Traceless Execution Support for Privacy Enhancing Technologies

Aluno: Daniela Lopes
Orientador: Nuno Santos
Instituto Superior Técnico

Why people use PETs

Censorship

Lack of privacy

Surveillance

Secure Messaging

VPNs

Anonymizing networks

Anti tracking

Many people are afraid of using PETs

Forensic inspection of devices + **Current PETs are not designed to remain unobservable during forensic investigations**

- Journalists
- Whistle blowers
- Travelers crossing border control

Many people fear the consequences of using PETs

Goals

Environment to execute PETs

Prevent observable persistent traces

Maintain a PET’s functionality and performance

Contents

1. Introduction and motivation
2. **Case study: Tor browser**
3. Related work
4. Proposed solution
5. Conclusions

Identify implications of using PETs

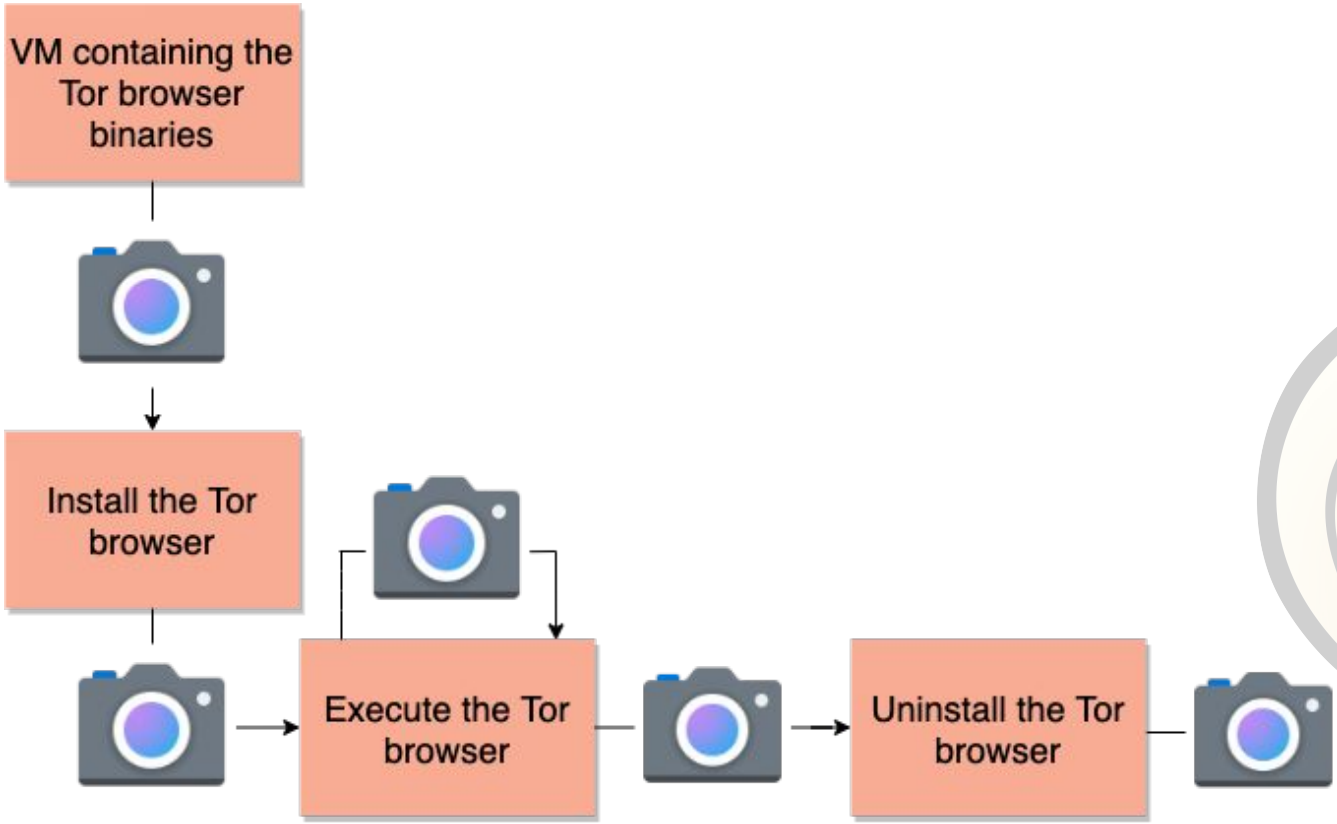
Understand the extension and nature of persistent traces left

The Tor browser is a widely used and complex PET

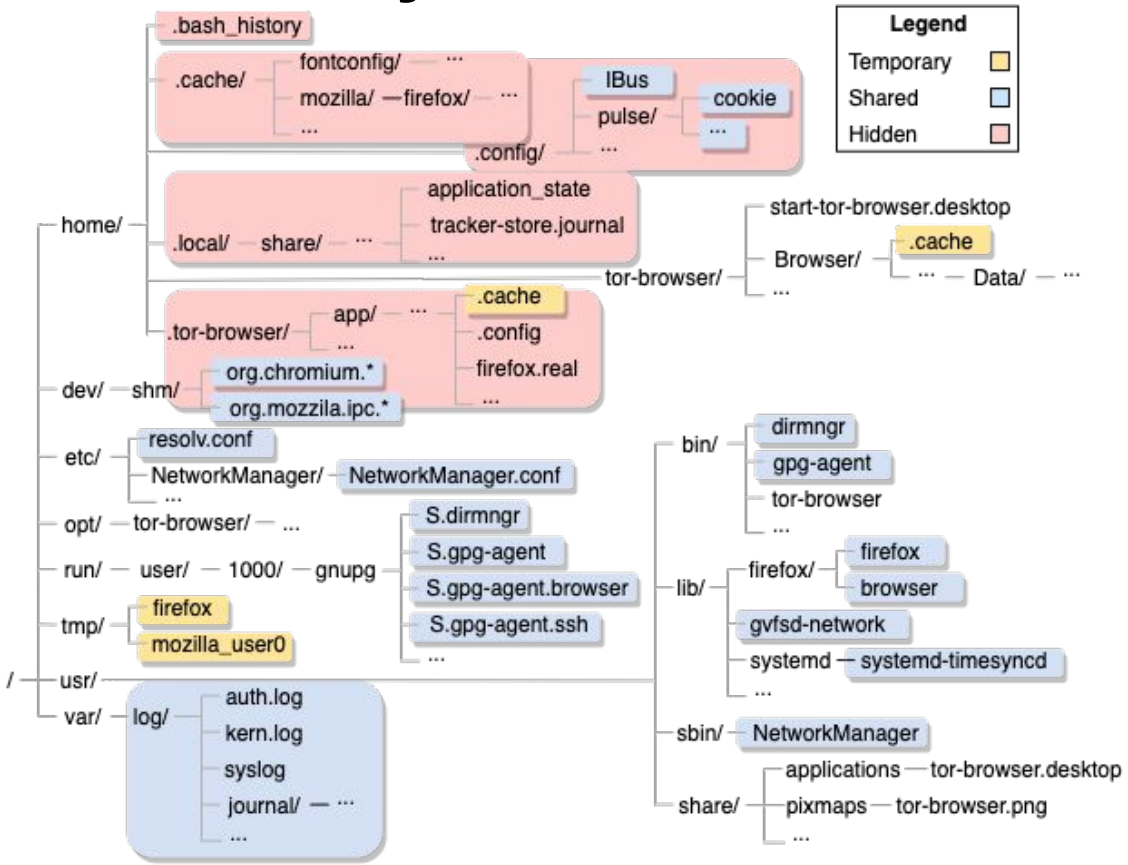
↓

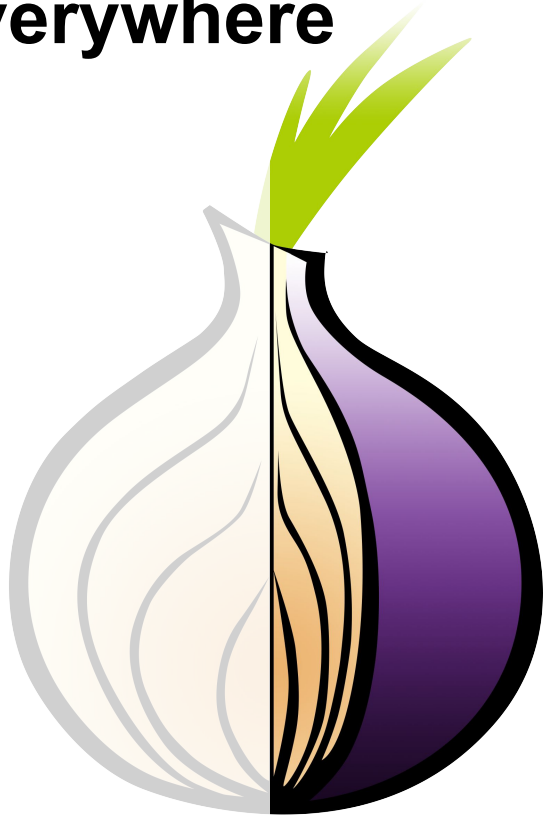
Perform empirical study on the Tor browser to potentiate our solution

Case study: methodology



Case study: Tor leaves traces everywhere





Case study: Tor artifacts can be recovered through forensics

Temporary files	→	File carving to recover files from unallocated space
Shared files	→	Analysis of patterns in the access of files by examining timestamps of files that are already known to be frequently accessed by the application
Hidden files	→	Direct observation of the files in the system

Contents

1. Introduction and motivation
2. Case study: Tor browser
3. **Related work**
4. Proposed solution
5. Conclusions

Digital forensic techniques

Versioning, logging and monitoring

Saving the state of a system in different points in time, allowing to identify the differences between multiple states

Memory and kernel analysis

Parsing and interpretation of kernel data structures to infer information in memory

Persistent data analysis

Analysis of data extracted from storage devices

Anti-forensic countermeasures

Hybrid systems

Focus on hiding flows of data within the components of a machine

Hiding data accesses

Conceal data accesses to prevent identifying patterns

Deniable steganographic file systems

Deniable data hiding on persistent storage

Artifice: overview

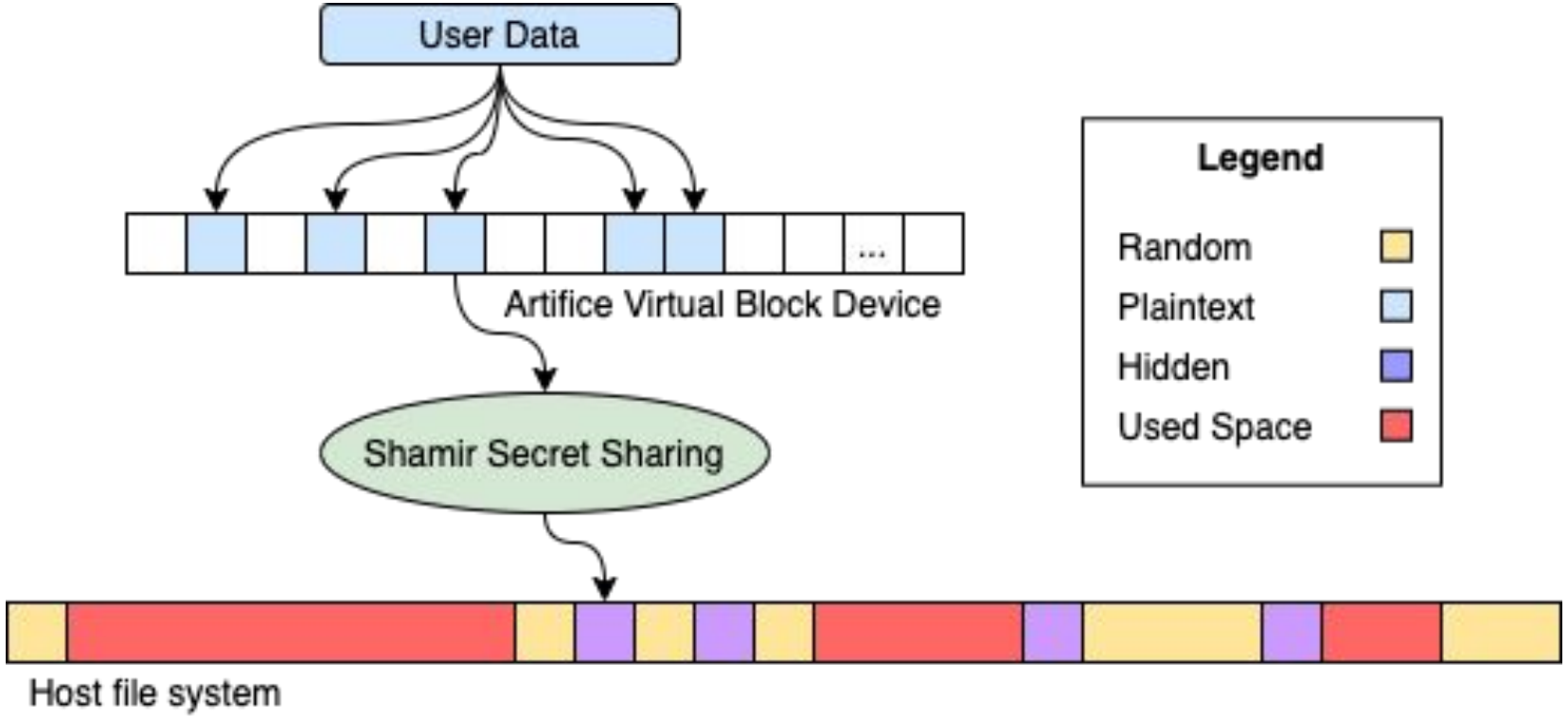
Protect users in life-threatening situations

+

Provide a deniable way to store sensitive data

Adversary is unable to distinguish blocks encoding hidden data from blocks with random data

Artifice: encoding user data



Artifice is not deniable

- Pseudo-random blocks change the bit entropy of the disk
- Dependency on external hardware components

Artifice is not deniable!

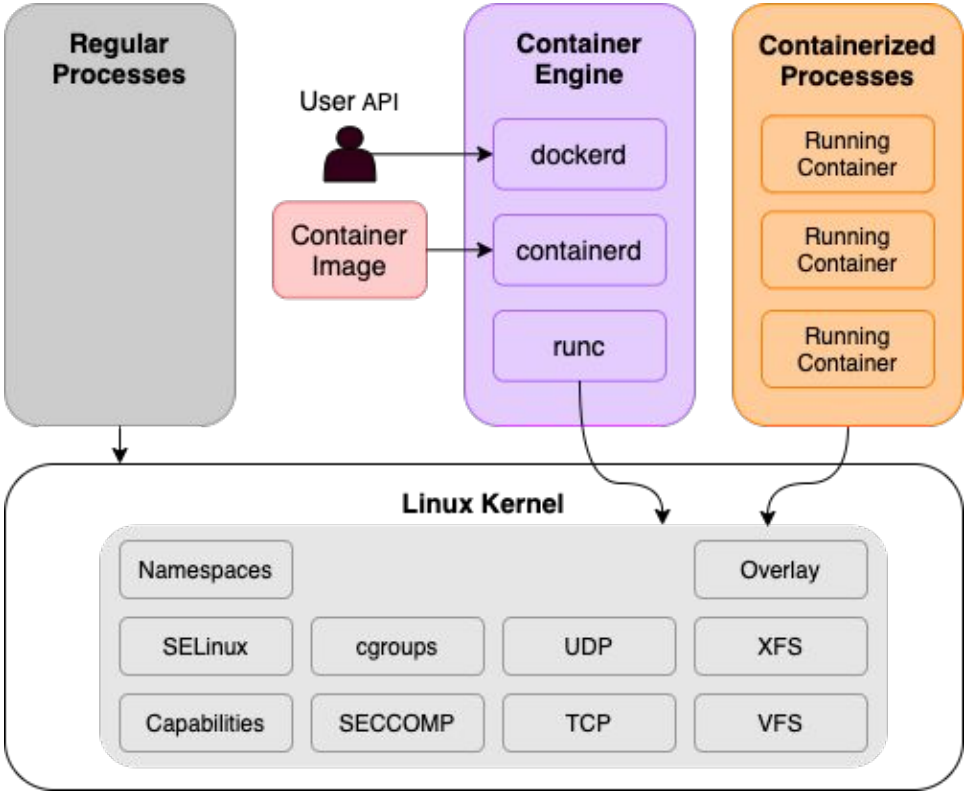
Why we chose to use containers

Traces left by executing PETs are too extensive

Artifice booting another OS for isolation will result in additional traces

Minimize the overhead of executing PETs

Containers: overview



Research around containers

Security and Linux native support mechanisms

Cloud computing and multi-tenancy

Enhancing existing security mechanisms for specific use cases

None target the deniable execution of PETs

Contents

1. Introduction and motivation
2. Case study: Tor browser
3. Related work
4. **Proposed solution**
5. Conclusions

Proposed solution: Calypso

Shadow drive

- Secondary file system that uses the unused blocks of the native file system

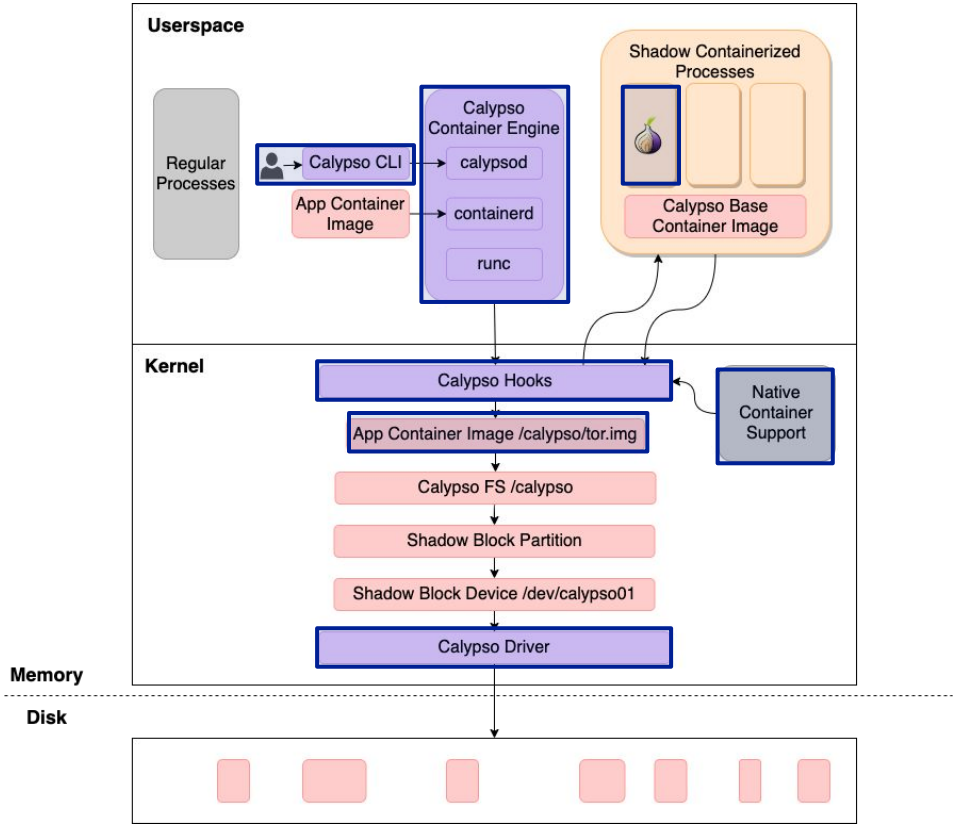
+

Shadow Containers

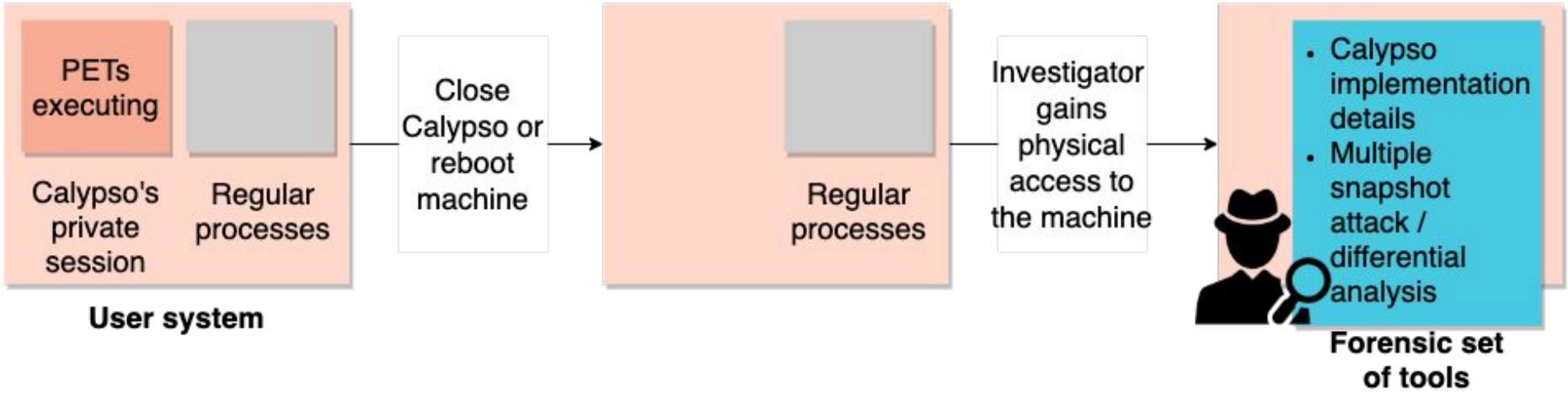
- Isolated environment to execute programs deniably

No observable changes are made to the persistent state

Architectural components



Threat model



Obstacles

Traceless bootstrap

- Launched outside the container environment
- Point of failure that can compromise the system's deniability

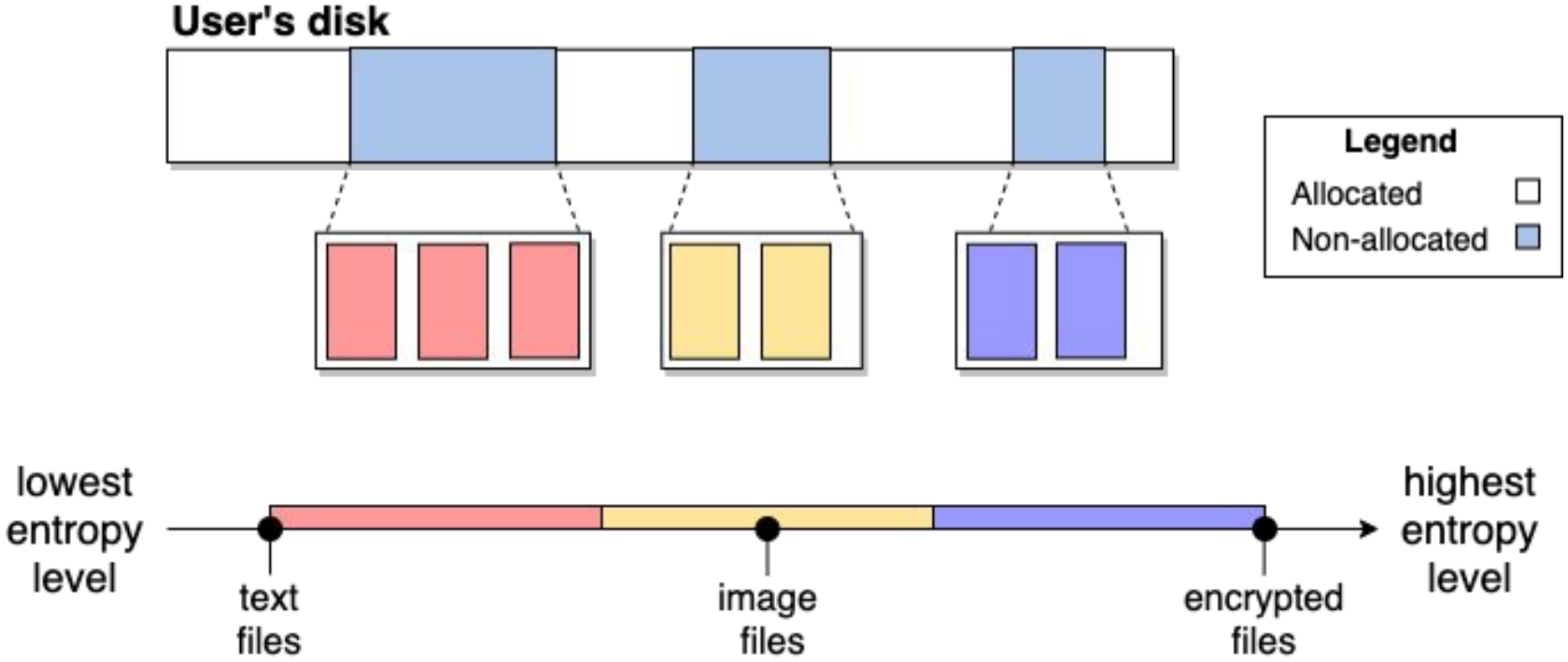
Obfuscated storage

- Finding and maximizing usable space
- Encoding data within the blocks while maintaining the entropy of the disk
- Fault tolerance and data loss

Isolated environment

- From the container engine to the native file system
- From the containers to the host
- Other challenges:
  - Swapping
  - Journaling and OS management
  - Hibernation

Profiling the user's disk



Evaluation

1. **Functionality**
2. **Performance**
3. **Efficiency**
4. **Fault tolerance and data resilience**
5. **Plausible deniability:**
    a. Isolation
    b. Non-observability

Contents

1. Introduction and motivation
2. Case study: Tor browser
3. Related work
4. Proposed solution
5. **Conclusions**

Conclusions

State-of-the-art PETs are vulnerable to forensic analysis

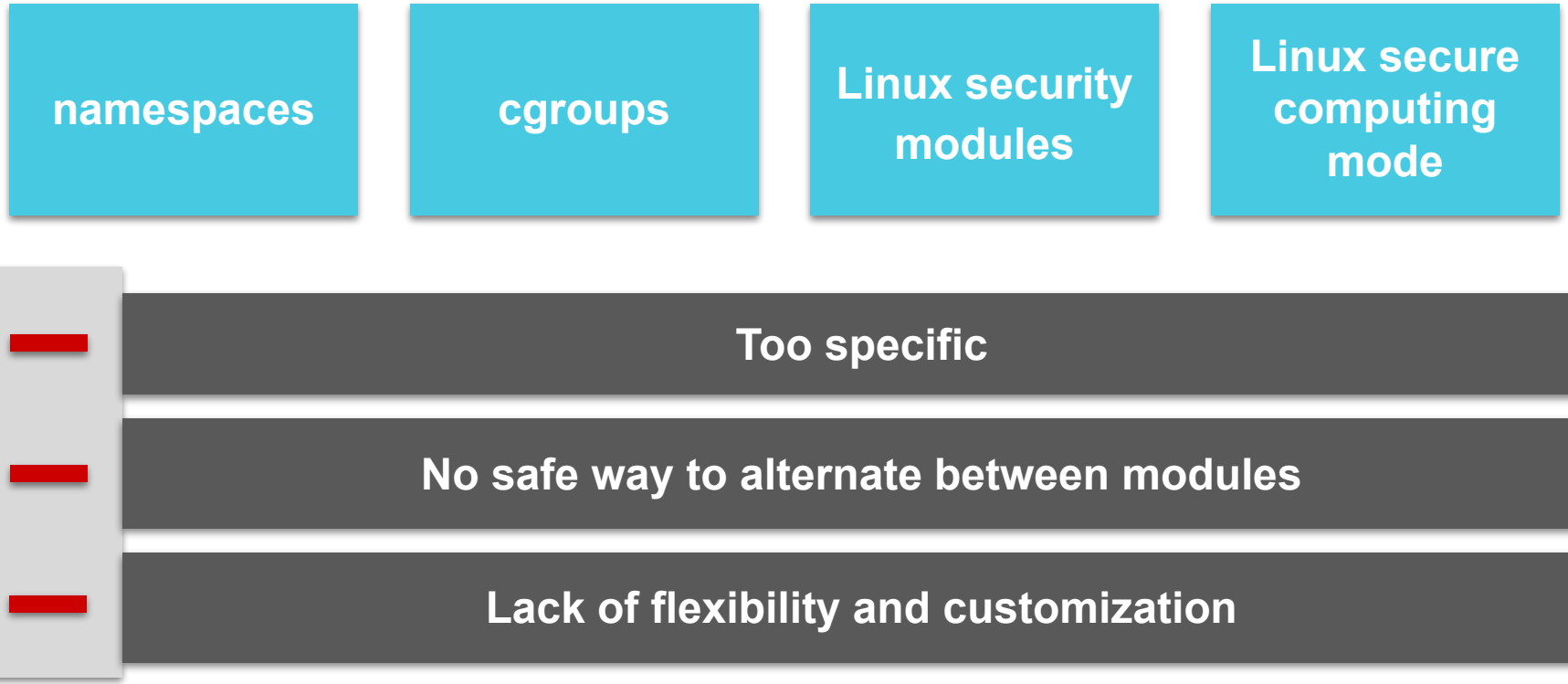
Calypso offers isolation and maintains the entropy of the disk

Provide a traceless execution environment to increase PET usage by censored users

Give protection against forensic analysis

The end

Containers: why they are not enough



Why we need to offer users a deniable way to execute PETs

There's no complete deniable system that includes storage and isolation, in particular in the context of PETs execution

Wide availability of forensic tools for persistent analysis