ✓ |
| | Two-Person Reviewed | | | | ✓ |
| Build | Scripted | ✓ | ✓ | ✓ | ✓ |
| | Build Service | | ✓ | ✓ | ✓ |
| | Ephemeral Environment | | | ✓ | ✓ |
| | Isolated | | | ✓ | ✓ |
| | Parameterless | | | | ✓ |
| | Hermetic | | | | ✓ |
| | Reproducible | | | | ○ |
| Provenance | Available | ✓ | ✓ | ✓ | ✓ |
| | Authenticated | | ✓ | ✓ | ✓ |
| | Service Generated | | ✓ | ✓ | ✓ |
| | Non-Falsifiable | | | ✓ | ✓ |
| | Dependencies Complete | | | | ✓ |
| Common | Security | | | | ✓ |
| | Access | | | | ✓ |
| | Superusers | | | | ✓ |

○ = *required unless there is a justification*

Un premier PoC vient d'être ajouté au dépôt du projet. Il propose une intégration de niveau 1 avec GitHub Actions.

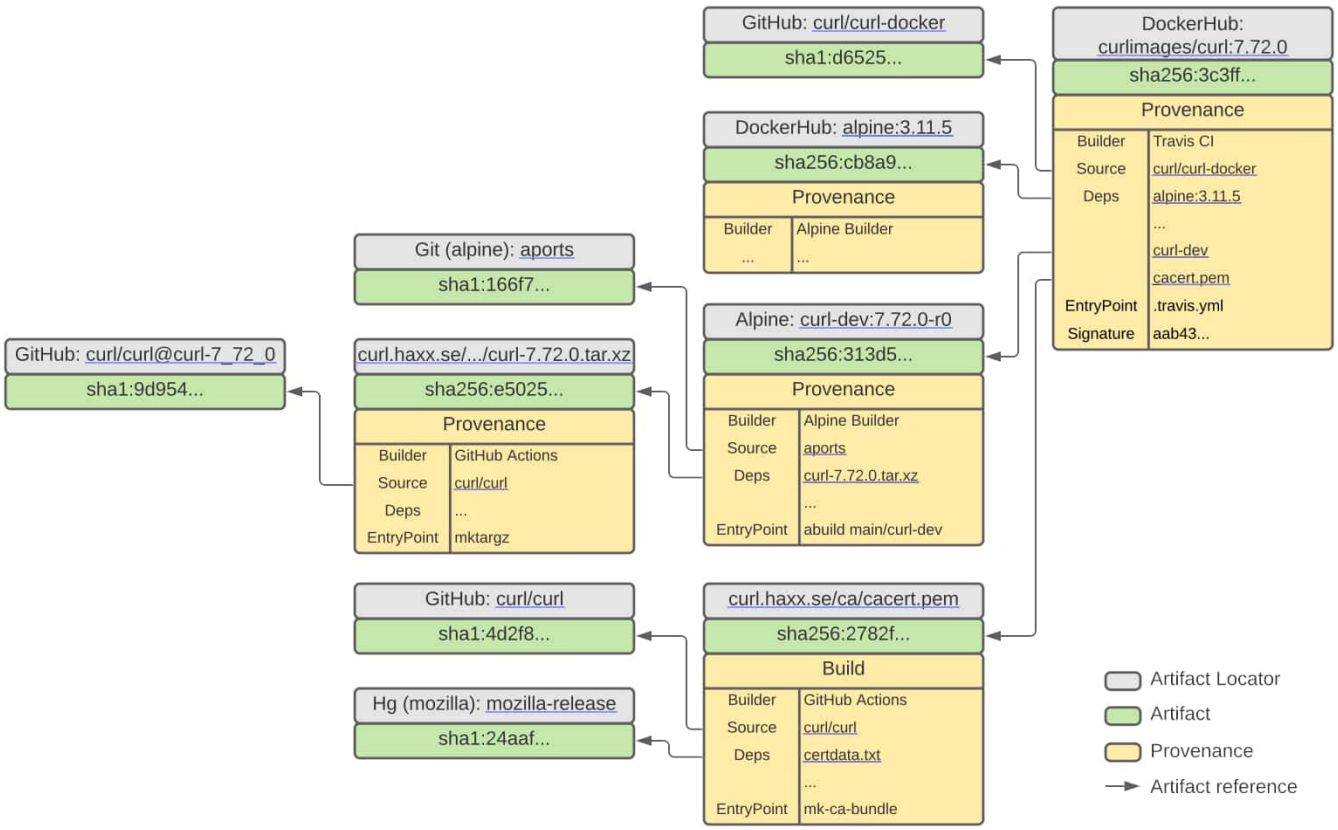



*Illustration principale © Ferenc Almasi – Unsplash*