



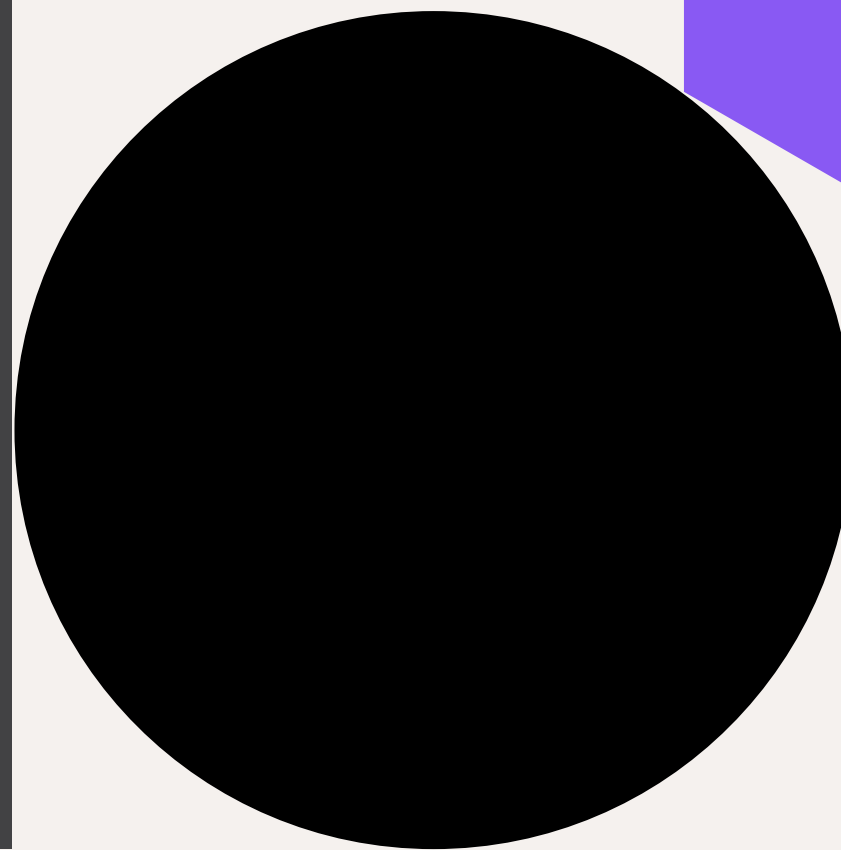
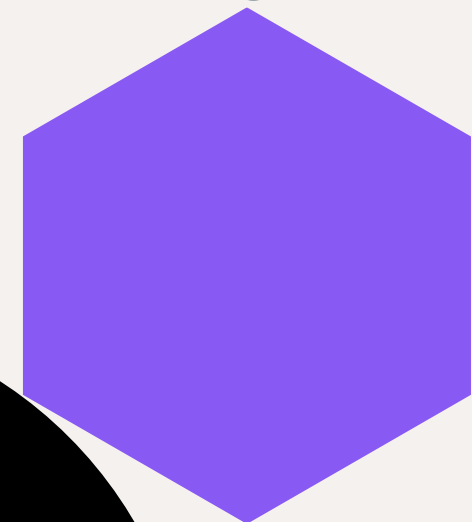
# CHERI ALLIANCE 101



RISC-V Tokyo Day 2025 Autumn  
オープンアーキテクチャ・ライトニングトーク ②

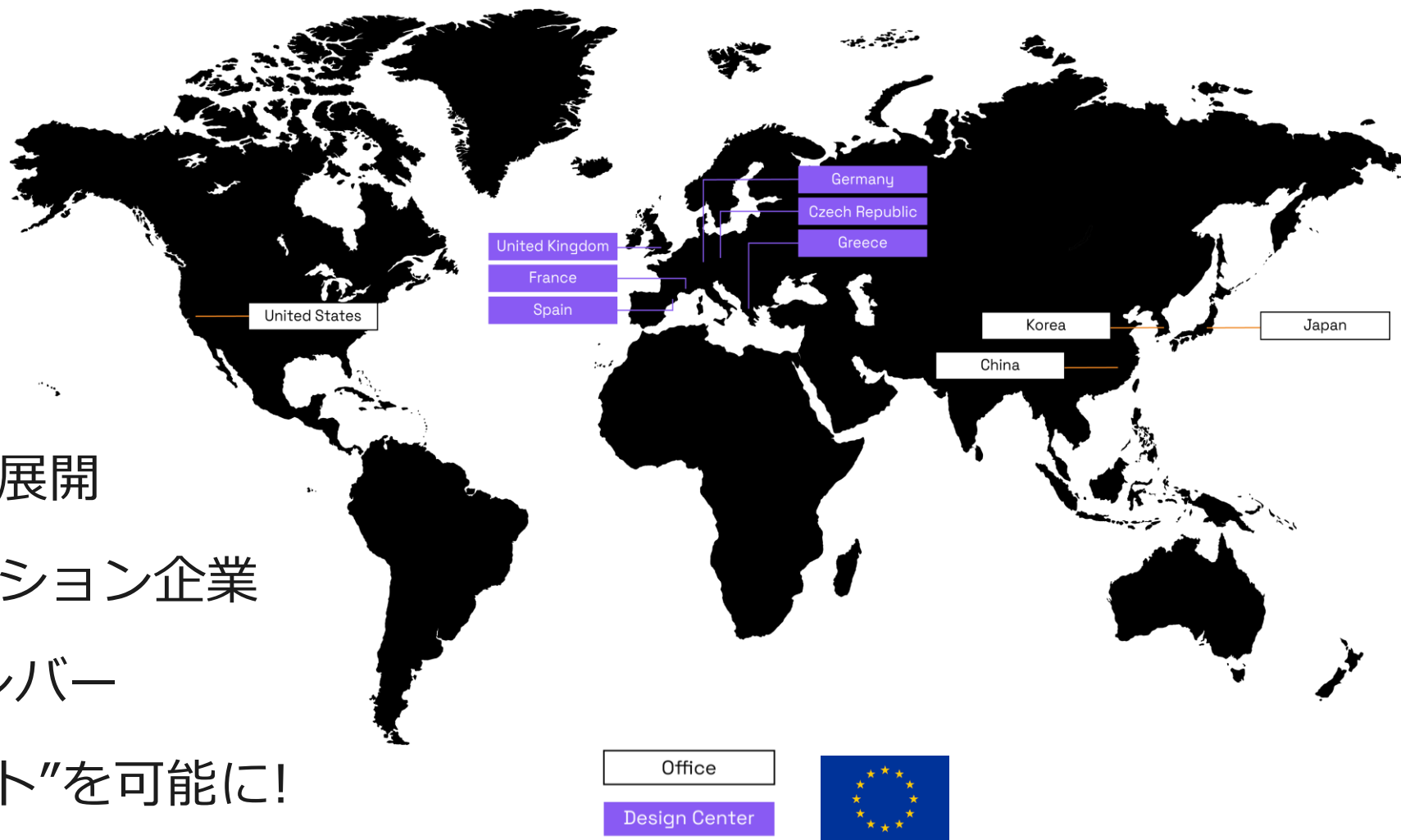
Takaaki Akashi, Country Manager – Japan, Cudasip GmbH

2025/12/04, 16:55 – 17:05



## → コダシップ概要

- 2014年設立
- 独ミュンヘン本社
- 約250名の社員
- ヨーロッパで開発
- グローバルなオフィス展開
- プロセッサ・ソリューション企業
-  **RISC-V®** 創設メンバー
- “カスタム コンピュート”を可能に!

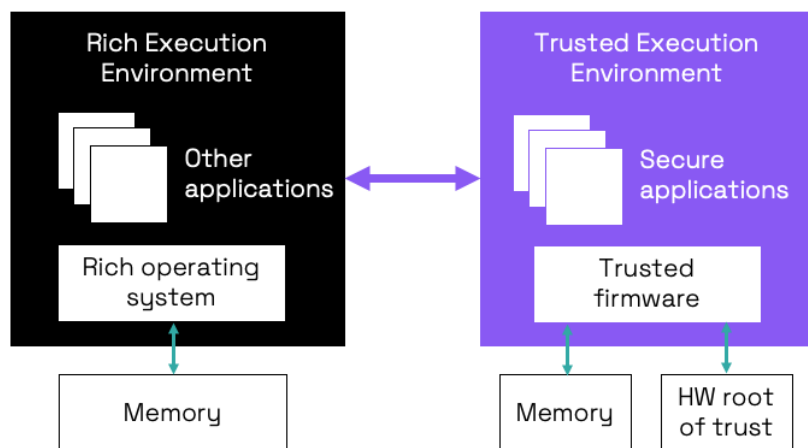


## → Press releases / blogs

- [Codasip RISC-VプロセッサはASIL-Dまで認証済](#)
- [コダシップ取締役会は、戦略的 会社売却の手続きを開始](#)
- [CHERIの採用を加速する完全な探索プラットフォームを発表](#)
- [EUが資金を提供するDAREプロジェクト向けのハイエンドRISC-Vプロセッサの設計を担当](#)
- [CHERIアライアンスにCHERI RISC-V SDKを寄贈し、安全なLinuxを実現](#)
- [汎用性の高い車載用組み込みRISC-Vコアを発表](#)
- [業界最高レベルの電力効率を達成したカスタマイズ可能なRISC-Vコアを発表](#)
- [自動車の機能安全とサイバーセキュリティの認証を取得](#)
- [最もポピュラーなサイバー攻撃を積極的に防ぐプロセッサ セキュリティを提供します](#)
- [カスタム コンピュートに最適な次世代RISC-Vプロセッサ ファミリを発表](#)
- [セキュアなハードウェア認証技術にVerilockを採用](#)
- [シーメンスと協業し、カスタム・プロセッサ向けトレース・ソリューションを提供](#)

# → プロセッサ周りのセキュリティ技術...

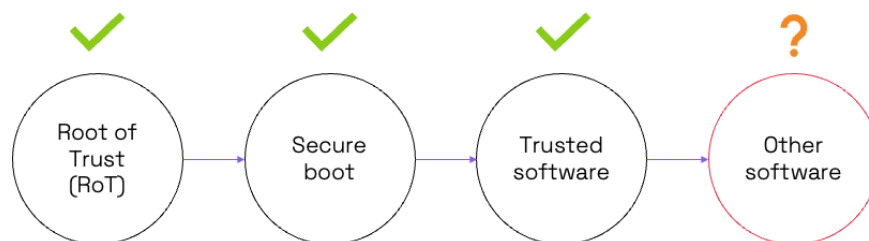
## 隔離実行環境(TEE)



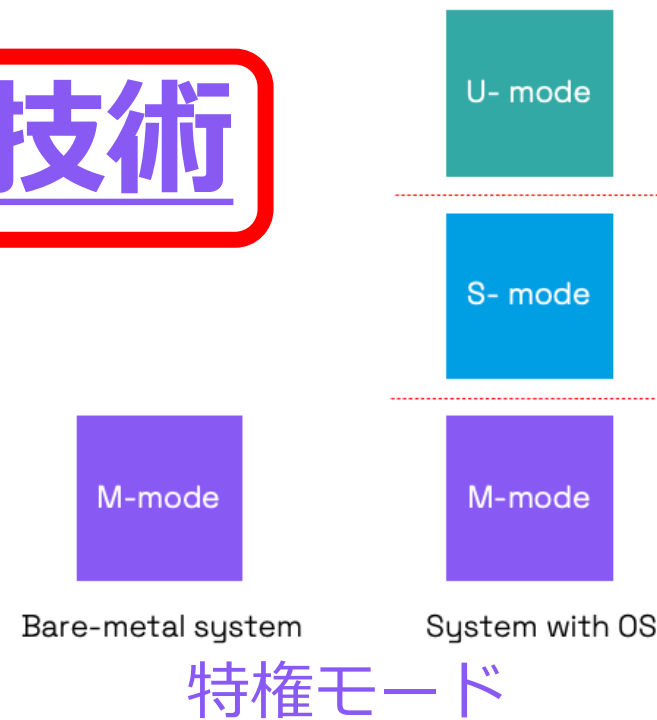
## サイドチャネル攻撃対策

# メモリ安全技術

## 各種暗号化



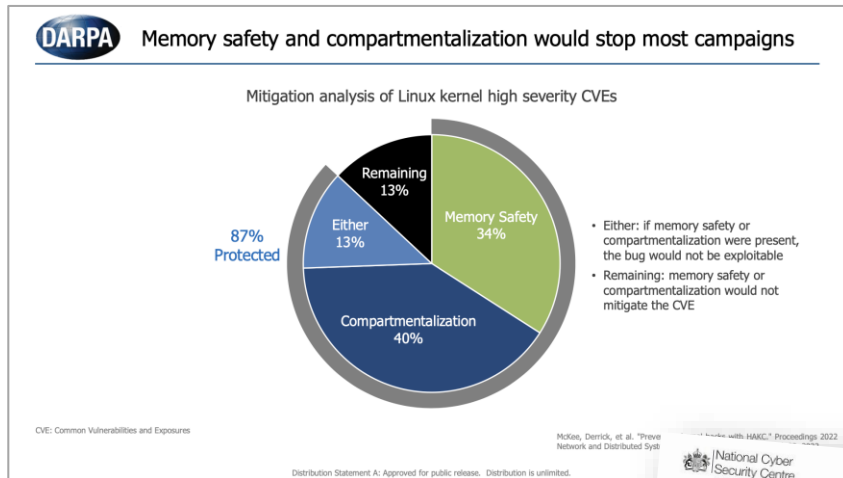
## 信頼の起点(RoT)/チェーン



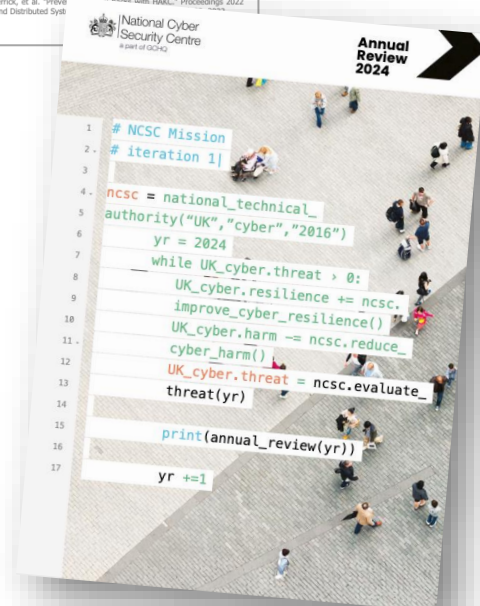
## 特権モード

## RISC-V PMP(Physical Memory Protection)

# Memory safety becomes a key topic



<https://www.ncsc.gov.uk/collection/ncsc-annual-review-2024>



Article: <https://bidenwhitehouse.archives.gov/oncd/briefing-room/2024/02/26/press-release-technical-report/>  
Report: <https://bidenwhitehouse.archives.gov/wp-content/uploads/2024/02/Final-ONCD-Technical-Report.pdf>  
(CHERI mentioned on p9)

Highlight  
CHERI as  
a solution



<https://www.cisa.gov/news-events/news/urgent-need-memory-safety-software-products>

# ○ About CHERI



- ◆ Initiated by a project from



- ◆ Originally developed by



- ◆ Matured for 15 years
- ◆ Supported by



**~ \$300 million investment in the development of CHERI**  
(by governments and industry)



# Introduction to the CHERI Alliance

**Why / What / How?**

# Executive Summary

# ○ The CHERI Alliance is an independent entity

- ◆ Non-profit
- ◆ Focused on accelerating CHERI adoption
- ◆ ISA-agnostic
- ◆ Worldwide reach
- ◆ Drive / stimulate ecosystem
  - ◆ Industry
  - ◆ Research
  - ◆ Government
  - ◆ Communities

- ◆ Industry financed
  - ◆ Long-term sustainability
- ◆ Start-up phase with financial support of

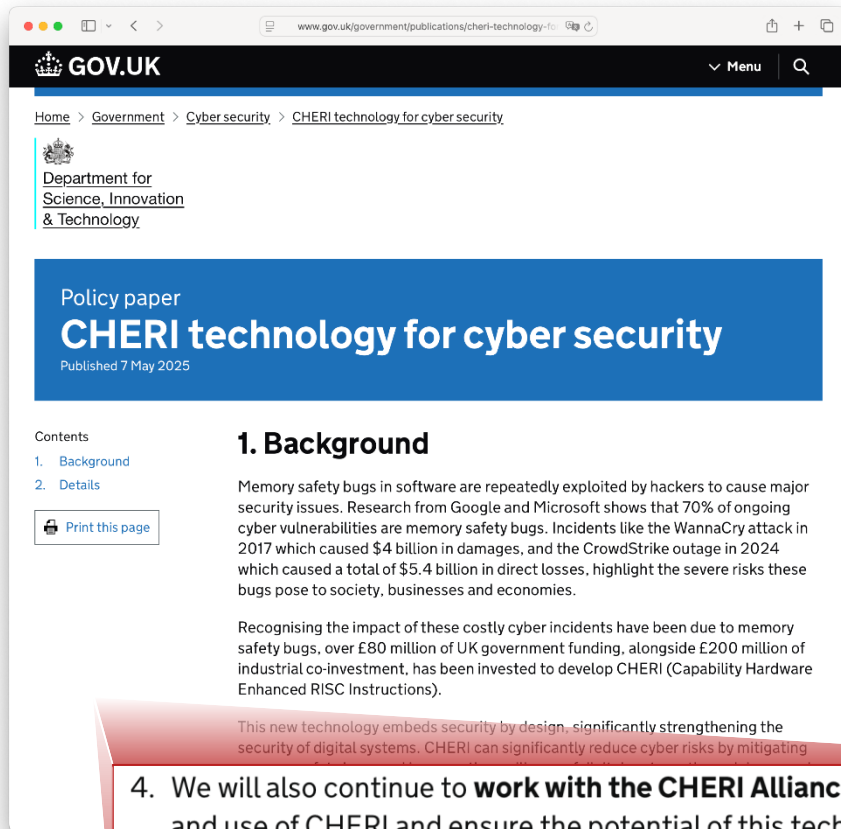


Department for  
Science, Innovation,  
& Technology

# CHERI Alliance members – updated on 08 December 2025



# UK govt's partner for CHERI adoption



GOV.UK

Home > Government > Cyber security > CHERI technology for cyber security

Department for Science, Innovation & Technology

Policy paper  
**CHERI technology for cyber security**  
Published 7 May 2025

Contents

1. Background
2. Details

 Print this page

### 1. Background

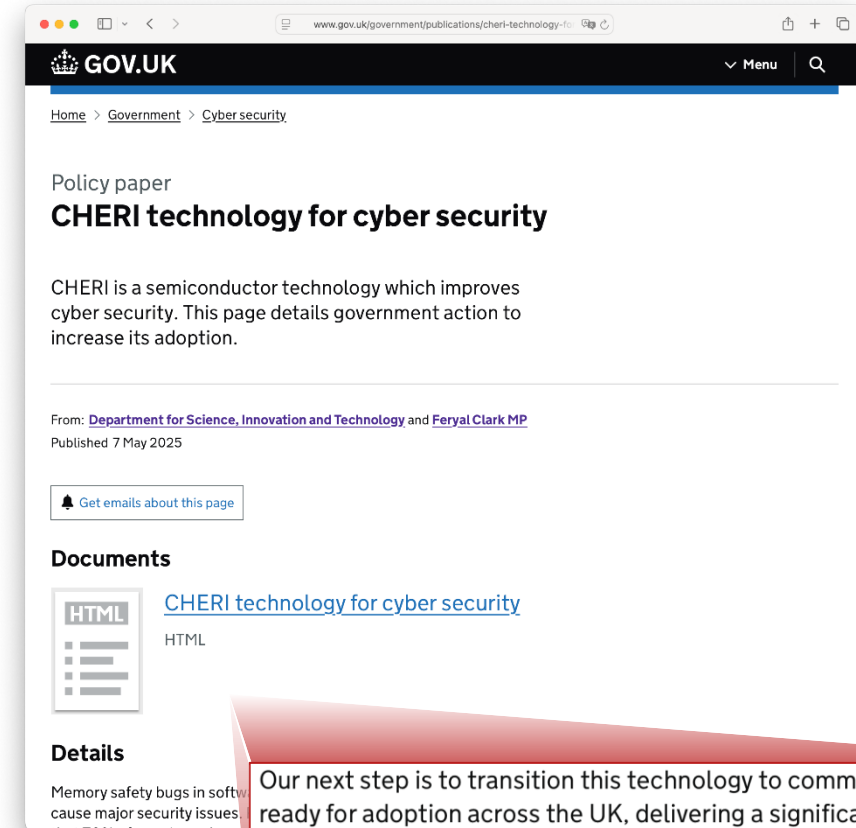
Memory safety bugs in software are repeatedly exploited by hackers to cause major security issues. Research from Google and Microsoft shows that 70% of ongoing cyber vulnerabilities are memory safety bugs. Incidents like the WannaCry attack in 2017 which caused \$4 billion in damages, and the CrowdStrike outage in 2024 which caused a total of \$5.4 billion in direct losses, highlight the severe risks these bugs pose to society, businesses and economies.

Recognising the impact of these costly cyber incidents have been due to memory safety bugs, over £80 million of UK government funding, alongside £200 million of industrial co-investment, has been invested to develop CHERI (Capability Hardware Enhanced RISC Instructions).

This new technology embeds security by design, significantly strengthening the security of digital systems. CHERI can significantly reduce cyber risks by mitigating

4. We will also continue to **work with the CHERI Alliance**, to encourage the take up and use of CHERI and ensure the potential of this technology is realised. Finally, we will work with international partners to persuade governments and industry to prioritise secure by design and memory safety.

<https://www.gov.uk/government/publications/cheri-technology-for-cyber-security/cheri-technology-for-cyber-security>



GOV.UK

Home > Government > Cyber security

Policy paper  
**CHERI technology for cyber security**

CHERI is a semiconductor technology which improves cyber security. This page details government action to increase its adoption.

From: [Department for Science, Innovation and Technology](#) and [Feryal Clark MP](#)  
Published 7 May 2025

 Get emails about this page

### Documents

 [CHERI technology for cyber security](#)  
HTML

### Details

Memory safety bugs in software cause major security issues. Research from Google and Microsoft shows that 70% of ongoing cyber vulnerabilities are memory safety bugs. Incidents like the WannaCry attack in 2017 which caused \$4 billion in damages, and the CrowdStrike outage in 2024 which caused a total of \$5.4 billion in direct losses, highlight the severe risks these bugs pose to society, businesses and economies.

Our next step is to transition this technology to commercial products, ready for adoption across the UK, delivering a significant advancement in innovative cyber security. Leveraging industry co-investment, we will **partner with the CHERI Alliance**, InnovateUK, the University of Cambridge and other stakeholders to drive this effort.

<https://www.gov.uk/government/publications/cheri-technology-for-cyber-security>

# ○ Hardware-based solution to a software problem

## ◆ CHERI

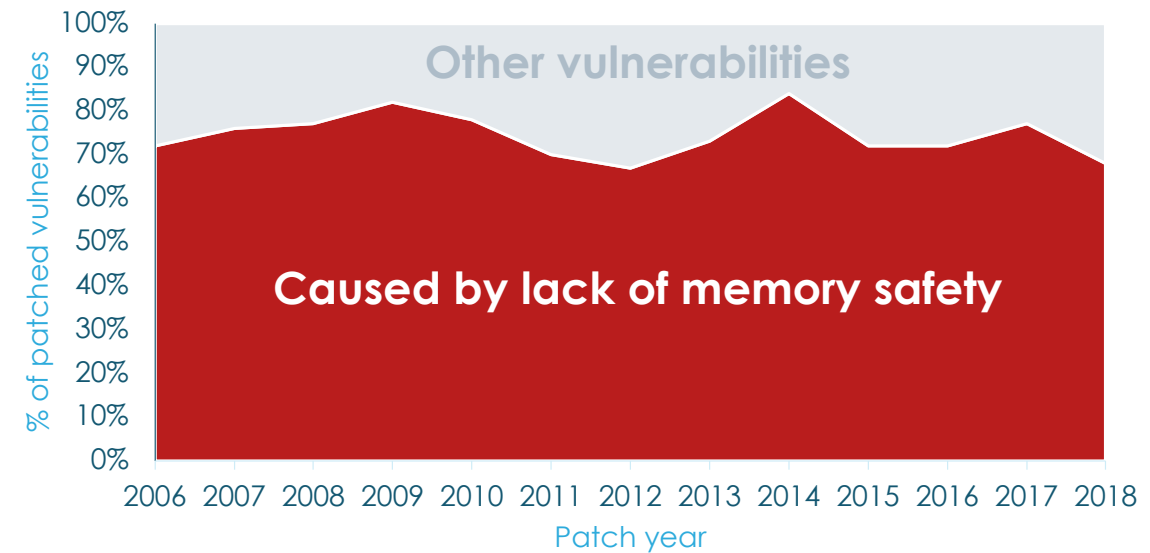
- ◆ Open technology
- ◆ Hardware protection at processor-level
- ◆ Brings strong security to **existing code**

## ◆ Now out of the lab

- ◆ 15 years of development by University of Cambridge (UK) / SRI (USA)
- ◆ Formally proven
- ◆ Demo and prototypes available – now getting into products
- ◆ Ready for industry to adopt

# ○ Solving the worst cybersecurity issue

- Cyberattacks cause more than \$10 trillion of damage / year
- Most attacks due to a lack of **Memory safety**
- Unsolvable problem with traditional software solutions (new languages, practices, tools...)
- **CHERI** technology solves the memory safety problem



Source: [Trends, challenge, and shifts in software vulnerability mitigation](#) - Microsoft 2019

## ○ Data breaches are very costly

>\$500M

Cost of addressing  
Heartbleed buffer overflow  
vulnerability

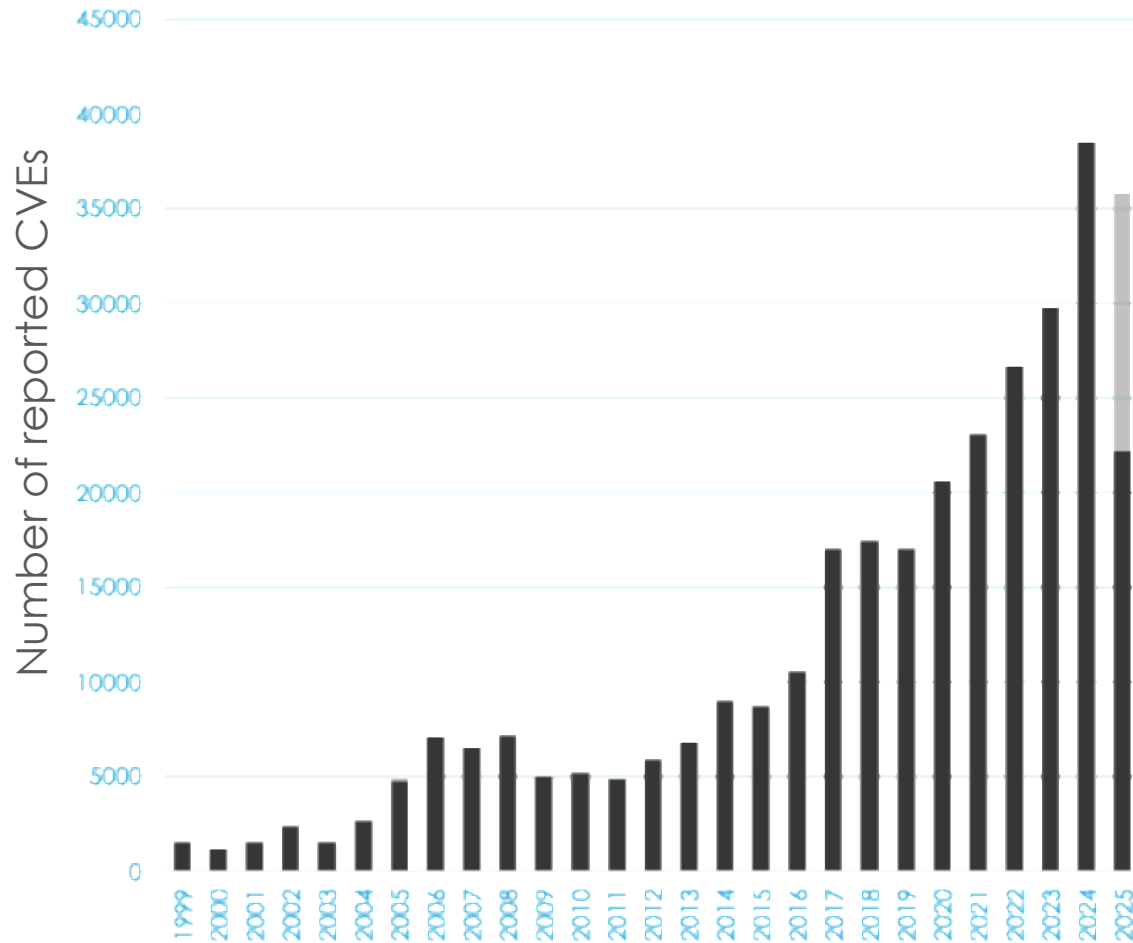
6X

Increase in firmware attacks  
reported by NIST between  
2017 and 2024

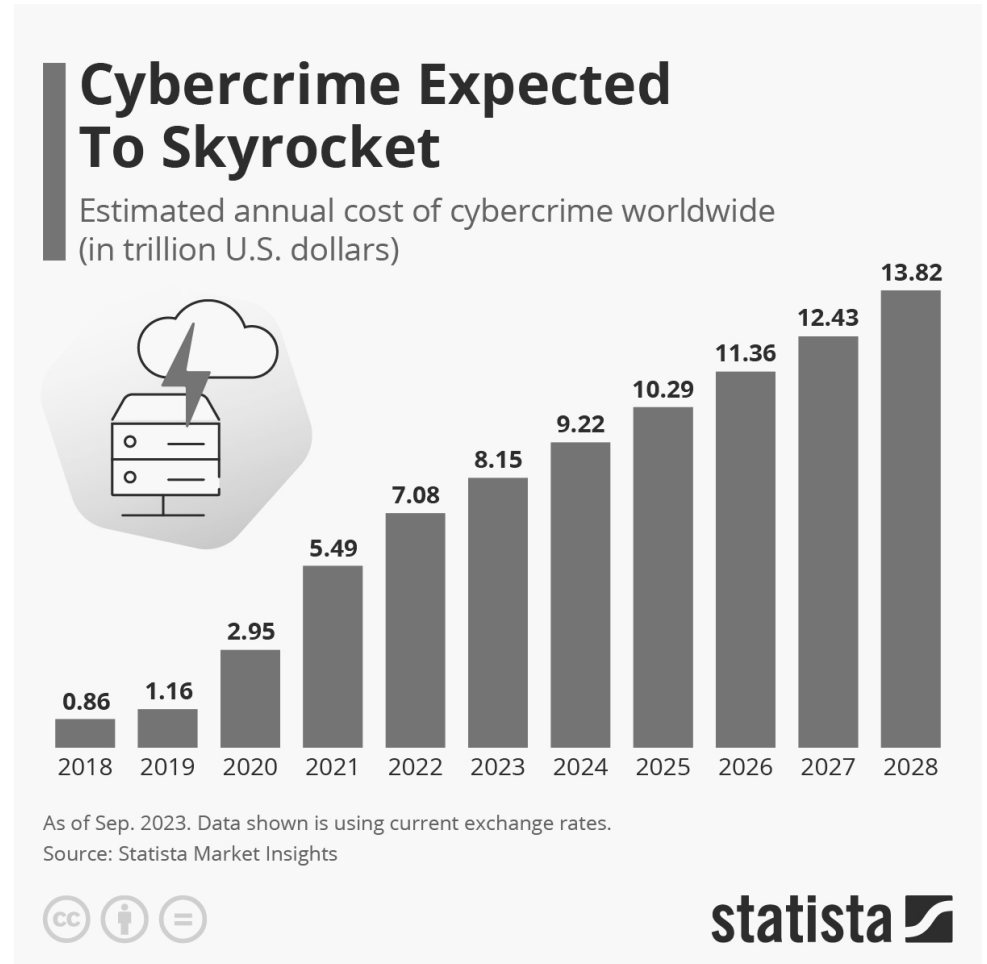
~\$10T

Worldwide cost estimate of  
cyberattacks per year  
(and growing fast)

# ○ Software vulnerabilities are causing increasing risk



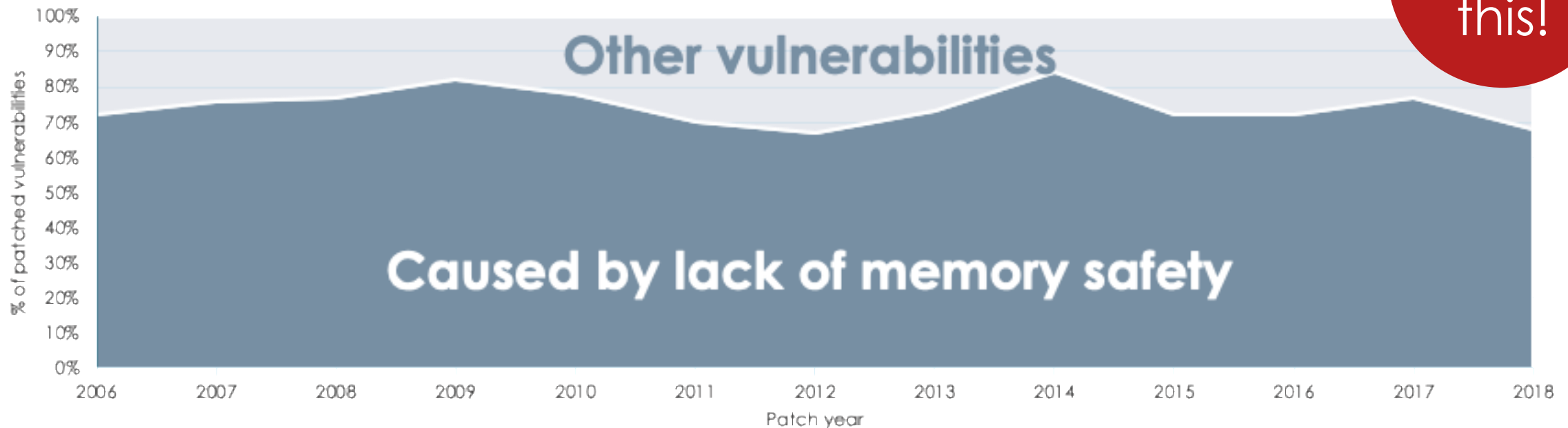
Source: <https://www.cve.org>



# ○ The main problem is (the lack of) memory safety

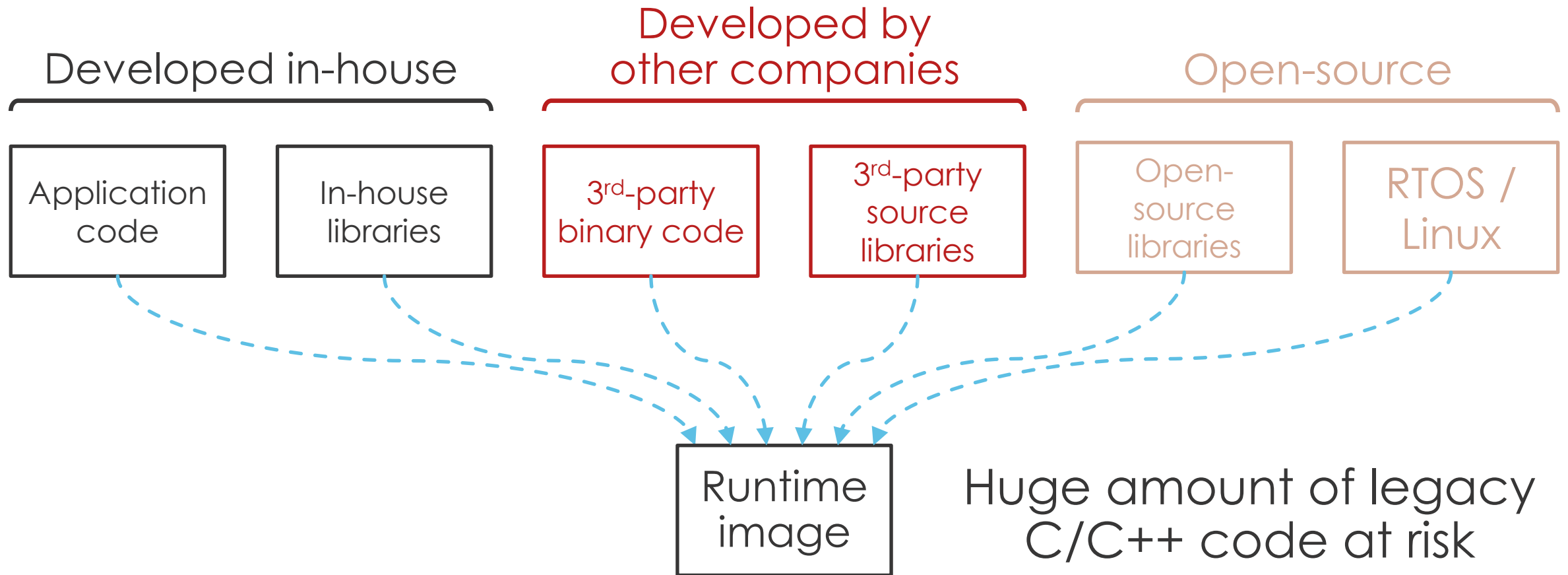
- ◆ Memory abuse (e.g. buffer overflows) is the main attack vector
- ◆ Constant ratio of over the past 20 years
  - ◆ ... even with all the work done on software to avoid this!

CHERI  
solves  
this!



Source: [Trends, challenge, and shifts in software vulnerability mitigation](#) - Microsoft 2019

# ○ Impossible to re-write software to fix the problem



# → Android 5.0 – 14.0 CVE 累計 2024/01/16 時点

| Year                 | Overflow                    | Memory Corruption           | Sql Injection            | XSS                     | Directory Traversal      | File Inclusion          | CSRF                    | XXE                     | SSRF                    | Open Redirect           | Input Validation           |
|----------------------|-----------------------------|-----------------------------|--------------------------|-------------------------|--------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|----------------------------|
| <a href="#">2015</a> | <a href="#">48</a>          | <a href="#">41</a>          | <a href="#">0</a>        | <a href="#">0</a>       | <a href="#">0</a>        | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">6</a>          |
| <a href="#">2016</a> | <a href="#">175</a>         | <a href="#">84</a>          | <a href="#">0</a>        | <a href="#">0</a>       | <a href="#">0</a>        | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">96</a>         |
| <a href="#">2017</a> | <a href="#">189</a>         | <a href="#">108</a>         | <a href="#">0</a>        | <a href="#">0</a>       | <a href="#">0</a>        | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">61</a>         |
| <a href="#">2018</a> | <a href="#">41</a>          | <a href="#">141</a>         | <a href="#">5</a>        | <a href="#">0</a>       | <a href="#">6</a>        | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">2</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">40</a>         |
| <a href="#">2019</a> | <a href="#">56</a>          | <a href="#">257</a>         | <a href="#">12</a>       | <a href="#">0</a>       | <a href="#">1</a>        | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">41</a>         |
| <a href="#">2020</a> | <a href="#">169</a>         | <a href="#">338</a>         | <a href="#">18</a>       | <a href="#">0</a>       | <a href="#">10</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">89</a>         |
| <a href="#">2021</a> | <a href="#">147</a>         | <a href="#">341</a>         | <a href="#">4</a>        | <a href="#">0</a>       | <a href="#">12</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">122</a>        |
| <a href="#">2022</a> | <a href="#">196</a>         | <a href="#">560</a>         | <a href="#">7</a>        | <a href="#">0</a>       | <a href="#">23</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">180</a>        |
| <a href="#">2023</a> | <a href="#">247</a>         | <a href="#">634</a>         | <a href="#">6</a>        | <a href="#">0</a>       | <a href="#">10</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">130</a>        |
| <a href="#">2024</a> | <a href="#">6</a>           | <a href="#">19</a>          | <a href="#">0</a>        | <a href="#">0</a>       | <a href="#">0</a>        | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>       | <a href="#">0</a>          |
| Total                | <a href="#">1274</a><br>27% | <a href="#">2523</a><br>54% | <a href="#">52</a><br>1% | <a href="#">0</a><br>0% | <a href="#">62</a><br>1% | <a href="#">0</a><br>0% | <a href="#">0</a><br>0% | <a href="#">2</a><br>0% | <a href="#">0</a><br>0% | <a href="#">0</a><br>0% | <a href="#">765</a><br>16% |

Source: <https://www.cvedetails.com/version-list/1224/19997/1/Google-Android.html> を集計

# → メモリセーフ vs メモリアンセーフ プログラミング言語

Source: 窓の杜  
[Google Security Blog](#)

[Java/Rust/Kotlin のメモリセーフ機能 : C/C++ と安全性を比較してみる](#)  
Source: [IoT Security News](#)

## ニュース

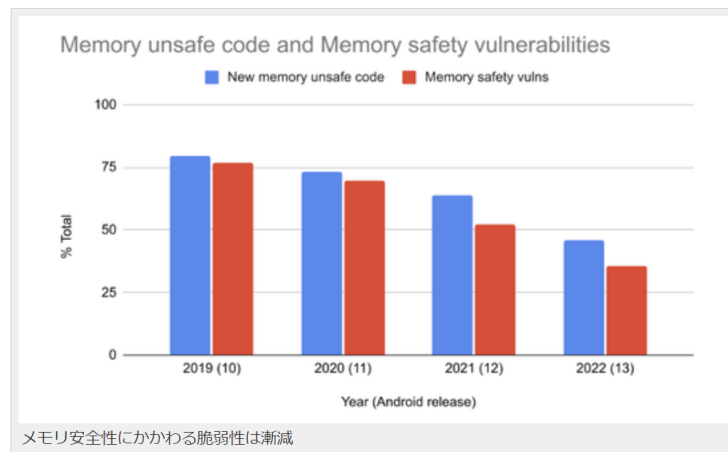
**Rustの採用が進んだ「Android 13」、メモリ破壊バグは減少、脆弱性の深刻度も低下傾向**

2022年はメモリ安全性の脆弱性がAndroidの脆弱性の大部分を占めない最初の年に

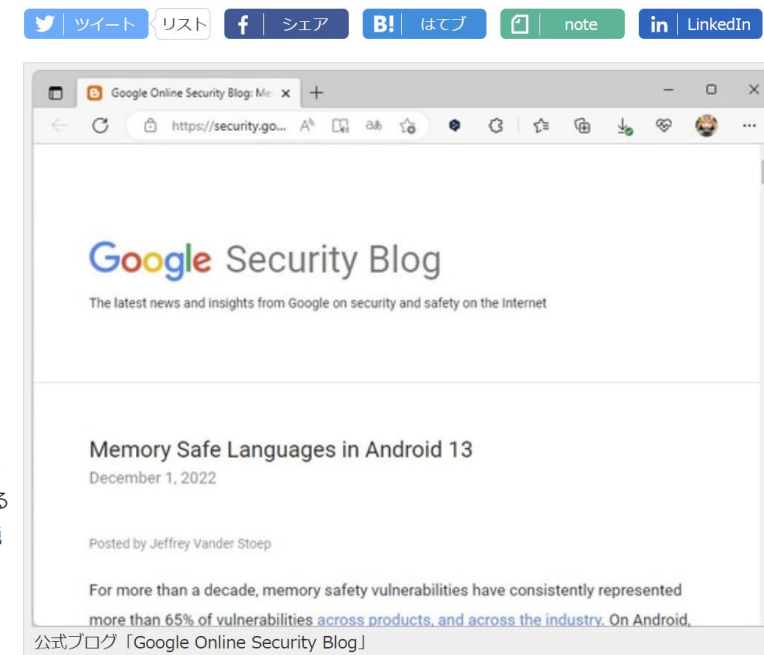
樽井 秀人 2022年12月9日 16:21

## Shift to Memory-Safe Languages Gains Momentum

2022/12/07 DarkReading — 今週に、ソフトウェア・セキュリティの専門家たちが発表したところによると、リモートからの悪用が可能で、野放し状態での攻撃の大部分に関与している、きわめて深刻な脆弱性のグループに対して、ソフトウェア業界は前進しているようだ。この脆弱性グループとは、いわゆるメモリ安全性の問題である。具体的には、バッファオーバーフロー/ユースアフターフリーなどを含むものであり、ソフトウェア会社が公表するアプリケーション・セキュリティ問題の大半を占めるものだ。今回の最新データでは、Java/C#/Rustなどの、メモリ安全性の高い言語の使用が増加したことで、このクラスの脆弱性全体が急速に減少していることが示されている。

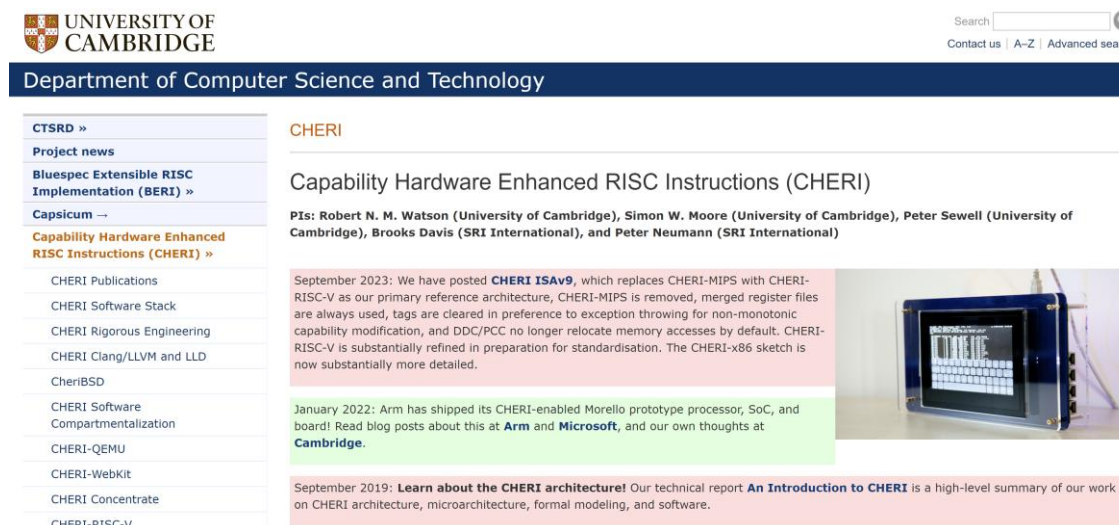


そして重要なのは、現在までのところ、AndroidのRustコードで発見されたメモリ安全性の脆弱性はゼロだということだ。Rustコードにも非Rustコードと対話するために安全ではない部分があり、追加で精査しなければならないこともあるが、純粋なC/C++実装よりもはるかに安全であるという。



# → CHERI (Capability Hardware Enhanced RISC Instructions)

- 英DSIT下のDSbDや米DARPA等が約14年、  
£300Mを投じて英ケンブリッジ大が中心と  
なって開発した**セキュリティ強化型の命令  
セットアーキテクチャ**
  - <https://www.cl.cam.ac.uk/research/security/ctsrld/cheri/>
  - ハードウェア上で実装された能力(capability)  
に基づくメモリ保護を提供し、セキュリティの  
向上を目指す
  - 最も脆弱なC/C++ソースの救済**



UNIVERSITY OF CAMBRIDGE  
Department of Computer Science and Technology

CTSRD »  
Project news  
Bluespec Extensible RISC Implementation (BERI) »  
Capsicum →  
Capability Hardware Enhanced RISC Instructions (CHERI) »

CHERI

## Capability Hardware Enhanced RISC Instructions (CHERI)

PIs: Robert N. M. Watson (University of Cambridge), Simon W. Moore (University of Cambridge), Peter Sewell (University of Cambridge), Brooks Davis (SRI International), and Peter Neumann (SRI International)

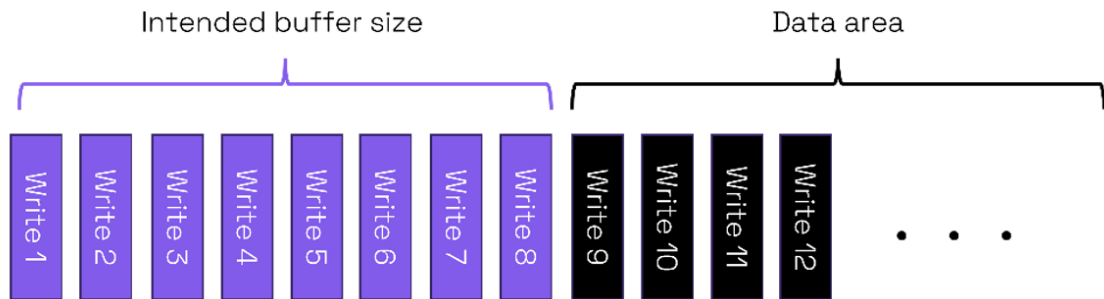
September 2023: We have posted **CHERI ISA v9**, which replaces CHERI-MIPS with CHERI-RISC-V as our primary reference architecture, CHERI-MIPS is removed, merged register files are always used, tags are cleared in preference to exception throwing for non-monotonic capability modification, and DDC/PCC no longer relocate memory accesses by default. CHERI-RISC-V is substantially refined in preparation for standardisation. The CHERI-x86 sketch is now substantially more detailed.

January 2022: Arm has shipped its CHERI-enabled Morello prototype processor, SoC, and board! Read blog posts about this at **Arm** and **Microsoft**, and our own thoughts at **Cambridge**.

September 2019: **Learn about the CHERI architecture!** Our technical report **An Introduction to CHERI** is a high-level summary of our work on CHERI architecture, microarchitecture, formal modeling, and software.

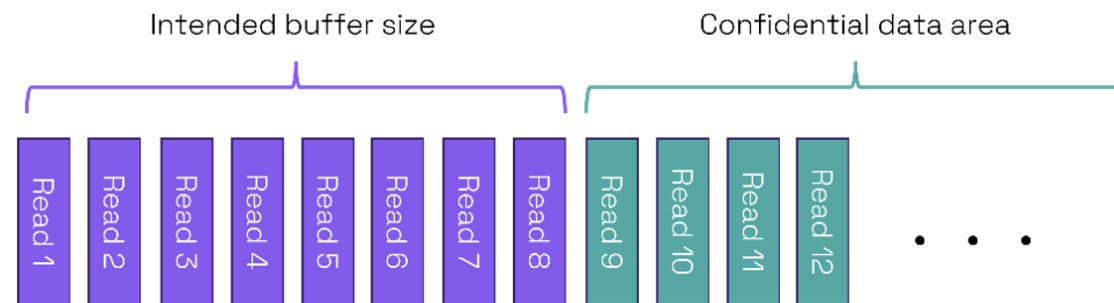
DSIT: [Department for Science, Innovation and Technology](#) / 英国科学・イノベーション・技術省  
DSbD: [Digital Security by Design](#)  
DARPA: [Defense Advanced Research Projects Agency](#)

# → バッファ オーバーフロー...



Data corrupted by  
buffer overflow

オーバーフロー



Confidential data accessed  
by buffer over-read

オーバーリード



Malicious code injected  
by buffer overflow

悪意コード実行

## 過去のサイバー攻撃:

- 1988: Morris Worm – sendmail, rsh/rexec
- 2003: SQL Slammer – MS SQL server
-  **2014: Heartbleed - OpenSSL**
- 2018: Adobe Flash Player – Win, Mac, Linux and ChromeOS
- 2019: WhatsApp VOIP - smartphones

Source: [Comparitech](https://www.comparitech.com/blog/information-security/heartbleed-openssl/)

# → CHERIのコンセプト ハードウェアによる保護

- CHERIはCHERI-ISAに適応したプロセッサを必要とする

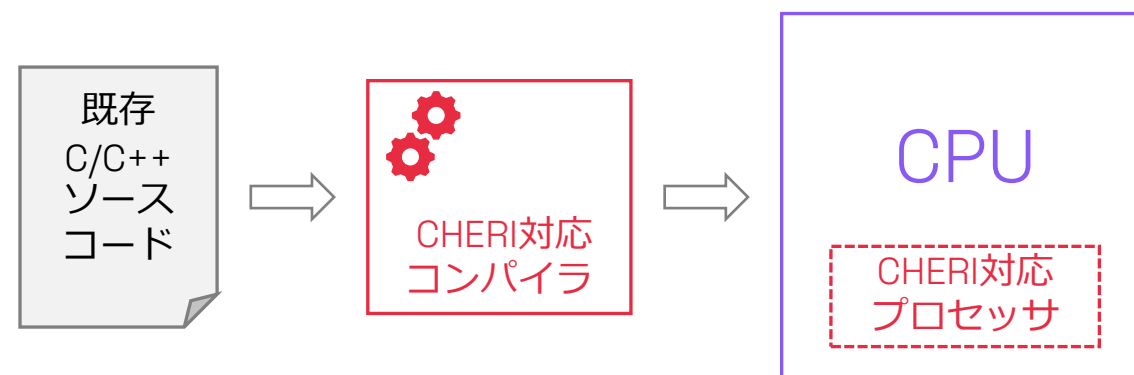
- 多様なコアを開発可能
- ハードウェアによるセキュリティ
- ソフトウェアによるバイパス不可能
- 実証済み

- 既存コードの再利用

- 必要なコード変更はゼロか最小限
- どの部分を保護するかを選択する

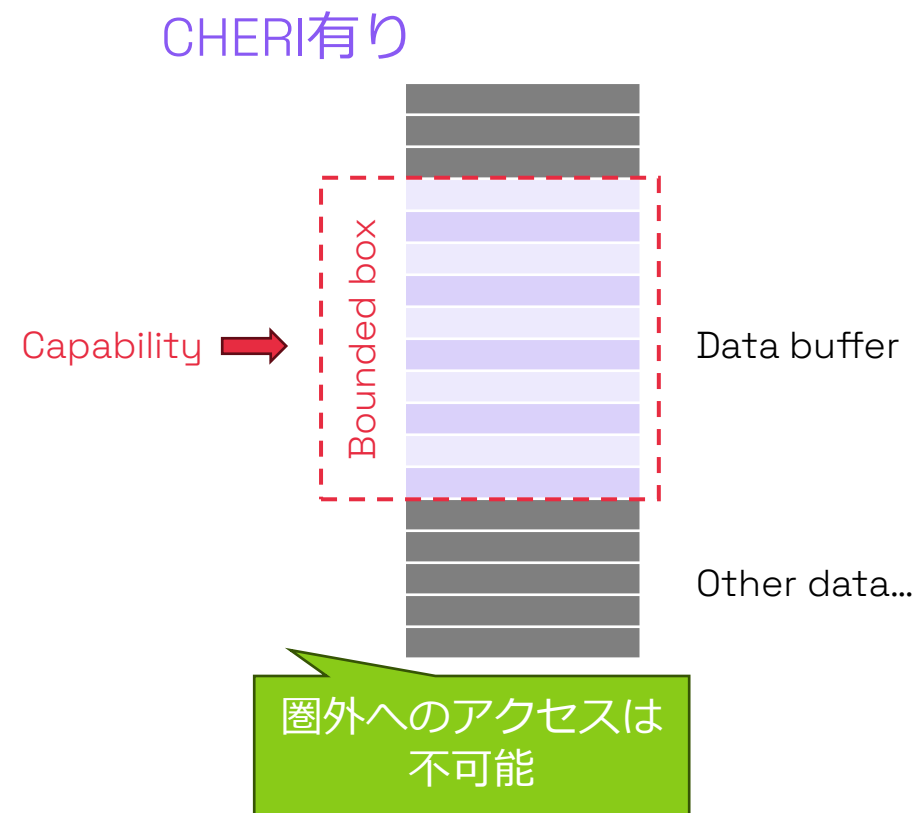
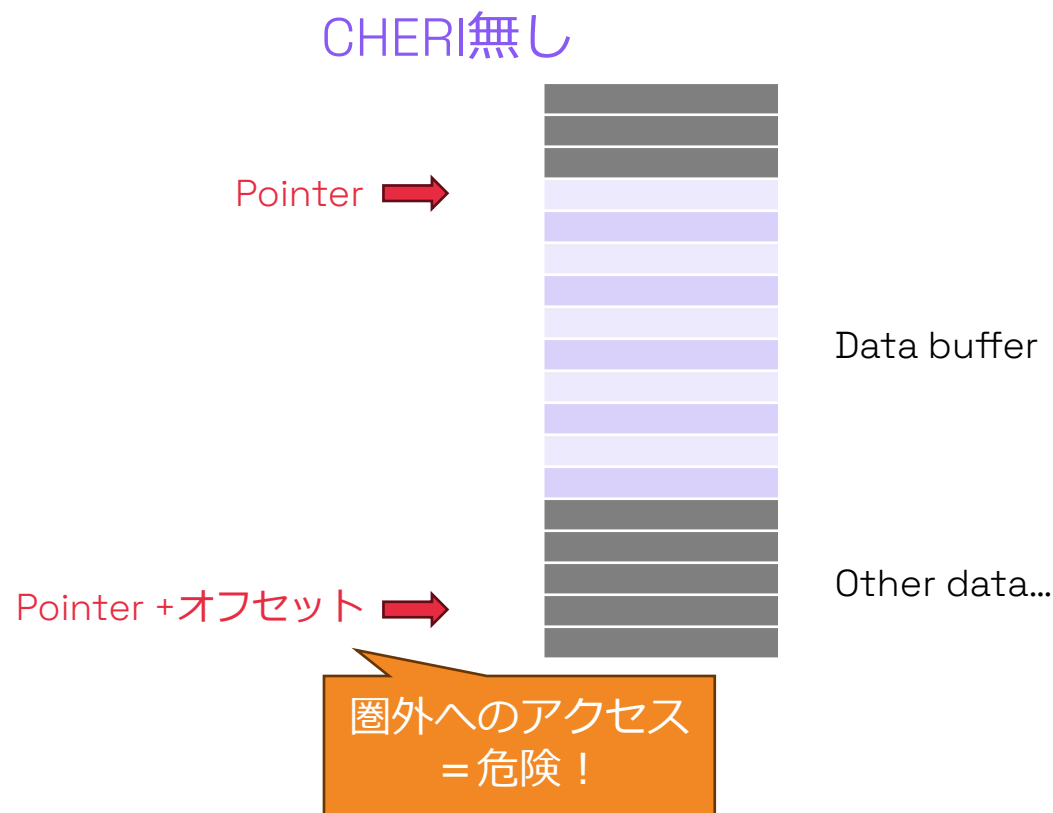
- CHERIのメリット

- 危険なコードは拒否
- クリティカルなコード用にCHERIコンパートメントを作成する
- Arm、MIPS、x86など、さまざまなアーキテクチャに対応



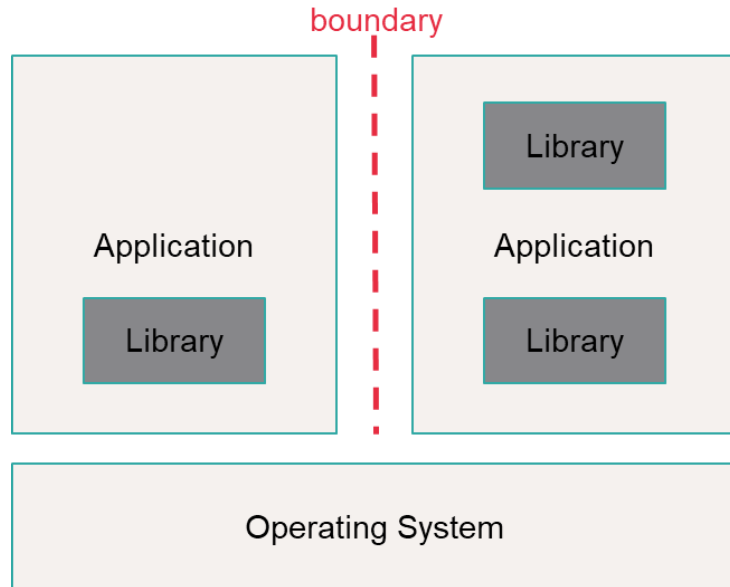
# → CHERIのコンセプト メモリ空間の安全性

- PointerをCapabilityに置き換える - ハードウェア制御によって

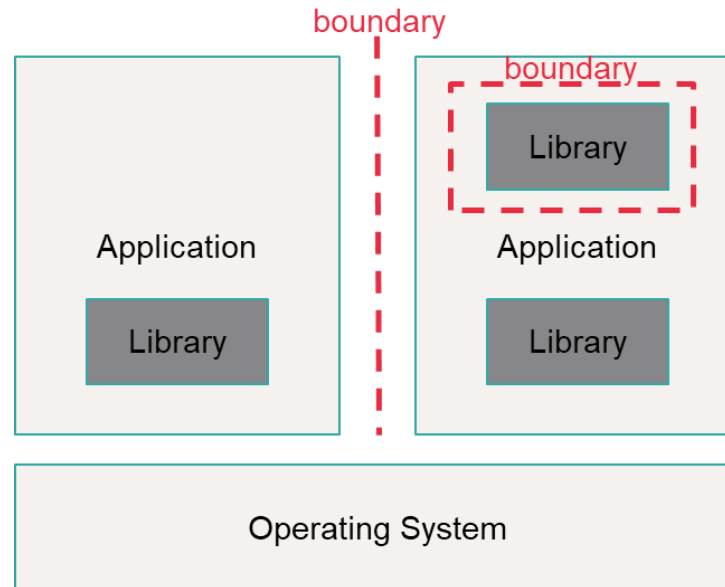


# → コンパートメント化

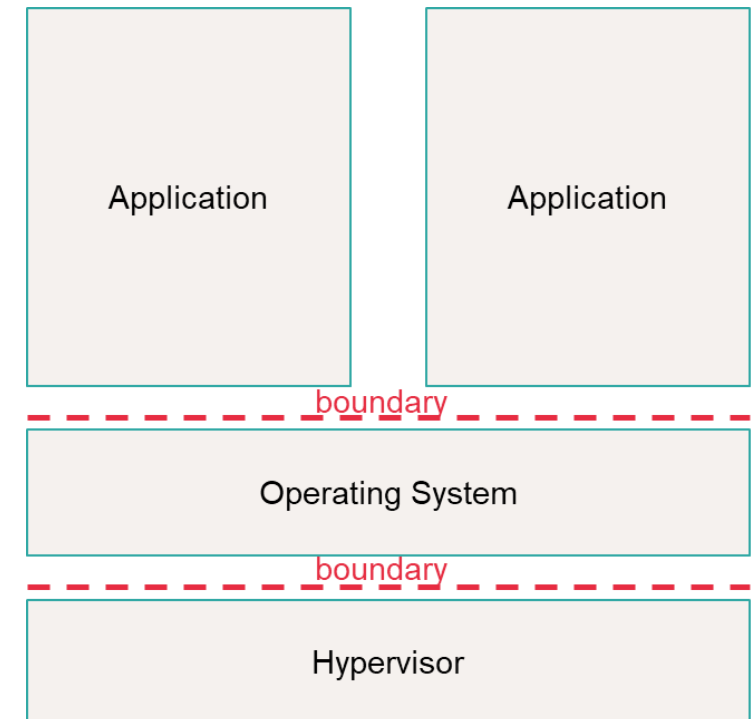
## 垂直



## 垂直 + 特定部分



## 水平



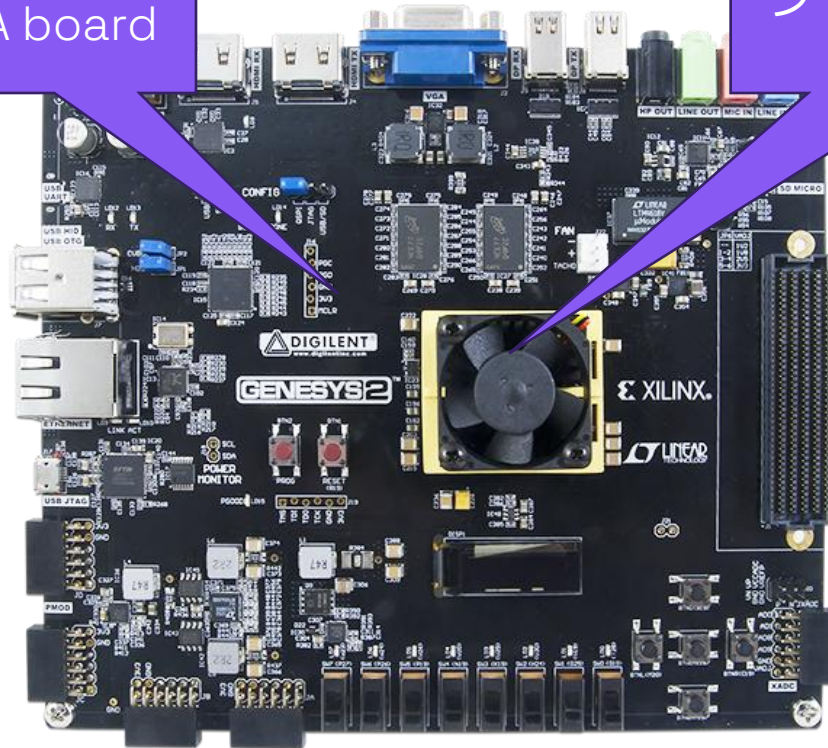


↓  
CHERIデモ

# → Codasip ChERIデモ: 環境

Digilent  
Genesys 2  
FPGA board

Codasip  
CHERI対応IPコアと  
プラットフォーム

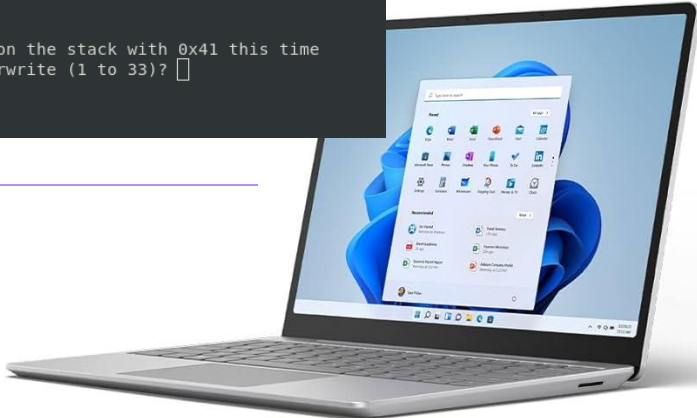


```
Now let's try a buffer overflow with CHERI ENABLED...
We will reset the stack arrays back to their previous contents
(but still keep the corrupted canary value).

Stack contents:
20017e80:b0 f2 00 20 00 00 00 00 84 b1 45 05 00 00 7d a5
20017e90:a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 50
20017ea0:41 53 53 57 4f 52 44 00 21 21 21 21 21 21 21
20017eb0:48 ea 00 20 00 00 00 00 4c aa 95 06 00 00 7d ff
20017ec0:3f 7f 01 20 00 00 00 00 3b 7f d2 07 00 20 fd ff
20017ed0:01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20017ee0:03 84 00 20 00 00 00 00 07 04 05 05 00 00 15 ff
20017ef0:54 0b 00 20 00 00 00 00 00 00 00 08 00 20 ff ff
20017f00:4f 7f 01 20 00 00 00 00 4b bf d7 07 00 20 fd ff

Secret password on the stack: PASSWORD
Canary value: 0x2121212121212121

We will overwrite the byte array on the stack with 0x41 this time
How many bytes do you want to overwrite (1 to 33)?
```



## → CHERIデモのゴール

- デモの着目ポイント:

- 攻撃者がスタックを破損し、重要なデータを変更
- スタック破壊保護 (スタック・カナリア) が無力であることを確認
- CHERIの優れたアクティブなランタイム・メモリ保護を確認

## → CHERIデモ: ソフトウェア

- 右に示すCコードを使用して、demo()関数内でスタック上に2つのローカル配列を作成する
- 注：このデモは、一般的なソフトウェア・セキュリティの慣行に沿った「強力なスタック・スマッシング保護」オプションでコンパイルされている

```
int demo(void)
{
    printf("CHERI Demonstration¥n¥n");

    // Add secret password to stack
    char password[9];

    // Create byte array on stack
    char array[16];
    ...
}
```

# → CHERIデモ: (デモ画面1)

```

CHERI Demonstration

This demonstration adds two things onto the stack:
  1. A character array containing a secret password ("PASSWORD")
  2. A byte array of 16 bytes before it (filled with 0xa5)

We assume the attacker can overflow the byte array.

NOTE: the stack protector software protection IS ENABLED
      The code is compiled with -fstack-protector-strong
      You can see the "canary" value 0xadbaadbaadbaadba

Stack contents:
20017e80:b0 f2 00 20 00 00 00 00 84 b1 45 05 00 00 7d a5
20017e90:a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 50
20017ea0:41 53 53 57 4f 52 44 00 ba ad ba ad ba ad ba ad
20017eb0:48 ea 00 20 00 00 00 00 4c aa 95 06 00 00 7d ff
20017ec0:3f 7f 01 20 00 00 00 00 3b 7f d2 07 00 20 fd ff
20017ed0:01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20017ee0:03 84 00 20 00 00 00 00 07 04 05 05 00 00 15 ff
20017ef0:54 0b 00 20 00 00 00 00 00 00 00 08 00 20 ff ff
20017f00:4f 7f 01 20 00 00 00 00 4b bf d7 07 00 20 fd ff

Secret password on the stack: PASSWORD
Canary value: 0xadbaadbaadbaadba

Now we will do a buffer overflow attack with CHERI DISABLED.
We will overwrite the byte array on the stack with 0x21 bytes
This is the ASCII character '!'
How many bytes do you want to overwrite (1 to 33)? 

```

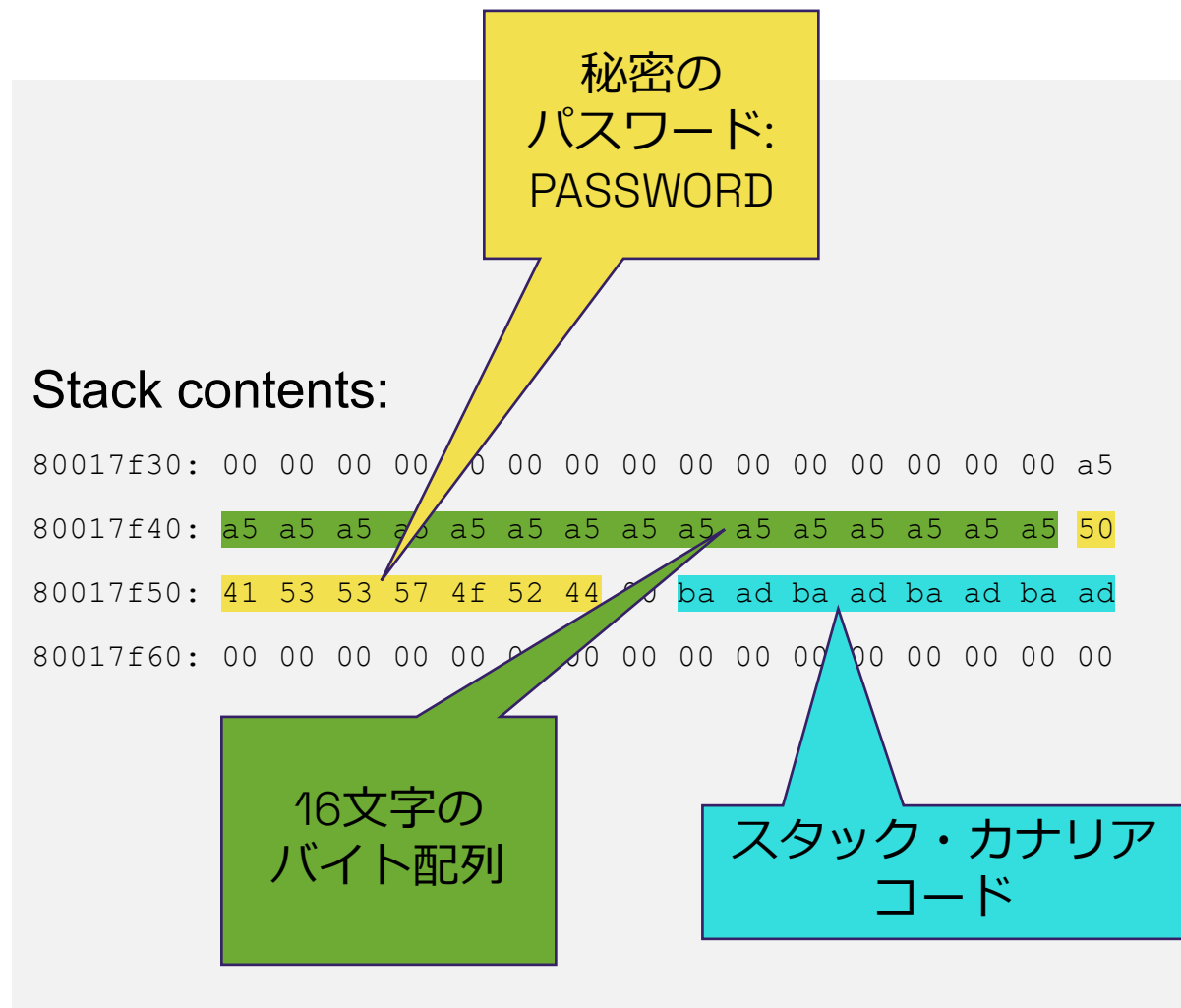
16文字の  
バイト配列

秘密の  
パスワード :  
PASSWORD

スタック・カナリア  
コード

# → CHERIデモ: ソフトウェア – 起動時

- コンパイラはスタック上に2つのメモリ領域を確保する
  - 9文字のパスワード
  - 16文字のバイト配列
- 16進数の50 41 53 53 57 4f 52 44は、文字のASCII値である
  - ‘P’, ‘A’, ‘S’, ‘S’, ‘W’, ‘O’, ‘R’, ‘D’
- “スタック・スマッシング保護”を有効にしているため、コンパイラはパスワードの後に64ビットのカナリア値を自動挿入
  - 0xadbaadbaadbaadba



## → CHERIデモ: CHERIをDISABLE

- バッファオーバーフロー攻撃のように、バイト配列の末尾を越えて書込む
  - 16バイト以下の書き込みは、バイト配列を上書きするだけである
  - これを超えて書き込むと、パスワードとカナリア値を破損する可能性がある

Now we will do a buffer overflow attack with CHERI **DISABLED**.

We will overwrite the byte array on the stack with 0x21 bytes This is the ASCII character '!'

How many bytes do you want to overwrite (1 to 33)?

-----

CHERIを**無効**にしてバッファオーバーフロー攻撃を行う

スタック上のバイト配列を0x21バイトで上書き  
0x21はASCII文字の「!」

何バイト上書きしますか (1~33) ?

ここでは**33**バイト書き込み

# → CHERIデモ: CHERIをDISABLE (デモ画面2)

```
How many bytes do you want to overwrite (1 to 33)? 33

Stack contents:
20017e80:b0 f2 00 20 00 00 00 00 84 b1 45 05 00 00 7d 21
20017e90:21 21 21 21 21 21 21 21 21 21 21 21 21 21 21
20017ea0:21 21 21 21 21 21 21 21 21 21 21 21 21 21 21
20017eb0:48 ea 00 20 00 00 00 00 4c aa 95 06 00 00 7d ff
20017ec0:3f 7f 01 20 00 00 00 00 3b 7f d2 07 00 20 fd ff
20017ed0:01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20017ee0:03 84 00 20 00 00 00 00 07 04 05 05 00 00 15 ff
20017ef0:54 0b 00 20 00 00 00 00 00 00 08 00 20 ff ff
20017f00:4f 7f 01 20 00 00 00 00 4b bf d7 07 00 20 fd ff

Secret password on the stack: !!!!!!!
** Attacker now controls password **

Canary value: 0x2121212121212121

Note that stack protector didn't detect the attack.
Even though the stack protector canary is corrupted.
It only checks the canary when we return from a function.
Canary's current value is : 0x2121212121212121.
Canary's expected value is : 0xadbaadbaadbaadba.
An attacker can use the corrupt values up to that point
and still do damage to the system.

Press a key to continue...□
```

33バイトの  
上書き

パスワードは  
攻撃者の手に  
渡った

スタック・カナリア  
コードは上書きされた

## → CHERIデモ: CHERIをENABLE

- CHERIを有効にして  
再度バッファオーバーフローを試す
- スタック配列をリセットして、  
以前の内容に戻す
  - しかし、破損したカナリアはそのまま
- 上書きするバイトの値は0x41

We will overwrite the byte array on the stack with 0x41 this time

How many bytes do you want to overwrite (1 to 33)?

----

スタック上のバイト配列を0x41で上書きする  
何バイト上書きしますか (1~33) ?

ここでは20バイト書き込み

# → CHERIデモ: CHERIをENABLE (デモ画面3)

Now let's try a buffer overflow with **CHERI ENABLED**...  
We will reset the stack arrays back to their previous contents  
(but still keep the corrupted canary value).

Stack contents:

```
20017e80:b0 f2 00 20 00 00 00 00 84 b1 45 05 00 00 7d a5
20017e90:a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 a5 50
20017ea0:41 53 53 57 4f 52 44 00 21 21 21 21 21 21 21 21
20017eb0:48 ea 00 20 00 00 00 00 4c aa 95 06 00 00 7d ff
20017ec0:3f 7f 01 20 00 00 00 00 3b 7f d2 07 00 20 1d ff
20017ed0:01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20017ee0:03 84 00 20 00 00 00 00 07 04 05 05 00 00 15 ff
20017ef0:54 0b 00 20 00 00 00 00 00 00 00 08 00 20 ff ff
20017f00:4f 7f 01 20 00 00 00 00 4b bf d7 07 00 20 fd ff
```

Secret password on the stack: **PASSWORD**  
Canary value: **0x2121212121212121**

We will overwrite the byte array on the stack with 0x41 this time  
How many bytes do you want to overwrite (1 to 33)?

秘密の  
パスワード：  
PASSWORD

16文字の  
バイト配列

スタック・カナリア  
コードは  
上書きされたまま

# → CHERIデモ: CHERIをENABLE (デモ画面4)

```

Try again (y/n)? y
How many bytes do you want to overwrite (1 to 33)? 20

+++++
++ EXCEPTION HANDLER: CHERI EXCEPTION ++
+++++

+++++
++ EXCEPTION HANDLER: CHERI EXCEPTION ++
+++++

+++++
++ EXCEPTION HANDLER: CHERI EXCEPTION ++
+++++

+++++
++ EXCEPTION HANDLER: CHERI EXCEPTION ++
+++++

Stack contents:
20017e80:b0 f2 00 20 00 00 00 00 84 b1 45 05 00 00 7d 41
20017e90:41 41 41 41 41 41 41 41 41 41 41 41 41 41 50
20017ea0:41 53 53 57 4f 52 44 00 21 21 21 21 21 21 21
20017eb0:48 ea 00 20 00 00 00 00 4c aa 95 06 00 00 7d ff
20017ec0:3f 7f 01 20 00 00 00 00 3b 7f d2 07 00 20 fd ff
20017ed0:01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20017ee0:03 84 00 20 00 00 00 00 07 04 05 05 00 00 15 ff
20017ef0:54 0b 00 20 00 00 00 00 00 00 00 08 00 20 ff ff
20017f00:4f 7f 01 20 00 00 00 00 4b bf d7 07 00 20 fd ff

Secret password on the stack: PASSWORD
Canary value: 0x2121212121212121

You will have just seen that CHERI stops any writes beyond the array.

Press a key to continue...

```

CHERI が  
バッファオーバーフロー  
を4回検出

これにより、攻撃者は  
パスワードを知ることが  
できない

スタック・カナリア  
コードは  
上書きされたまま

## → CHERIデモ: スタック・カナリアが遅れて動作 (デモ画面5)

```
CHERI provides a much more robust runtime protection  
  
Exiting the function now...  
It is only at this point that -fstack-protector will work  
  
*****  
** Stack-protector: Stack corruption detected **  
*****
```

その後  
スタック・カナリアコードが  
メモリ破壊を検出

## → このデモでCHERIは何を達成できたか？

### • CHERI無しの場合:

- 攻撃者はパスワードを発見し、変更することができる
- システムは攻撃に気づかず、動作を継続する
- 製品が危険にさらされる可能性がある
- スタック・カナリアはバイパスされ、効果がない

### • CHERI有りの場合:

- 攻撃者がパスワードを盗んだり変更することを防ぐ
- 攻撃を検知し、対応策を講じることができる
- 製品は無傷のまま
- スタック・カナリアのような統計的メカニズムは必要ない

### • CHERIは、すべてのメモリアクセスを100%チェックする

CHERIは、業界の高度なメモリ保護の新たなセキュリティコンセプトである

Codasipは、RISC-V IPポートフォリオ全体をCHERI対応します

# 「プログラマ依存のメモリ安全」からの脱却

# → 2025 Spring Agenda

|                        |   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15:40 - 16:00<br>(JST) | 英 | <b>CHERIアライアンス：電子システムに改善されたセキュリティを組み込む</b><br><br>Abstract: CHERIアライアンスは、CHERI技術の採用と普及を通じて、メモリ安全性の問題を軽減することを目的としている。サイバーセキュリティのコストが増大し続ける現代社会において、メモリ安全性に起因する脆弱性は、全脆弱性の3分の2以上を占めている。CHERIは、メモリ・ポインタに境界と許可のメタデータを追加することでこれを解決します。CHERIアライアンスは、メンバーがCHERIの採用を促進、調整、サポートするために協力できる中立的な組織です。<br><br>※ リモートセッションです                                                                                                                                                                                                                                                                                                | <b>Dr Marno van der Maas</b><br><b>Principal Hardware Engineer</b><br><b>lowRISC C.I.C. (英)</b><br><br>Bio: Marno van der Maas 博士は、2022年にケンブリッジ大学でサイドチャネル攻撃から信頼された実行環境を保護する研究で博士号を取得した。この間、Capability Enhanced RISC Instructions (CHERI)と名付けられた技術を使用し、メモリポインタの境界とパーミッションを強制するプロセッサの実装とテストにも取り組んだ。LowRISC CICでは、Sonata CHERIoT開発プラットフォームおよびOpenTitanとSunburst全体の主要ハードウェアエンジニアを務める。検証やセキュリティに関する経験を活かし、オープンシリコンの実現というビジョンに貢献している。 |
| 16:05 - 16:25<br>(JST) | 英 | <b>CHERI - メモリ安全アーキテクチャとコンパートメント化が実用化に向けて本格始動</b><br><br>Abstract: CHERIプロジェクトは、約14年前、ブレイク不可能なケイバリティ・トークンを最新のRISCアーキテクチャと融合させる方法を検討することから始まった。最初のMIPSベースのプロトタイプは、C/C++のポインタをケイバリティにコンパイルできること、効率的な共有でプログラム・コンポーネントを分離できること、さらにパワーズメタデータを効率的に圧縮する方法を教えてくれた。Armとのコラボレーションは、Morelloチップへと発展し現在では、空間的にも時間的にも安全で、すべてのライブラリの自動コンパートメント化を可能にする大規模なデスクトップおよびサーバーも実行できている。ハードウェアでもソフトウェアでも、多くの教訓が得られた。マイクロソフト社のCHERIIoTやCodasip社の700ファミリーを含む、最初の商用CHERI実装と平行して、我々は現在CHERI-RISC-Vの批准に近づいている。本講演では、CHERI のコンセプトと保証の基本、CHERI 実装ファミリーの興味深いマイクロアーキテクチャの特徴、および現在の CHERI-RISC-V の批准に向けた取り組みの詳細について説明します。<br><br>※ リモートセッションです | <b>Dr Jonathan Woodruff</b><br><b>Senior Research Associate</b><br><b>ケンブリッジ大学 (英)</b><br><br>Bio: Jonathan Woodruff博士は、プロセッサアーキテクチャとマイクロアーキテクチャ、および低レベルソフトウェアの最適化を専門とするシニアリサーチアソシエイトです。ケイバリティ・プロセッサ設計を専門とし、キャッシュ階層、コアタイミング、マルチコア設計を含むフルシステム最適化、制御フローインテグリティやプライベート実行を含む主要なセキュリティアプローチの探求に取り組んでいます。                                                                                                                             |
| 16:30 - 16:50<br>(JST) | 英 | <b>CHERI_RISC-Vの標準化</b><br><br>Abstract: CHERIタスク・グループは、RISC-V ISAでCHERIをサポートする拡張機能を提案するために、RISC-V International内に組織された。これらの拡張の動機は、Microsoft's Security Response Center (MSRC)がレポートした、ソフトウェアの脆弱性の70%以上を占めるメモリ安全性の問題に対処して、RISC-Vを強化することである。本プレゼンテーションでは、CHERI TGの現在および将来の作業の概要と、CHERI拡張機能の仕様策定状況について説明する。<br><br>※ リモートセッションです                                                                                                                                                                                                                                                                                | <b>Tariq Kurd</b><br><b>Distinguished Engineer and Lead IP Architect</b><br><b>Codasip UK デザインセンター (英)</b><br><br>Bio: Tariq Kurdは、主に組み込み分野で20年以上にわたってCPUアーキテクチャ、設計、検証に携わってきたVLIW、マルチスレッド、アウトオブオーダーコア、セキュリティ、DSPスタイルのコアに携わった。Codasipには3年以上在籍しており、以前はHuawei UK、Broadcom、Nvidiaに在籍した。これまでTariqは、RISC-V InternationalのZfinx、およびコードサイズ削減タスクグループの議長を歴任した。                                                                            |
| 16:55 - 17:15<br>(JST) | 日 | <b>CHERI - どのように機能するのか？</b><br><br>Abstract: このプレゼンテーションでは、CHERIがCodasip X730 RISC-V Core FPGA評価キットでどのように動作するかストーリーボード形式で紹介します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <b>明石 貴昭</b><br><b>Country Manager – Japan</b><br><b>Codasip GmbH (独)</b><br><br>Bio: 1990年代前半にケイデンス社で日本最初のVerilog-XL FAEとして、Verilog (IEEE std. 1364-1995)標準化、そして2000年代前半にシノプシス社でSystemVerilog (IEEE std. 1800-2005)標準化、UVMの前進であるVMMのエバンジェリスト、JEITA(旧EIA)標準化委員として活動し、約30年以上日本の半導体設計に関わってきました。Codasipの日本カントリー・マネージャーとして4年目となる。                                                                                                   |

→ Videos and PDFs are available.

<https://riscv.or.jp/risc-v-day-tokyo-2025-spring/>

→  
CHERI ALLIANCE 101



**CHERI**

RISC-V Tokyo Day 2025 Spring  
⑤ RISC-V研究とオープン半導体

2025/02/27, 15:30 ~ 17:15



CHERI – Architectural memory safety and compartmentalisation  
Full steam ahead to commercialization

Robert N. M. Watson, Simon W. Moore, Peter G. Neumann, **Jonathan Woodruff**  
Hesham Almatary, Ricardo de Oliveira Almeida, Jonathan Anderson, Alasdair Armstrong, Rosie Baish, Peter Blandford-Baker, John Baldwin, Hadrien Barrel, Thomas Bauereiss, Ruslan Bukin, Brian Campbell, David Chisnall, Jessica Clarke, Nirav Dave, Brooks Davis, Lawrence Esswood, Nathaniel W. Filardo, Franz Fuchs, Dapeng Gao, Ivan Gomes-Ribeiro, Khilan Gudka, Brett Gutstein, Angus Hammond, Graeme Jenkinson, Alexandre Joannou, Mark Johnston, Robert Kovacsics, Ben Laurie, Marno van der Maas, A Theo Markettos, J. Edward Maste, Alfredo Mazzinghi, Alan Mujumdar, Prashanth Mundkur, Steven J. Murdoch, Edward Napierala, George Neville-Neil, Kyndylan Nienhuis, Robert Norton-Wright, Philip Paeps, Lucian Paul-Trifu, Allison Randal, Alex Richardson, Michael Roe, Colin Rothwell, Peter Rugg, Hassen Saidi, Peter Sewell, Thomas Sewell, Stacey Son, Ian Stark, Domagoj Stofa, Andrew Turner, Munraj Vadera, Konrad Witaszczyk, Hongyan Xia, Vadim Zaliva, and Bjoern A. Zeeb

 UK Research and Innovation  
 University of Cambridge and SRI International  
RISC-V Tokyo Summit  
February 27, 2025

Approved for public release; distribution is unlimited. This research is sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under contract FA8750-18-C-0037. The views, opinions, and/or findings contained in this article/representation are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.



27 February 2025



**CHERI**

**The CHERI Alliance**

Embedding improved security in electronic systems

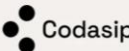
**Dr Marno van der Maas**  
lowRISC – Founding Member of the CHERI Alliance



→  
Let's make a  
standard for CHERI-  
RISC-V

To make memory safety available for everyone,  
from small cores to high performance

Tariq Kurd, Chief Architect at Codasip, Based in Bristol UK  
RISC-V Day Tokyo, February 2025



# Join the Alliance now!



Demonstrate  
security  
leadership



Network, learn  
and get visible



Accelerate  
adoption of  
CHERI



Save on costs  
(learning, promotion,  
development, ...)



Collaborate  
with the  
community

# ○ Member benefits



## For all members

- ◆ Demonstrate security leadership
- ◆ Vote in elections, get elected
- ◆ Join committees / working groups
- ◆ Drive alignment and standards
- ◆ Leverage joint open-source efforts
- ◆ Early access to CHERI code
- ◆ Network and exchange
  - ◆ Member-only address book
  - ◆ Member-only content, reports, info
- ◆ Accelerate adoption of CHERI
- ◆ Benefit from CHERI marketing
- ◆ Free visibility on CHERI Alliance Website
  - ◆ Member page
  - ◆ Blog
- ◆ Free visibility at events \*
  - ◆ Speaking slots, webinars
  - ◆ Booth, demo
- ◆ Show product compliance \*\*

# ○ Organized by the CHERI Alliance

## CHERITech

- ◆ Technical focus
  - ◆ Free entry (even non-members)
  - ◆ Talks, demos, workshops
  - ◆ Research updates
- ◆ Next edition
  - ◆ CHERITech'2025
  - ◆ 14<sup>th</sup> November 2025
  - ◆ Manchester

## CHERI Blossoms Conference

- ◆ Main event
  - ◆ Free entry (even non-members)
  - ◆ Talks, demos, networking
  - ◆ Published online
- ◆ Next edition
  - ◆ CHERI Blossoms Conf 2026
  - ◆ 26-27<sup>th</sup> March 2026
  - ◆ Cambridge



# CHERI

---

# THANK YOU

Contact [contact@cheri-alliance.org](mailto:contact@cheri-alliance.org)

Web [www.cheri-alliance.org](http://www.cheri-alliance.org)

→  
Thank you!

[takaaki.akashi@codasip.com](mailto:takaaki.akashi@codasip.com)  
[contact\\_japan@codasip.com](mailto:contact_japan@codasip.com)