SecHack365

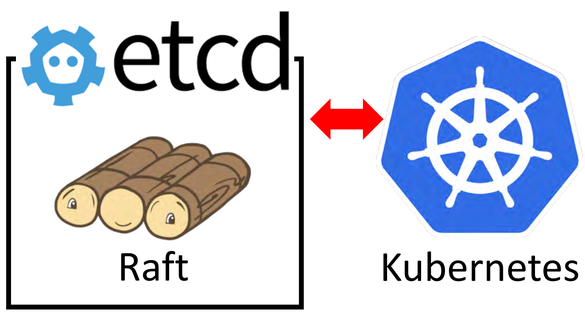
etsecd: 一部サーバーが窃取されても データが漏洩しない分散KVS

開発駆動コース 仲山ゼミ
27Dn 永見 拓人

背景: etcdとその課題

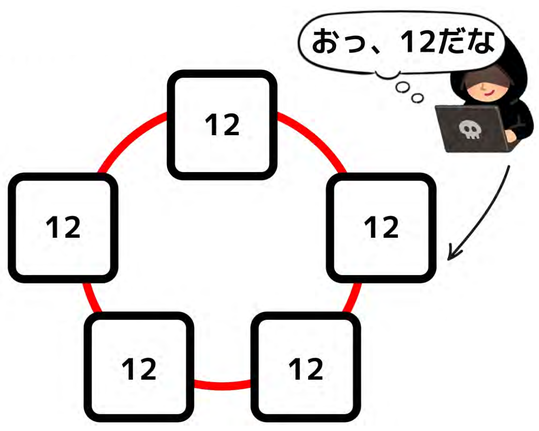
etcdとは

- **分散KVS** - 分散データベースの一種
  - **Kubernetes**のデータストア
  - **クラウド基盤の要**
- 合意アルゴリズム**Raft**がコア技術



etcdの課題

- データを**全サーバー**に**完全複製**する
  - 1つでもサーバーが窃取されれば **データが漏洩**

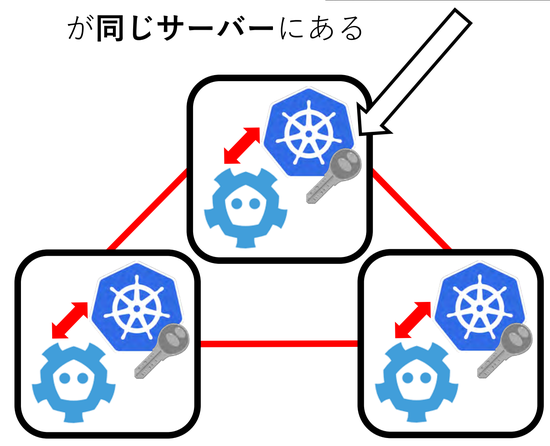


漏洩耐性を強化する必要性

- KubernetesのSecretリソース
  - **APIキー**などを保存
  - これが漏れたら**一巻の終わり**
- デフォルトのKubernetesでは **Secretも平文** (厳密にはbase64) で保存されている

鍵暗号も不十分

- **鍵暗号**によって漏洩耐性は強化可能
- **鍵管理**は非常に**面倒**で、**脆弱**になりがち
  - 特にKubernetesでは**復号する**サービスが**同じサーバー**にある



提案: etsecdとそのしくみ

(k,n)しきい値秘密分散法

- データからn個の**シェア**を生成
- **しきい値k個以上のシェア**が集まれば元データが**復元可能**
  - それ以下の個数では**復元不可能**
- **鍵管理無し**で上記の性質を実現

例: (k, n) = (3, 5)の場合

etsecd

- 秘密分散をetcdと**組み合わせた**分散KVS
  - 各サーバーで保存するデータが**シェア**になる
- **一部サーバーが窃取**されてもデータが**漏洩しない**

etsecdのあゆみ

世界初の**アルゴリズム**を考案・採用！！

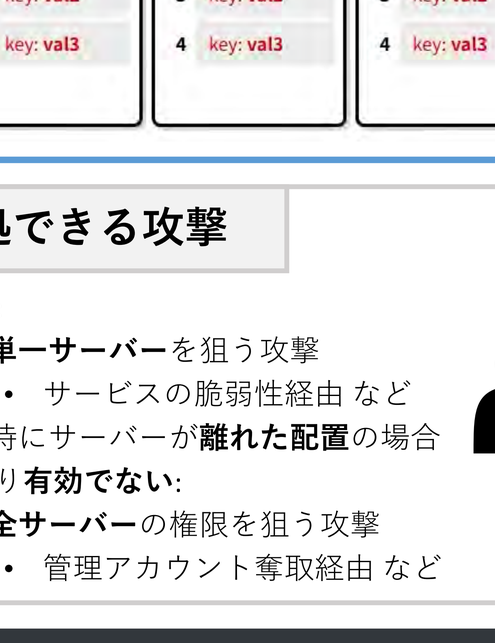
- **6月** (SecHack365開始時): **データ配置**の工夫による保護
  - アルゴリズムに手を入れるつもりでは**無かった**
- **8〜9月**: **アルゴリズム**の考案に**方針を転換**
  - 保存データをシェアに置き換えた際の**齟齬を埋める**
    - **コミットに必要な台数を調整**
    - 複数台での**データ復元のルール**を定義 etc...
- **10月**: **CSS2025**にて発表、リーダー選挙の**穴が見つかる**
- **11〜12月**: リーダー選挙の**穴を埋める**
  - リーダー交代で**読み込めなくなるデータ**を**無くした**
- **1月**: **SCIS2026**にて発表
- **1月**: **可視化ダッシュボード**の実装

SecHack365開始**当初の計画**

+α: 分散DBの可視化

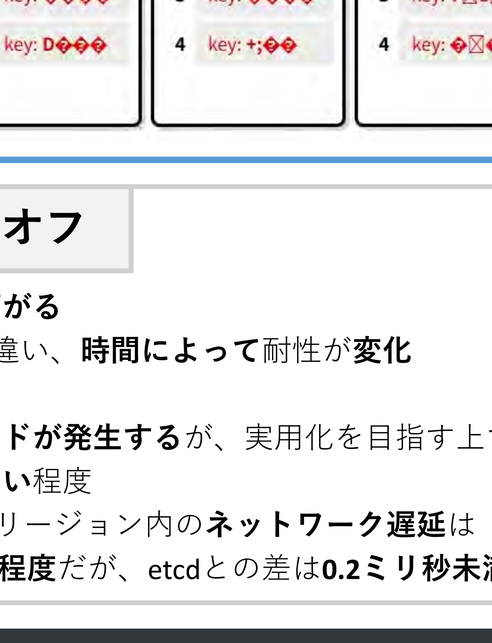
etcdとetsecdは**どちらもDB**ゆえ**中身が見えづらい**プロダクトです。
etsecdの仕組みを**わかりやすく見せる**ため、特に**可視化**に力を入れました。
実際に動いている様子は左のQRコードからご覧下さい。

デモ動画



トポロジー図

- 現在の**サーバーの状態**
- **流れる通信**　を表示



トポロジーとサーバー内データを**上下に分けて表示**

- **データの違い**を強調
- 最初はトポロジー図にまとめて表示予定だった

etcdのデータ分散

- 全サーバーに**複製**

etsecdのデータ分散

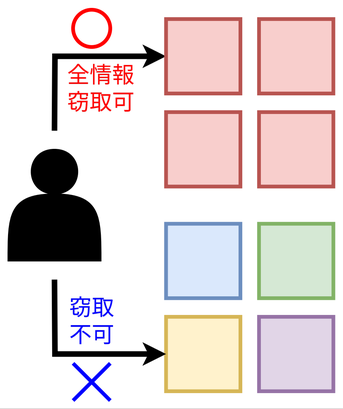
- **秘密分散**されている
- サーバーごとに**異なる**
- **推測困難な値**

Node 1	Node 2	Node 3	Node 4	Node 5
1 key: value	1 key: value	1 key: value	1 key: value	1 key: value
2 key: valval	2 key: valval	2 key: valval	2 key: valval	2 key: valval
	3 key: val2	3 key: val2	3 key: val2	3 key: val2
	4 key: val3	4 key: val3	4 key: val3	4 key: val3

Node 1	Node 2	Node 3	Node 4	Node 5
1 key: ◆F□>◆	1 key: 8◆□◆◆	1 key: ◆◆w◆C	1 key: ◆ϴ□y	1 key: □□□M□
2 key: \◆◆r3P	2 key: ◆◆◆◆□	2 key: ◆pE◆J◆	2 key: 3◆◆J?	2 key: □SzN□□
	3 key: ◆◆◆◆	3 key: ◆◆◆◆	3 key: 7□8□	3 key: 8If◆
	4 key: D◆◆◆	4 key: +;◆◆	4 key: ◆□◆◆	4 key: ◆◆◆◆

対処できる攻撃

- **有効:**
  - **単一サーバー**を狙う攻撃
    - サービスの脆弱性経由 など
  - 特にサーバーが**離れた配置**の場合
- あまり**有効でない**:
  - **全サーバー**の権限を狙う攻撃
    - 管理アカウント奪取経由 など



トレードオフ

- **障害耐性が下がる**
  - etcdとは違い、**時間によって**耐性が**変化**
- **オーバーヘッドが発生する**が、実用化を目指す上で**障壁にならない**程度
  - AWS同一リージョン内の**ネットワーク遅延**は**数ミリ秒程度**だが、etcdとの差は**0.2ミリ秒未満**

GitHubリンク

NICT National Institute of Information and Communications Technology