Don’t forget to share audio

Maxime Rainville
Senior software engineer

Handling product security like a PRO.

Maxime Rainville
Senior software engineer

About Maxime

- Working with Silverstripe CMS since 2011
- Worked at Silverstripe for 6 years
  - 2 years as CMS Squad team lead
- 4th time presenting at StripeCon Europe

ARCHIPRO

Our mission is to simplify the end to end building experience for homeowners and professionals

Homeowner
Planning
Residential New Build
Auckland

Homeowner
Planning
Residential Renovation
Christchurch

Professional
Planning
Commercial New Build
Auckland

Homeowner
Completing
Residential New Build
Wellington

Homeowner
Planning
Residential New Build
Whitianga

Homeowner
Building
Residential New Build
Dunedin

Homeowner
Exploring
Residential Renovation
Auckland

Professional
Planning
Commercial New Build
Auckland

Professional
Planning
Residential New Build
Auckland

Architect

Builder

Interior Designer

Landscaper

Product Designer

Architectural Designer

Construction

Architects

Consultants

ARCHIPRO

People Building do not have an effective and trusted way to find pros and products online

Companies face significant challenges to cost-effectively promote their products and services

ArchiPro is hiring!

Infrastructure Engineer

Product Designer (UX/UI)

Media Sales and Partnerships

archipro.co.nz/careers

GM Customer Success

Head of Sales Australia

Client Account Director

Disclaimer

- I’m NOT an InfoSec expert.
- I was member of Silverstripe’s ISSC.
- But I don’t work for Silverstripe anymore.

Why you should care about product security

- You use Silverstripe CMS.
- You sell Silverstripe CMS.
- You maintain an OSS library or a software product.
- You are a human living in 2024.

The good old security days.

Let’s go back to 2000

- Yahoo was worth 125 billions USD
- Y2K bug
- ILOVEYOU virus
- Mafia Boy DDOS attack

2000’s approach to security

- Aim to never have any vulnerabilities
- Prescriptive approach to security
- Security by obscurity
- Vulnerabilities are shameful

The problems with the *Good old* approach to security

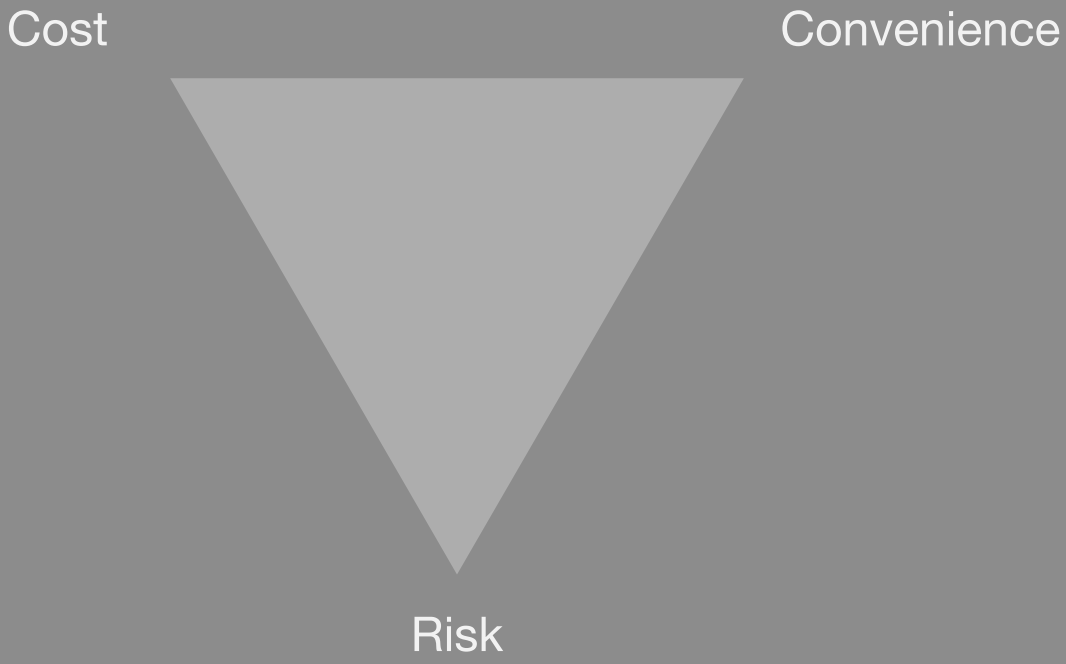
- **Every** product has vulnerabilities.
- Prescriptive rules don’t work.
- Hackers and security researchers are really good.
- Blame culture get in the way of finding and fixing problems.

The risk management approach.

Risk management

1. Identify risks
2. Evaluate impact/probability
3. Decide how to mitigate them
    a. Accept
    b. Avoid
    c. Mitigate
    d. Transfer
4. Plan responses

Balancing risk with other constraints



Example risk

A user upload an SVG file with an XSS payload.

The new approach

- ~~Aim to never have any vulnerabilities~~
  You will be vulnerable … plan accordingly
- ~~Presciptive approach to security~~
  Adapt your approach to your changing context
- ~~Security by obscurity~~
  Be transparent with your customers
- ~~Vulnerabilities are shameful~~
  Collaborative industry best practices

How vulnerabilities are managed for Silverstripe CMS.

Try not to introduce vulnerability in the first place

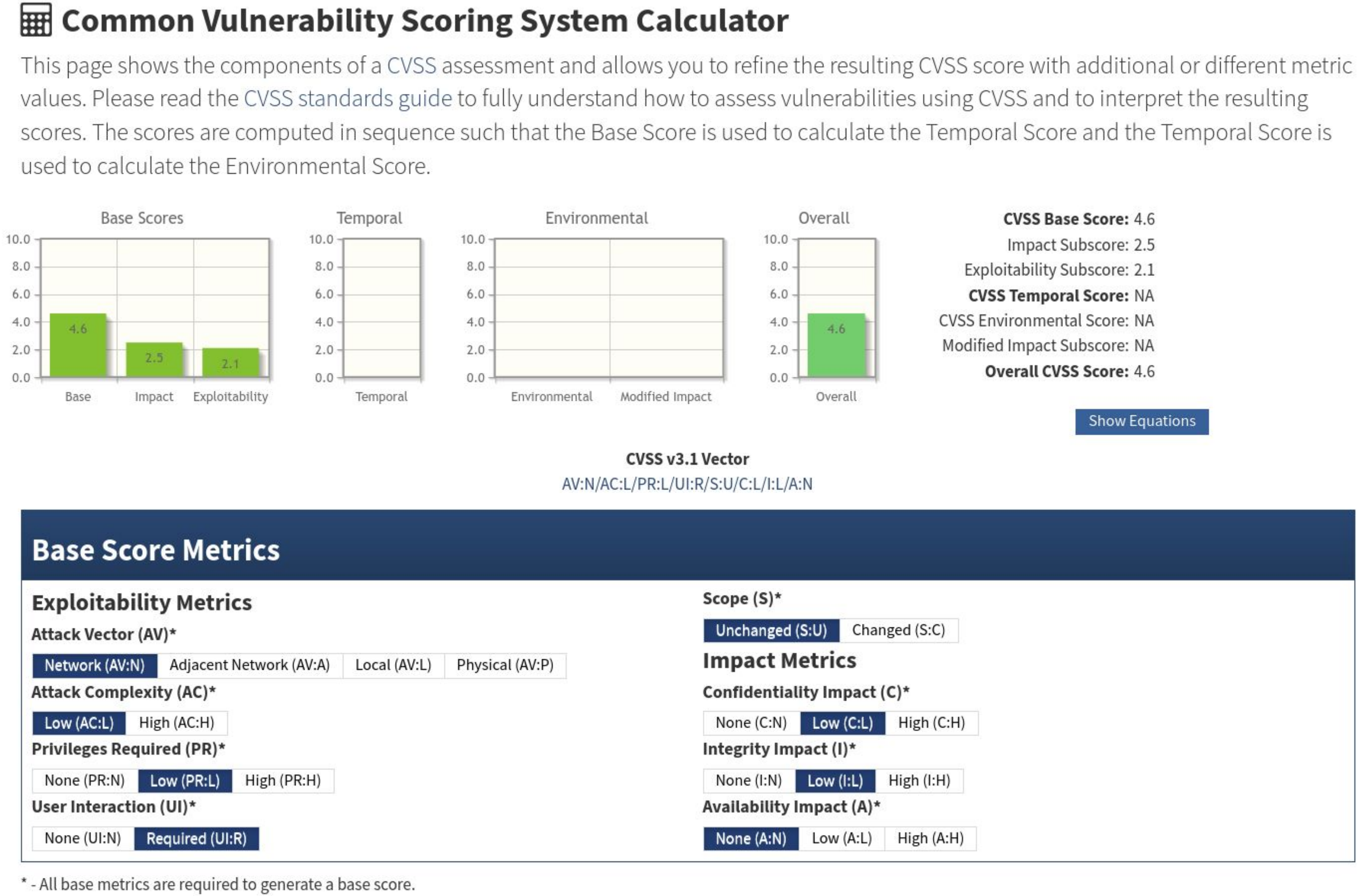
- Technical risk analysis
- Peer review
- Secure coding standards
- Independent code audit

Reporting vulnerabilities

- Provide a clear way to report vulnerabilities
  - Email security@silverstripe.org
- Report even if you are not sure

Evaluating a potential vulnerability report

- Is it SPAM?
- Can it be replicated?
- Can it be exploited and in what context?
- What’s the severity of the vulnerability?



Common Vulnerability Scoring System Calculator

This page shows the components of a CVSS assessment and allows you to refine the resulting CVSS score with additional or different metric values. Please read the CVSS standards guide to fully understand how to assess vulnerabilities using CVSS and to interpret the resulting scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.

CVSS Base Score: 4.6
Impact Subscore: 2.5
Exploitability Subscore: 2.1
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 4.6

CVSS v3.1 Vector
AV:N/AC:L/PR:L/UI:R/S:U/C:L/I:L/A:N

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*
[✓ Network (AV:N)] Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*
[✓ Low (AC:L)] High (AC:H)

Privileges Required (PR)*
None (PR:N) [✓ Low (PR:L)] High (PR:H)

User Interaction (UI)*
None (UI:N) [✓ Required (UI:R)]

Scope (S)*
[✓ Unchanged (S:U)] Changed (S:C)

Impact Metrics

Confidentiality Impact (C)*
None (C:N) [✓ Low (C:L)] High (C:H)

Integrity Impact (I)*
None (I:N) [✓ Low (I:L)] High (I:H)

Availability Impact (A)*
[✓ None (A:N)] Low (A:L) High (A:H)

* - All base metrics are required to generate a base score.

Impact level

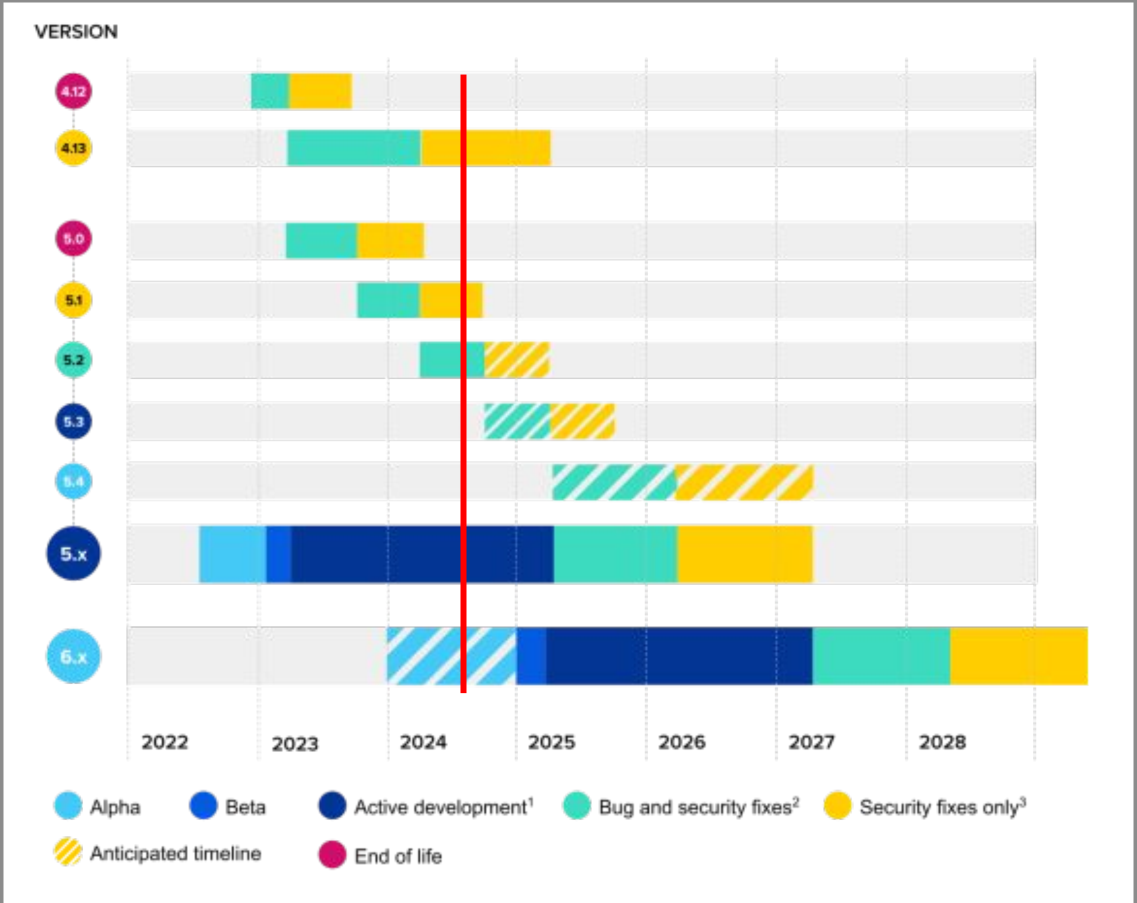
- 0.1-3.9: Low impact
- 4.0-6.9: Medium impact
- 7.0-8.9: High impact
- 9.0-10: Critical impact

Patching the vulnerability

- Develop the fix
- Peer review
- Look for related issues
- Learn from your mistakes

Releasing the security patches

- Be predictable
    - Which version will be patched
    - When will the patch be released
- Security patches are released
    - April
    - June
    - October
    - January


VERSION
4.12
4.13
5.0
5.1
5.2
5.3
5.4
5.x
6.x
2022
2023
2024
2025
2026
2027
2028
Alpha
Beta
Active development[1]
Bug and security fixes[2]
Security fixes only[3]
Anticipated timeline
End of life

- Write good communication
  - Common Vulnerability Scoring System (CVSS)
  - Common Vulnerabilities and Exposures (CVE)

Allow community to mitigate risk

- Review the score
- Review the affected component
- Manage your own risk

Putting it all in practice.

As a developer

- Manage the risk in your code
- Most vulnerabilities are simple mistakes
  - Forget to do a *CanView* check
  - API endpoint not checking the user is authenticated
  - Not escaping a query parameter

- Manage risks
    - What’s your OSS project used for?
    - How is it put together?
    - What are some of the risk that might affect my users?
- Balance risk with your commitment

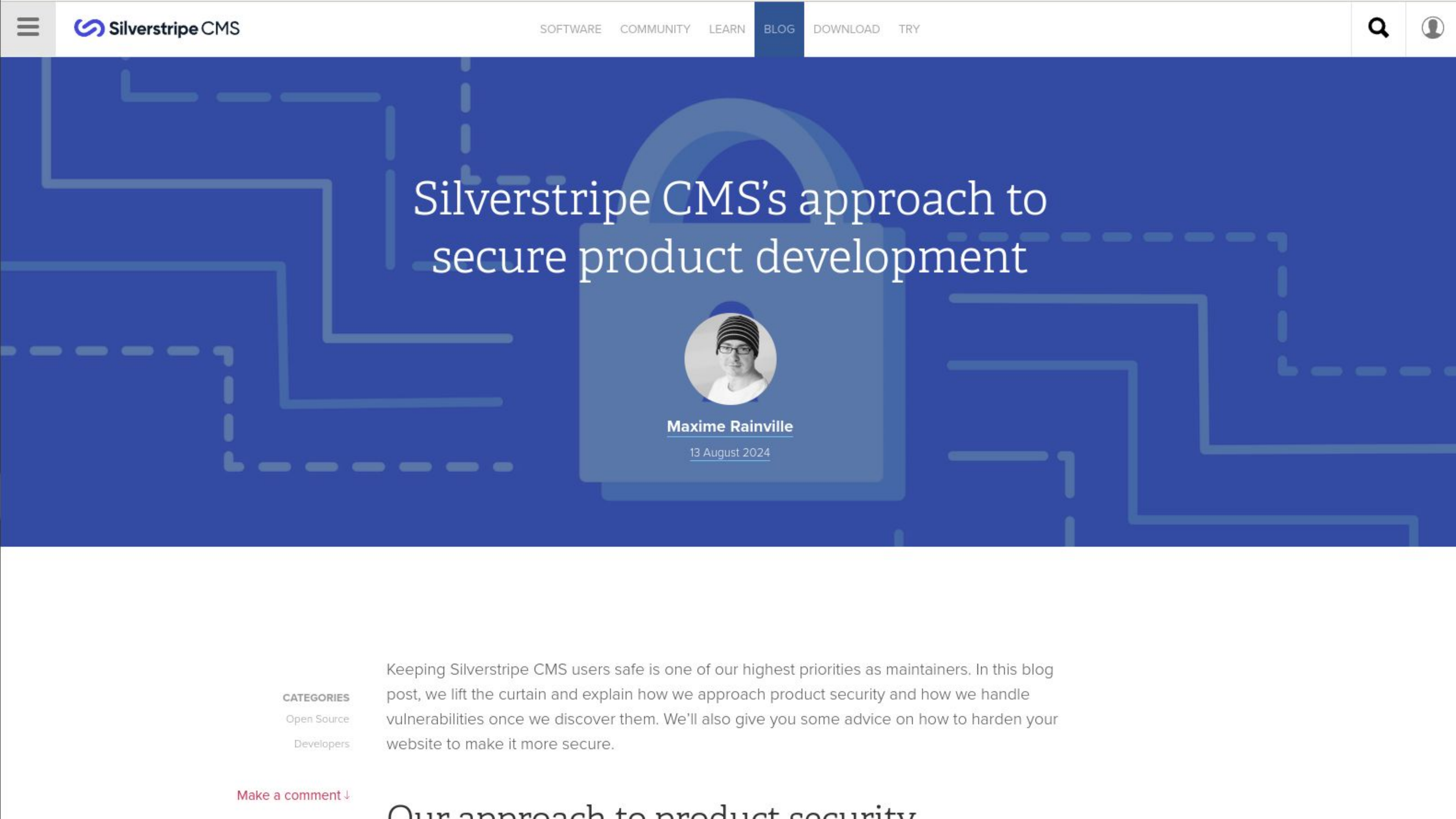
Vulnerability management tools built into GitHub

- Vulnerability report
- Developing confidential fix
- CVE and CVSS
- Automatically notification of third parties

As an organisation using Silverstripe CMS

- Risk management
- Align your processes with Silverstripe’s processes
- Adapt to your customer risk profiles

Next steps.



Silverstripe CMS's approach to secure product development

Maxime Rainville

13 August 2024

CATEGORIES

Open Source

Developers

Make a comment ↓

Keeping Silverstripe CMS users safe is one of our highest priorities as maintainers. In this blog post, we lift the curtain and explain how we approach product security and how we handle vulnerabilities once we discover them. We'll also give you some advice on how to harden your website to make it more secure.

Our approach to product security

Join OWASP

- Open Worldwide Application Security Project
  - Non-profit,
  - Driven by volunteers
- Read the OWASP top-10

Certifications

AICPA
SOC
aicpa.org/soc4so
SOC for Service Organizations | Service Organizations
™
SOC II
COMPLIANT

Thank you!