IUBIK: Isolating User Bytes in Commodity Operating System Kernels via Memory Tagging Extensions

IEEE S&P 2025

オペレーティングシステム特論　輪講

48256109 菊池凱成

背景

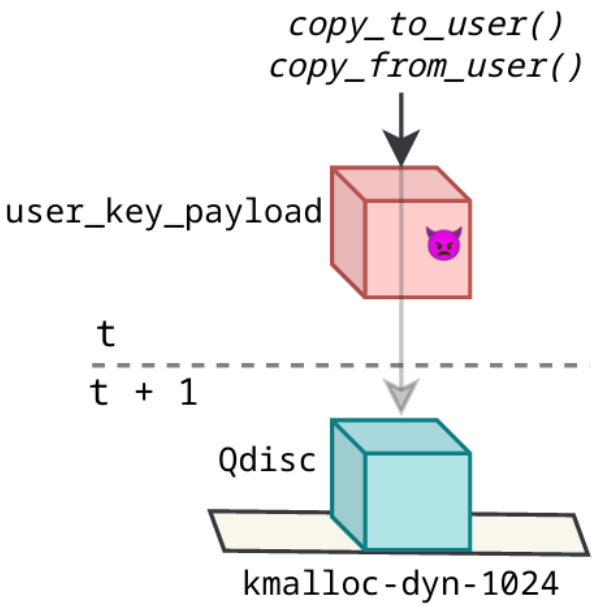
- メモリエラーを悪用する攻撃
  - メモリ安全、型安全でない言語（C, C++, ASMなど）
    - 空間メモリエラー
    - 時間メモリエラー
  - システムを完全に制御されうる

- セキュリティ上機密性の高いデータを保護
  - 保護すべきデータを特定するのは困難

→機密オブジェクトを特定しない手法を提案

同一キャッシュ攻撃

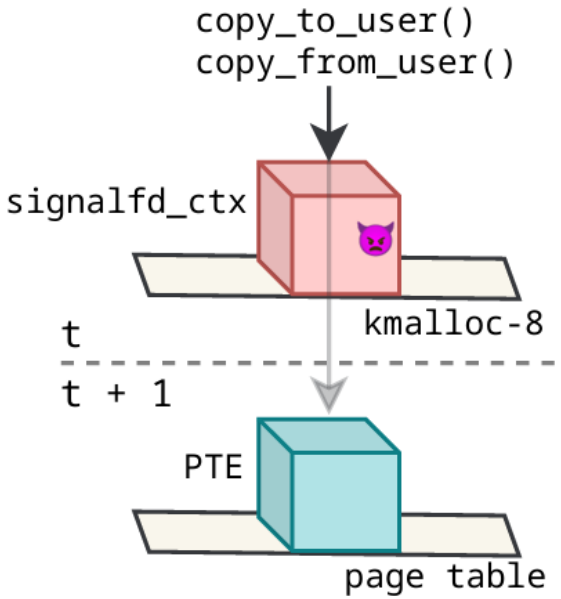
- カーネルオブジェクトとユーザ制御オブジェクト
  - Qdiscの関数ポインタをリークし、KASLRを回避
  - 権限昇格のためのROPチェーンを構築
- ユーザ制御オブジェクトを分離する？
  - 依然としてカーネルオブジェクトと同じキャッシュに



(a) Same-Cache

クロスキャッシュ攻撃

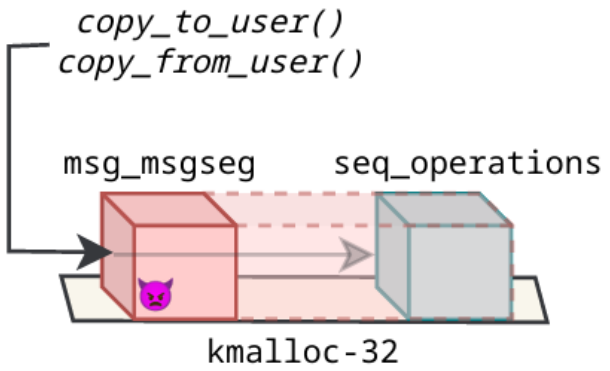
- 異なるキャッシュにある場合でも攻撃可能
- サイズが固定されているオブジェクトも利用可能
  - これまでの研究では対象とされていなかった



(b) Cross-Cache

境界外攻撃

- fsconfigシステムコールのバグを利用
  - msg_msg を攻撃対象の直後に配置
  - msg_msg のサイズフィールドを破壊



(c) Out-of-Bounds

提案：IUBIK

- Memory Tagging Extension (MTE) を用いて分離
  - ARMv8.5命令セットで導入されたハードウェア機能
  - ポインタのタグとメモリのタグが一致していればアクセスできる

- 空間的分離・時間的分離の両方を効率的に実現
  - パフォーマンスとメモリオーバヘッド

メモリの分離

- ヒープ領域をDomUとDomKの二つのドメインに
  - 割り当てられたメモリにTagU, TagK
  - メモリ割り当てのルーチンで指定
- 既存の仮想メモリレイアウトを変更する必要がない

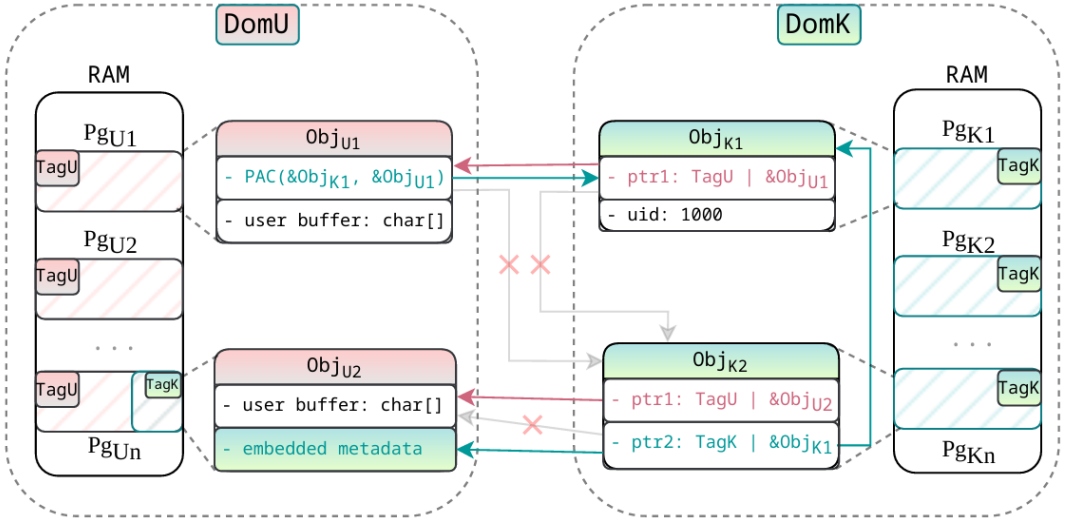


Figure 2: High-level overview of IUBIK's memory layout.

構造体の分離

- 構造体を書き換え、2つのオブジェクトに分割
  - ユーザ制御オブジェクトの多くは、機密カーネルフィールドを持つ
  - DomUでのメモリエラーを介して攻撃が可能

- 様々な課題
  - ユーザ制御フィールドの解放
  - オブジェクトのベースの取得
    - ARM PAによって認証

ユーザコピープロファイラ

- どのカーネルオブジェクトが危険かを自動で特定
  - 手作業ですべて特定するのは不可能
  - Linuxに導入されているプロファイリング機能の上に構築
  - カーネルのコードベースを探索
  - copy_from_user, memcpy, freeなどを追跡

評価：ユーザコピープロファイラ

- オーバーヘッドは3.12%
  - テストの実行時間が12時間35分から12時間59分
- 292の特権割り当てサイトと212の非特権割り当てサイトを記録（6902の内）
  - 多くがフレキシブルな配列を含む可能性が高いなど、悪用される可能性があるものだった
- 結果の妥当性も確認
  - 誤検知を記録しない

評価：実行時間

- Lmbench
  - fstat で8%、open/close で7%
  - その他は速度低下はごくわずか（5%未満）
- Phoronix Test Suite (PTS)
  - 全てのテストでわずかな速度低下（3%未満）
- System V Message Queues
  - レイテンシに大きな違いは見られなかった

評価：セキュリティ

- 3つの攻撃シナリオのテストケースで有効性を確認
  - 同一キャッシュ攻撃、クロスキャッシュ攻撃、OOB攻撃

- その他の実環境での攻撃に対しても有効性を確認
  - 対応できない3つの攻撃を確認
    - ユーザが制御するメモリに依存しない
    - 解放されたページをBaddy経由で割り当てられたユーザ制御ページとして再利用
    - SLUBの脆弱性を悪用して、割り当てられたページのUAFを取得

評価：メモリオーバーヘッド

- 新しい専用キャッシュの構成や、分離されたユーザ制御バッファを参照するポインタなどの影響を評価
- カーネル起動時に発生する最大 Resident Size Set (RSS) を比較
- SLUBに比べて2.17%多かった

まとめ

- メモリエラーを用いた攻撃に対し、機密データを保護するのではなく、攻撃の道具を隔離する手法を提案
- ARMのMTE/PAを用いて効率的にユーザ制御オブジェクトとカーネル制御オブジェクトを分離
- 合わせて、構造体の書き換え、ユーザコピープロファイラの実装
- 実行時間最大8%増、メモリオーバーヘッド2.17%
- セキュリティの有効性も確認