検査方法の検討と評価

フェリカネットワークス株式会社　　礒崎 亮多　Ryota.Isozaki@FeliCaNetworks.co.jp

開発における問題と課題

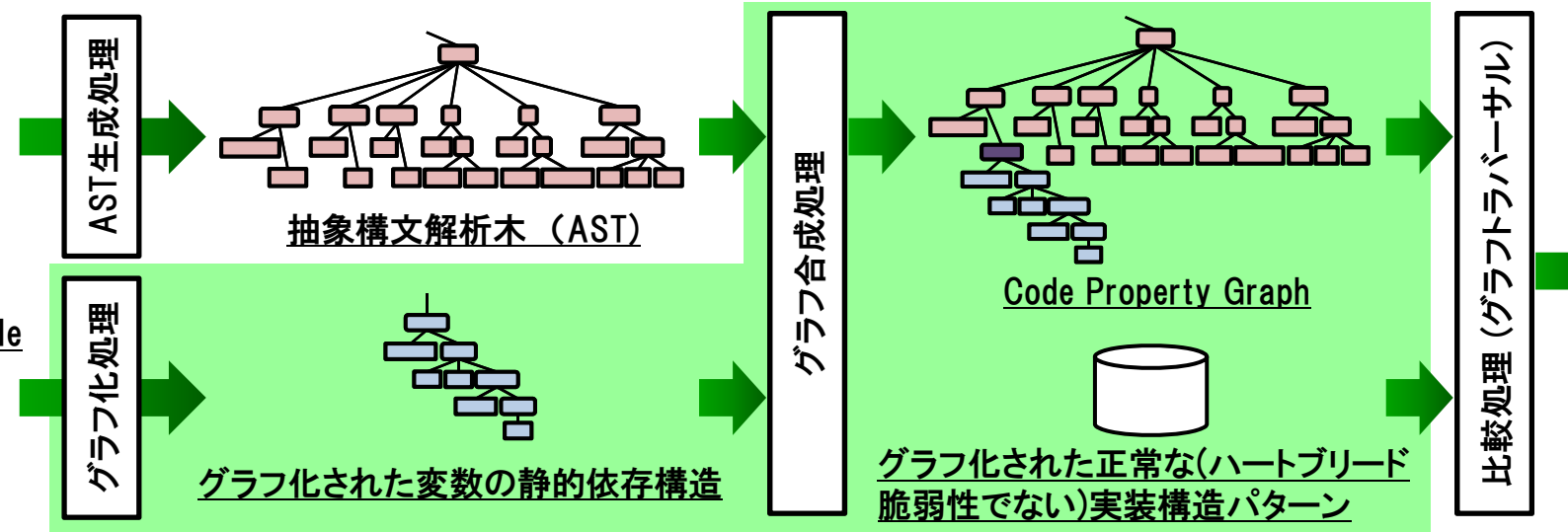
静的解析等のツールでは、ハートブリード
が一般的に広くサポートされてない

グラムコードに脆弱性が含まれてないか
実装の妥当性を目視確認する必要がある
自で自動検査ツールを実装したくとも、
弱性の検査手法の報告がされてない

検査手法の提案による解決

ハートブリード脆弱性を自動検査するため
手法（アルゴリズム）を提案し、課題を解決
[1] 従来未検知であった問題を検知可能
[2] 誤検知を極力少なくする
（誤検知が多いと目視確認工数は低減
されないため）

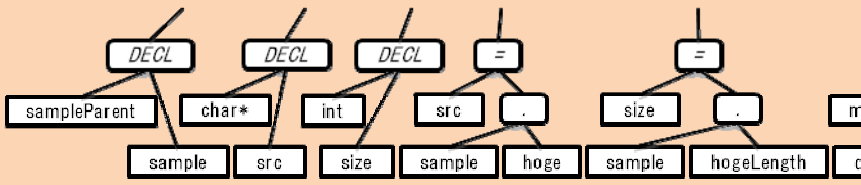
提案手法の概要



た正常な(ハートブリード脆弱性でない)実装構造パターン（一例）

```
)          src = sample.hoge;
           size = sample.hogeLength;
ample;          ⋯
rc;        memcpy( dst, src, size );
ize;            ⋯
        }
```

```
typedef struct _sampleParent
{
    char*  hoge;
    int    hogeLength;
} sample;
```



opensslに対する評価結果および結論 (提案手法の効果)

① 従来「未検知」であったものについて、「検知」が可能となった
② 「誤検知」は存在するが、目視確認の内容を約50%低減することが可能となった

分類	期待値	比較対象	提案手法
(true positive)	2	0 (※1)	2
(true negative)	182	- (※2)	88
(false negative)	0	2 (※1)	0
(false positive)	0	- (※2)	94
合計	184	-	1

	目視確認の箇所数	目視確
提案手法を利用しない場合	184件 (全件)	294
提案手法を利用する場合	2件 (検知箇所)	32min
	94件 (誤検知箇所)	1504min
提案手法利用による効果		1408min(