Delegated using systems and application software while supporting gradual adoption of its novel friendly, high-performance and incrementally adoptable hardware/software platform: as granularity increases, rights delegated to individual sandboxes decrease.

Conventional "fetch" program – utilization of increasing numbers of vulnerable logic principle of least to mitigate inevitable bounds checking. References


Capabilities and the application compartmentalization motivation

Programs running in application compartmentation to mitigate exploitable vulnerabilities: software is decomposed into sandboxed components, each with the right to operate in isolation. This exploits the principle at base privilege: as granularity increases, rights delegated to individual sandboxes decrease. As vulnerabilities are exploited, only the rights of the affected component are lost, forcing attackers to exploit many more vulnerabilities to accomplish the same goals.

The Capsicum hybrid capability model developed by Cambridge and Google/Berkeley provides a dynamic capability-based system, with compartments as a foundational concept. The principle is straightforward: software is decomposed into interdependent components, each with different roles and capabilities. The components are typically processes or services, and the interactions between them are mediated by capabilities. Capabilities are tokens that encode security labels and access control policies, allowing fine-grained control over resource access. The model provides a framework for designing secure systems, enabling flexible and scalable resource management. The Combustion architecture integrates this capability model with virtualization technologies and hardware extensions to support compartmentalization.

Security-enabled Cap and LLVM

We have extended the Combustion architecture to support the new security model. The Combustion architecture includes a variety of features and application compartmentalization. Rights are encoded in capabilities, which are tokens that carry security labels and access control policies. Capabilities provide a fine-grained way to manage access to resources, allowing for more precise control over what operations are allowed. The Combustion architecture integrates this capability model with virtualization technologies and hardware extensions to support compartmentalization.

The CHERRI-enabled Cap and LLVM

The CHERRI-enabled Cap and LLVM is a research project that aims to improve the security and performance of software running on modern systems. The project focuses on developing a new virtualization technology that enables fine-grained control over resource access, providing a way to segregate and protect different parts of the system. The CHERRI-enabled Cap and LLVM uses a combination of hardware and software components to achieve this goal. The project is led by researchers from the University of Cambridge and collaborating institutions. The long-term goal of the project is to develop a secure and efficient virtualization platform that can be used in a wide range of applications, from consumer electronics to data centers. The CHERRI-enabled Cap and LLVM is an ongoing research project, with active development and collaboration taking place. The project is open to collaboration and contributions from the broader academic and industrial communities. The University of Cambridge has previously published research on virtualization and security, and has a strong track record of innovation in these areas. The project is expected to have significant implications for the future of virtualization and security, and could lead to new applications and technologies that improve the security of modern computing systems.