

The Impact of a Major Security Event on an Open Source Project: The Case of OpenSSL

James Walden waldenj@nku.edu Northern Kentucky University Highland Heights, KY

MSR2020

紹介者：大森伊月

背景・目的

- ・ 背景

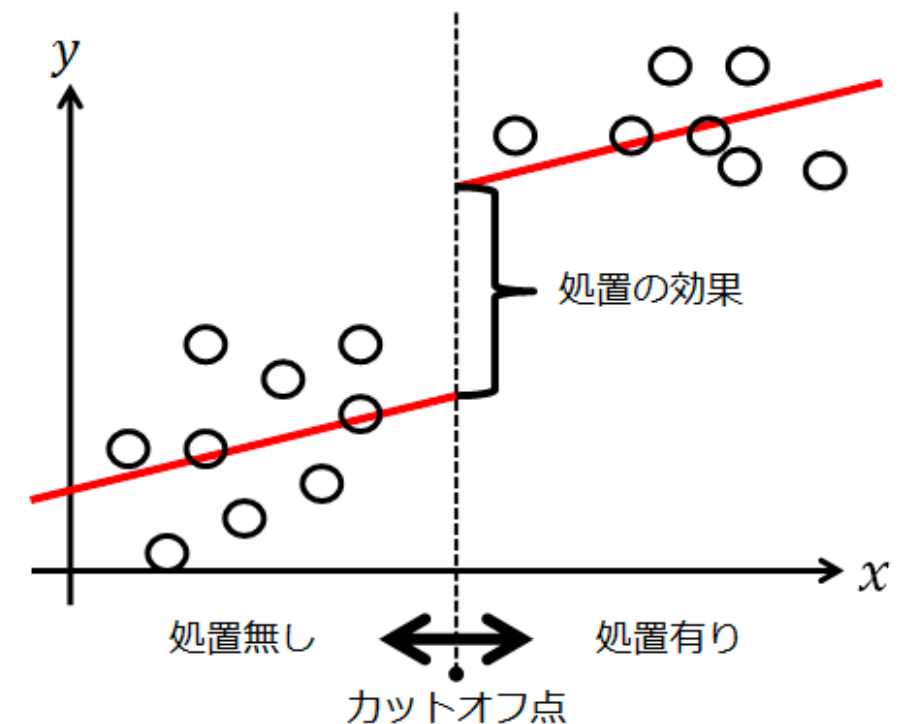
- ・ OpenSSLにおいて2014年に脆弱性Heartbleedが見つかった
 - ・ 暗号化しても漏洩する恐れがある脆弱性
- ・ それまでのOpenSSLはリリースポリシーなどがなく、品質やセキュリティが低かった

- ・ 目的

- ・ HeartbleedがOpenSSLの進化をどのように変化させたかを明らかにする

アプローチ

- ・ アプローチ
 - ・ 回帰不連続デザイン(RDD)を用いてメトリクスに変化が生じたか判断する
 - ・ ある点において行った処置が効果の有無を測定する手法
 - ・ 今回は脆弱性発見日をカットオフ点として、その効果を探る
 - ・ Heartbleed直後にプロジェクトに参加していた開発者へのインタビュー



研究課題

- RQ 1 : Heartbleedの後, 報告された脆弱性の数と深刻度はどのように変化したか
- RQ 2 : Heartbleed後のOpenSSLのサイズはどのように変化したか
- RQ 3 : Heartbleedの後, OpenSSLのソースコードの複雑さはどのように変化したか
- RQ 4 : Heartbleedの後, OpenSSLのソースコードのコーディングスタイルはどのように変化したか
- RQ 5 : Heartbleed後, コミット数と著者数はどのように変化したか
- RQ 6 : Heartbleedの後, ソフトウェアエンジニアリングの手法はどのように変化したか
- RQ 7 : Heartbleedから5年後, OpenSSLのセキュリティ・ソースコード・ソフトウェアエンジニアリングの手法の変更は維持されているか

RQ 1 : Heartbleedの後, 報告された脆弱性の数と深刻度はどのように変化したか

- 3つの時代に分けて考える
 - Heartbleed以前の時代
 - 脆弱性報告の多い時代(2014~2016)
 - 現在まで(2017~現在)
- 評価
 - 脆弱性の数とCVSSv2による深刻度のスコアを比較
- 結果

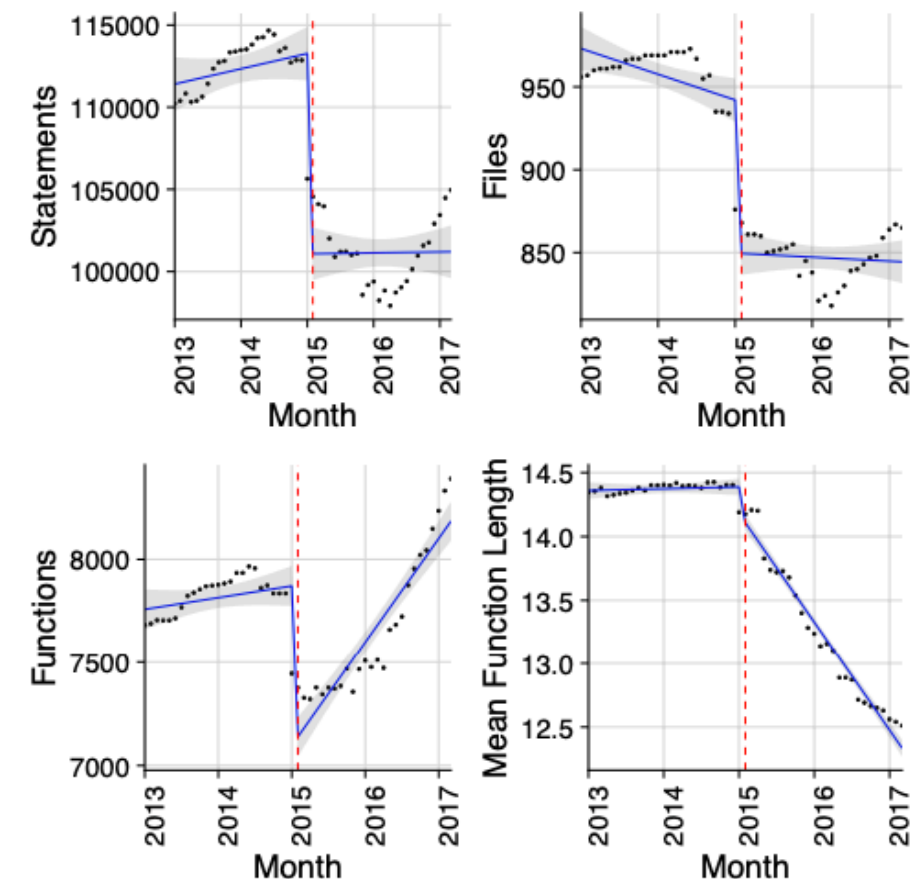
Table 2: OpenSSL Vulnerabilities

	Pre-Heartbleed	2014-2016	2017-2019
Vulnerability Count	64	91	22
Vulnerability Percent	36.2%	51.4%	12.4%
Vulnerabilities per Year	4.00	30.33	7.33
Average CVSS2	5.63	5.25	4.18
StdDev CVSS2	1.11	1.11	0.39

Heartbleed以降の脆弱性の年間発生数は多いが、深刻度は低下しているのでセキュリティは改善している

RQ2:Heartbleed後のOpenSSLのサイズはどのように変化したか

- プロジェクトのコードサイズなどが測定できるプログラムcqmetricsを使用して測定し，RDDに適用し評価した
- 結果
 - 2015年にコードクリーンアップが行われたため，StatementsとFilesは大幅に減少
 - 関数の数が上昇し関数の平均長が減少しているのはコードをより小さい関数に分解したため



(a) Code Size Metrics

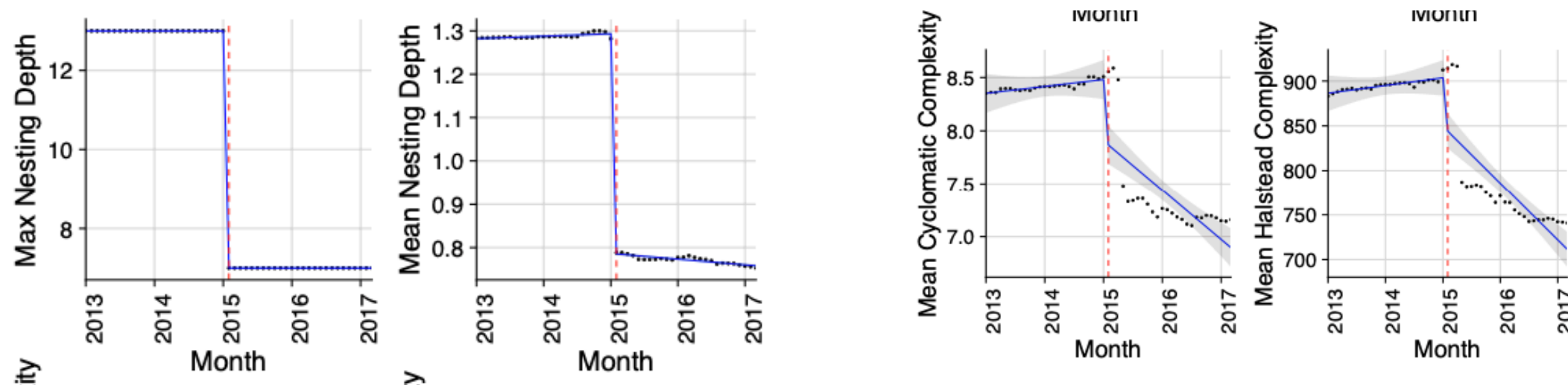
RQ3:Heartbleedの後, OpenSSLのソースコードの複雑さはどのように変化したか

・ 評価

- ・ ネスティング複雑性：インデントの深さを複雑度として測定する手法
- ・ 循環的複雑性：分岐の数によって複雑度を測定する手法
- ・ Halstead複雑性：オペランドの種類や数によって複雑度を測定する手法

・ 結果

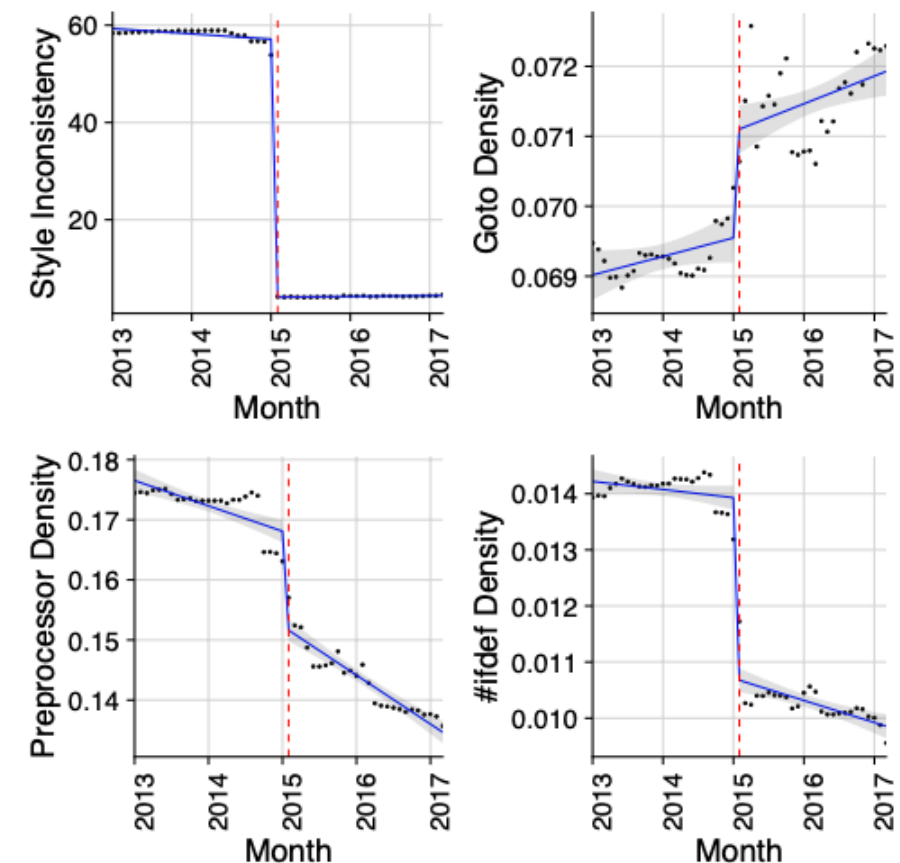
- ・ コードクリーンアップによって可読性と保守性が向上



(b) Code Complexity Metrics

RQ4:Heartbleedの後, OpenSSLのソースコードのコーディングスタイルはどのように変化したか

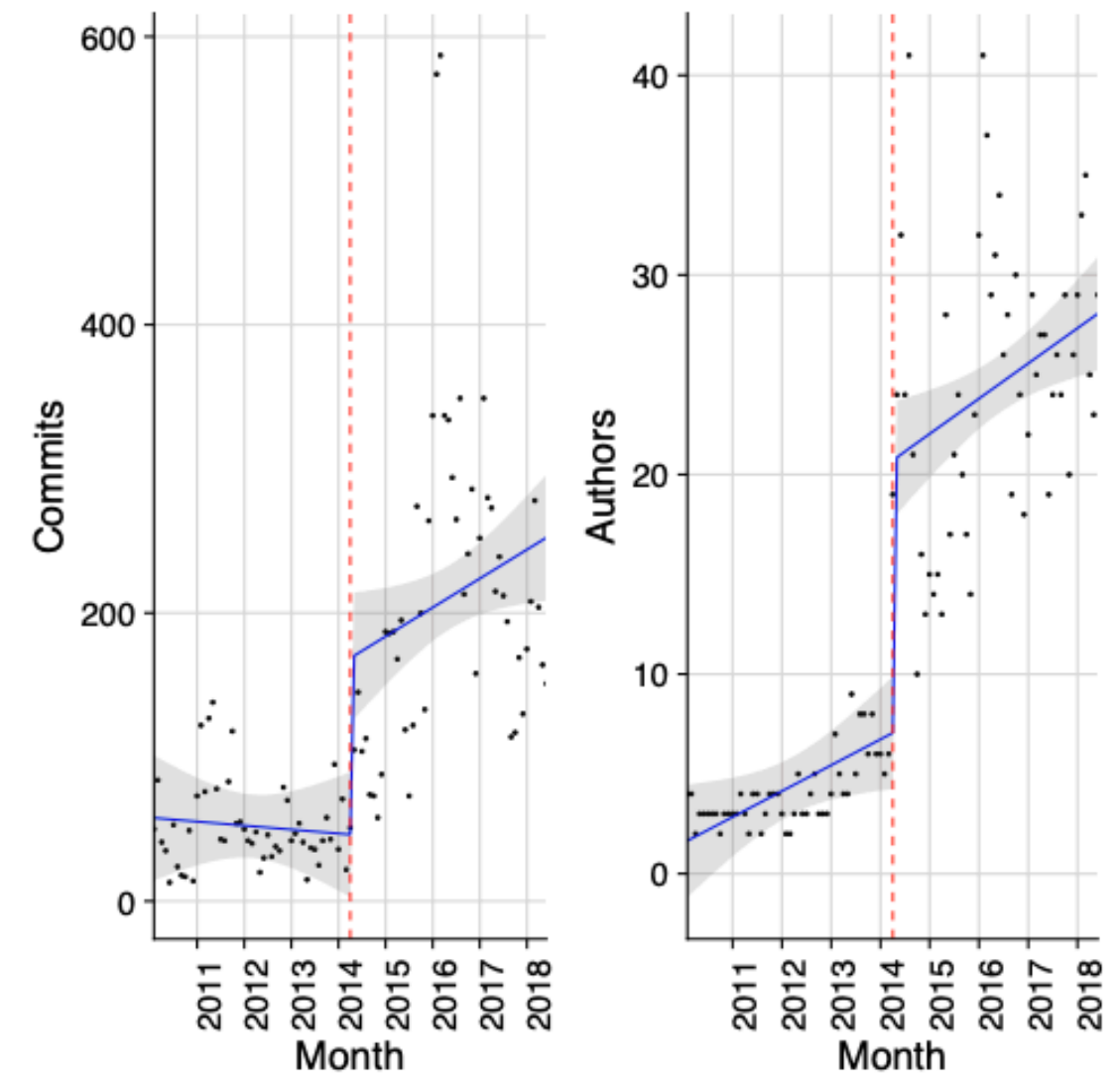
- 評価
 - Cqmetricsを用いてフォーマット規則に従っているかを評価
 - Gotoの使用密度
 - Cプリプロセッサ文の密度
 - 条件付きCプリプロセッサ密度
- 結果
 - Goto以外は大幅に減少
- 考察
 - コードの可読性と保守性の向上に注力したと考えられる



(c) Code Style Metrics

RQ5:Heartbleed後, コミット数と著者数はどのように変化したか

- 評価
 - gitでのcommit数とcommit著者の数をRDDに適用して評価
- 結果
 - プロジェクト活動には
明確かつ大幅な変化が見られた



(d) Project Activity Metrics

RQ6:Heartbleedの後、ソフトウェアエンジニアリングの手法はどのように変化したか

- 当時の開発者一人に聞き取りを行うとともに、OpenSSLのGitHubリポジトリ・ML・webサイトを調査
 - プロジェクトロードマップを発表
 - CIIがプロジェクトに資金提供
 - GitHubへ移行
 - リリース戦略を発表し、将来のバージョン計画を策定
 - 長期サポートバージョンのリリース・以前バージョンとの互換性を破棄
 - OpenSSL Management Committee (OMC) やコミッターなどの役割を明文化
 - コミッターポリシーの更新でコードレビューには二人のコミッターがレビューする必要がある、そのうち一人以上はOMCである必要があるようになった
 - CIIのBest Practices Badgeを取得(62% -> 105%)
 - 単体テスト数の増加、静的解析ツールの導入

RQ7:Heartbleedから5年後, OpenSSLのセキュリティ・ソースコード・ソフトウェアエンジニアリングの手法の変更は維持されているか

- セキュリティ監査がCIIに委託されたOpen Crypto Audit Project(OCAP)によって実施された
 - コード実行が可能になる2つの脆弱性とサービス停止の可能性のある2つの脆弱性(DoS)が見つかった
- Open Source Technology Improvement Fund(OSTIF)が擬似乱数生成期に加えられた変更に焦点を当てて監査を実施
 - リリース日前に2つのDoS脆弱性を発見し, チームがリリース前に修正

結論

- OpenSSLはHeartbleedから5年経った現在でも広く普及している
 - →これらの改善は大規模なセキュリティイベントが発生した後にOSSプロジェクトがいかに適応し改善していくかのモデルになる
- この3年間で発見された91件の脆弱性のうち、Heartbleed後のコードに含まれていたものは5件であった
 - →脆弱性の数はプロジェクトのセキュリティを示す有用な指標ではない
 - ただ脆弱性の発見に力を入れていないだけかもしれない

まとめ

- 目的
 - HeartbleedがOpenSSLの進化をどのように変化させたかを明らかにする
- アプローチ
 - RDDを用いた定量的な分析
 - プロジェクト参加者への聞き取り調査
- 結論
 - OpenSSLはOSSプロジェクトがいかにしてセキュリティイベントに適応し、改善できるのかのモデルにすることができる
 - 既知の脆弱性の数はプロジェクトのセキュリティに有用な指標ではない

所感

- ・ 聞き取りは一人で十分なのか？
 - ・ データの偏りがある気がする
- ・ 2度の監査において少なからず脆弱性が発見されているが問題はないのか？