プロトコルの形式検証と脆弱性発見の現実

- Case of CCS Injection -

株式会社レピダム 林 達也 (@lef)

HAYASHI, Tatsuya / Lepidum Co. Ltd.

"CRYPTREC シンポジウム 2015"

(2015/3/20)

自己紹介・業務領域

名前

- 林 達也 (@lef)

所属

- 株式会社レピダム 代表取締役
- Internet Society Japan Chapter Online Identity WG チェア/ プログラム委員
- OpenID Foundation Japan 事務局長
- Identity Conference (#idcon) オーガナイザー

業務領域

応用・実用研究

- 標準化支援
- アイデンティティ、プライバシー
- 認証・認可
- ソフトウェア&ネットワークセキュリティ, 脆弱性
- ネットワーク技術
- プログラミング言語処理系
  - コンパイラ, インタプリタ, 言語設計
- 各種コンサルテーション

Focus Area | Lepidum

- ***Applied Research and Development***
- ***Personal Data, Digital Identity and Privacy***
- ***Secure and Safety Software Technology***
- ***Web and Internet Technology***
- ***De-Facto and Forum Standardization***

- ***Keywords:***
  - Personal Data, Trust Framework, Privacy, ID Federation, Authentication/Authorization, Protocol Specification, * of Things(IoT, WoT), Software Defined Network, Autonomic Network, etc...

Projects | Lepidum

- ***LACCOONS***
  (Large-scale Automatic Configuration and Control Over Open Network for SvDI)
- ***HTTP Mutual Access Authentication Protocol***
  (New protocol for preventing Phishing attacks against Web systems)
- ***IDzumo*** (Digital Identity Components)
- ***Skyeye*** (Wi-Fi Connect with AuthN/Z Components)
- ***Fail-Safe C*** (Memory-safe implementation of the full ANSI C language Compiler and libraries)
- ***Service Defined Network*** (SvDI)

Acknowledgement

- 暗号の専門家ではないので不正確な部分がある可能性があります
- 発表者は脆弱性の発見者ではありません(発見者：レピダム 菊池正史)
- コードの細部等、技術的に突っ込んだ点には回答出来ないかもしれませんが、お許し下さい

Point of View

脆弱性発見者 (組織の経営者)の視点	脆弱性対応者 (非サービス事業者)の視点	エンジニア・ 標準化策定者の視点
基盤技術の脆弱性 発見の意味	脆弱性発見経験を 踏まえて	基盤技術の 重要性
瀬戸際ぎりぎりを 歩くような危うさ	適切な情報開示と 共有	事後より事前のリ ソースの投入

プロトコルの形式検証と脆弱性発見の現実

- Case of CCS Injection -

CCS INJECTIONの発見とその経緯

lepidum

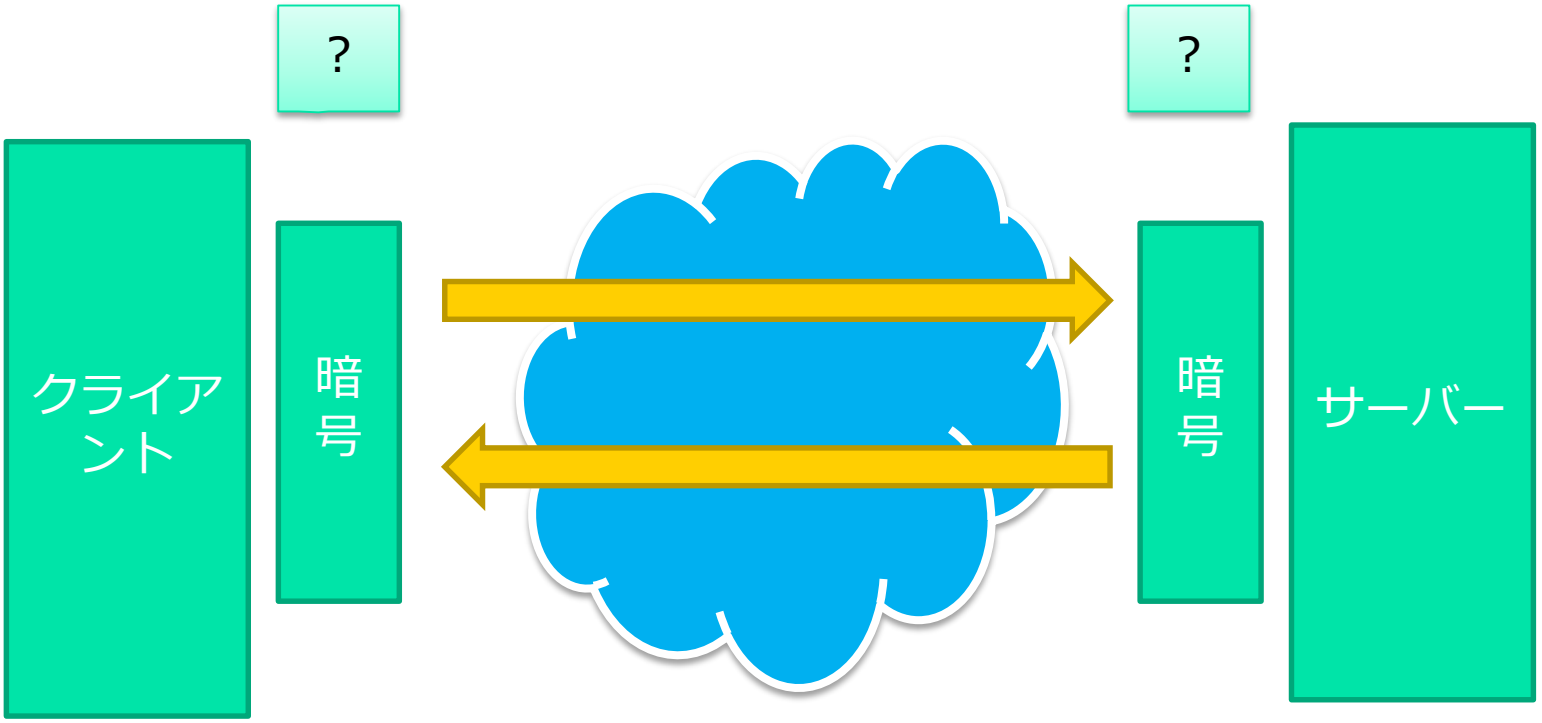
https://lepidum.co.jp/

CCS Injection脆弱性とは

- CCS = Change Cipher Spec
  - TLS/SSLのメッセージ
  - "ここから暗号を変えますよ！"
- CCS Injection脆弱性
  1. 中間者(攻撃者)がCCSを挿入すると
  2. OpenSSLが適切な検証をせず受理してしまい
  3. 変なタイミングで暗号が変わって
  4. 中間者が通信を完全に解読できる

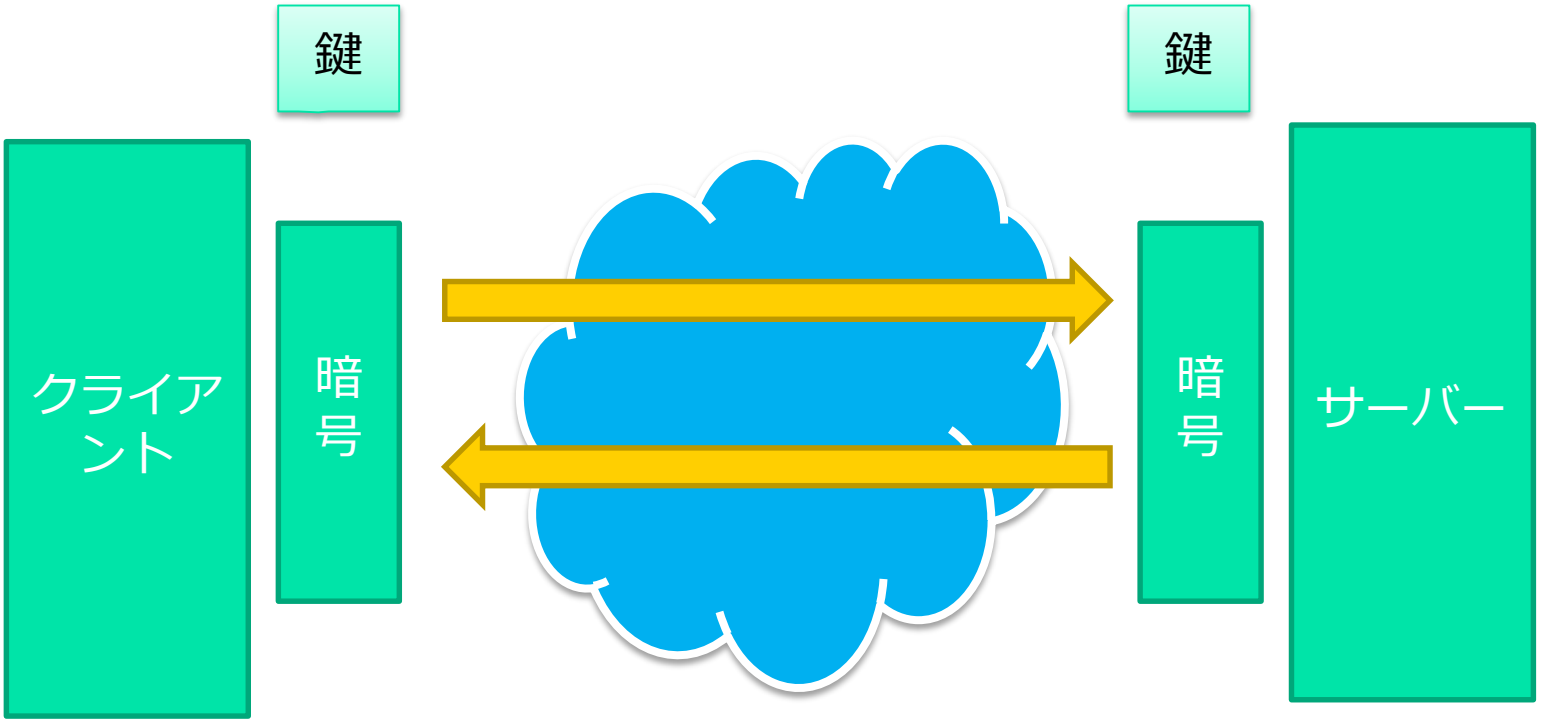
通常のCCS処理(1)

- 最初は鍵はない



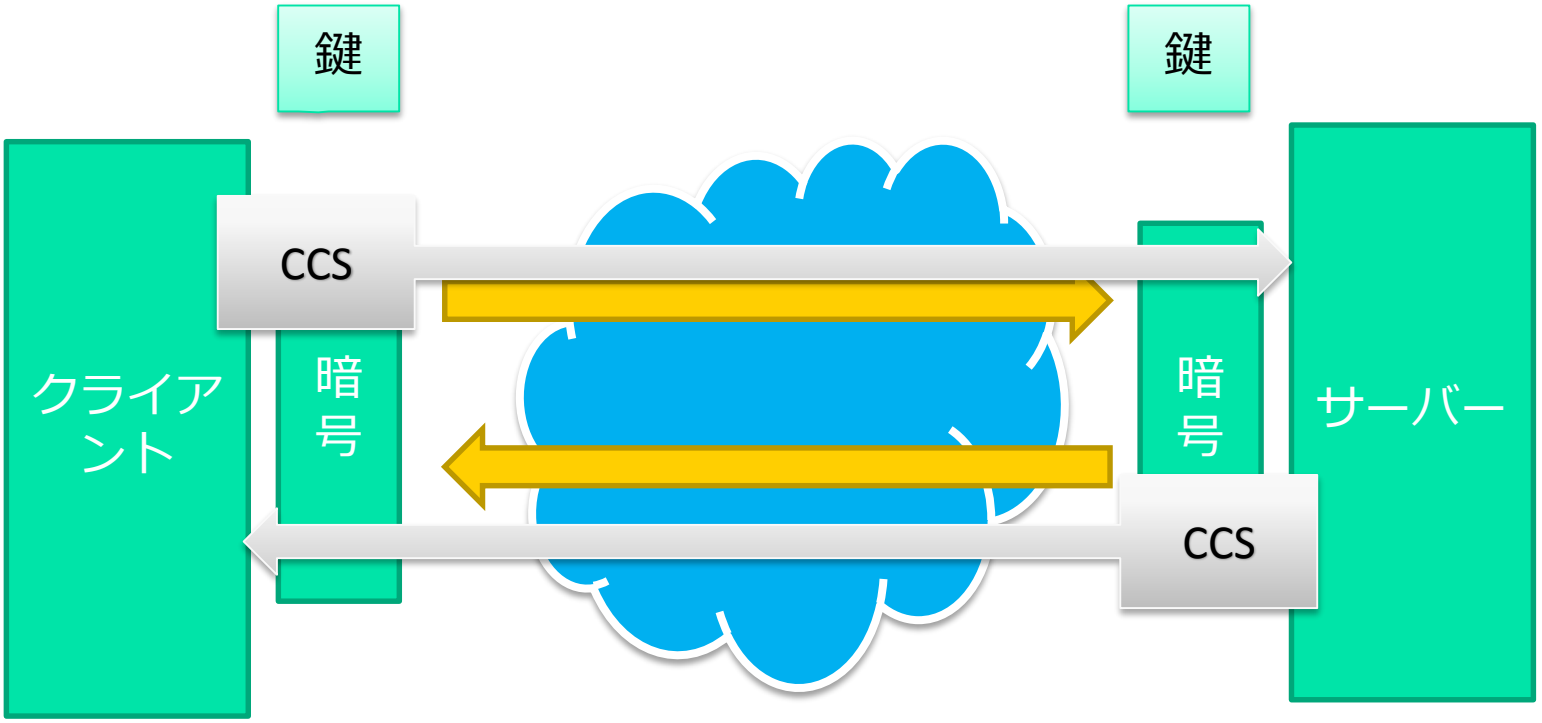
通常のCCS処理(2)

- PKIを使って鍵が交換される



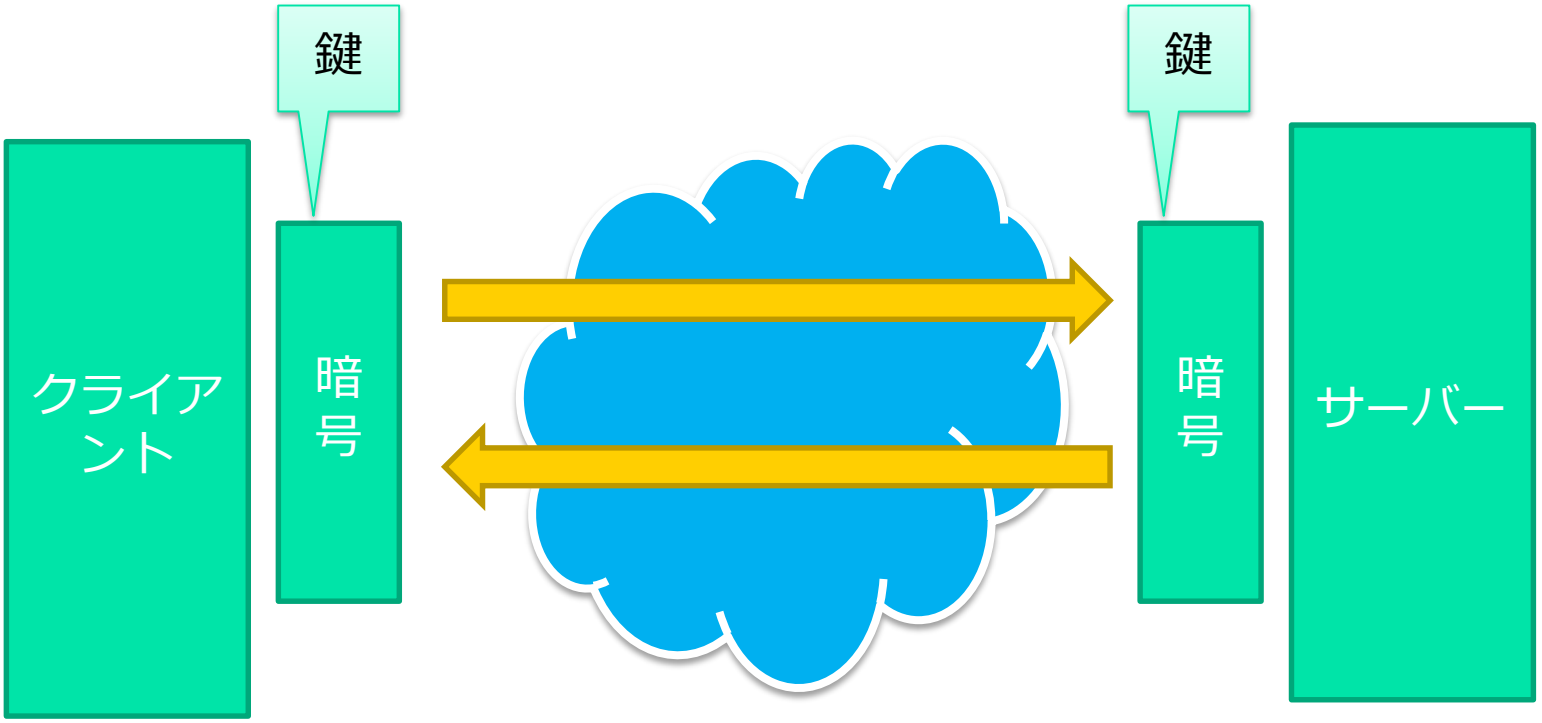
通常のCCS処理(3)

- CCSメッセージを送って暗号を有効化



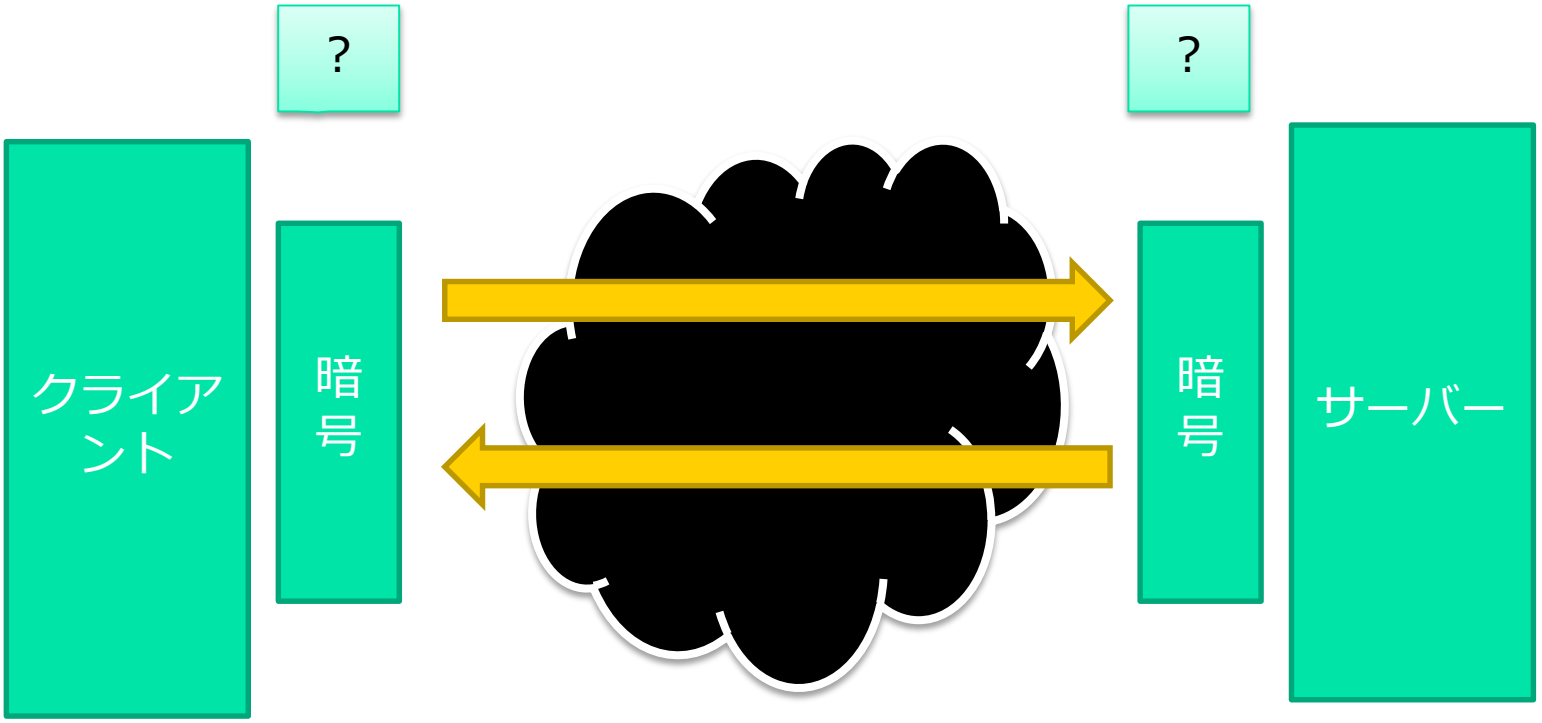
通常のCCS処理(4)

- 暗号通信の開始



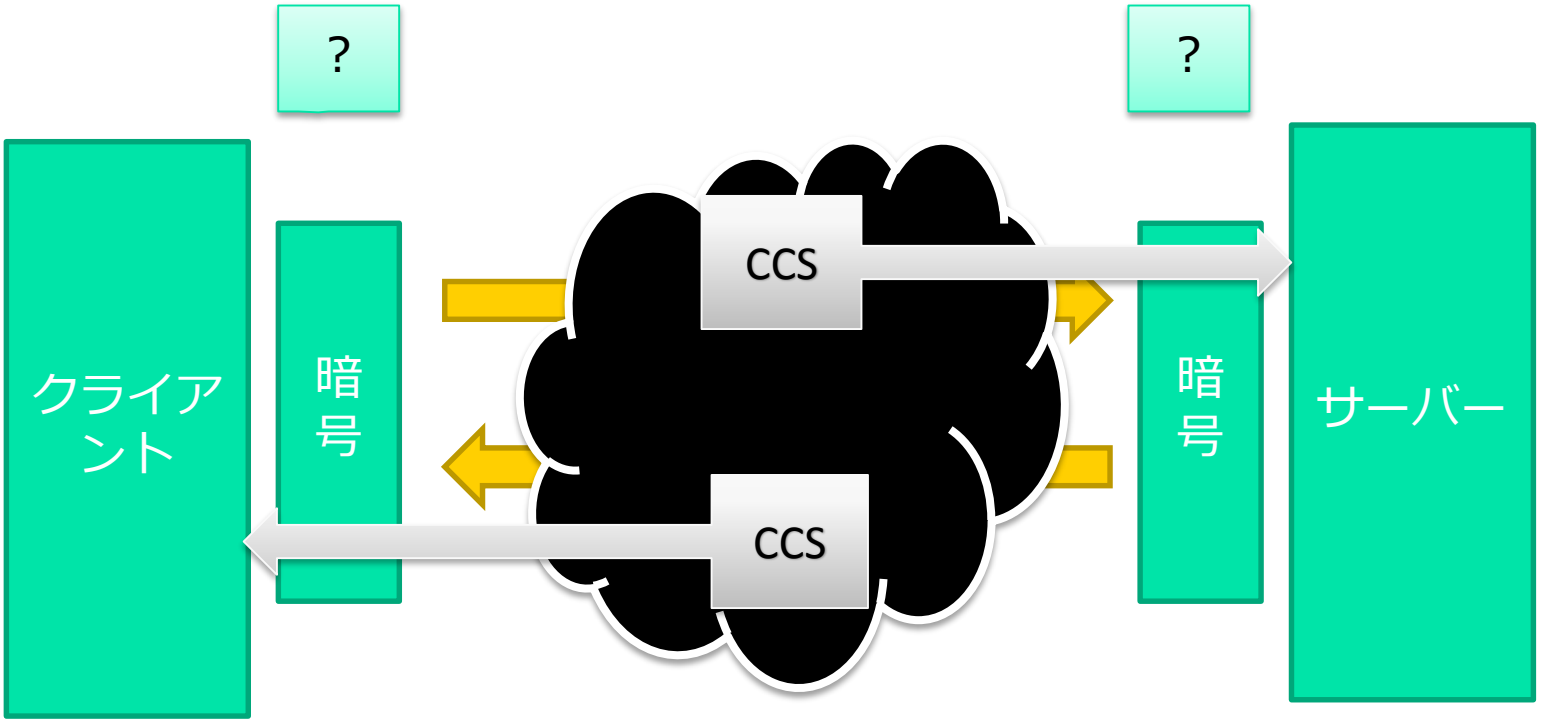
CCS Injection攻撃(1)

- 最初は鍵はない (同じ)



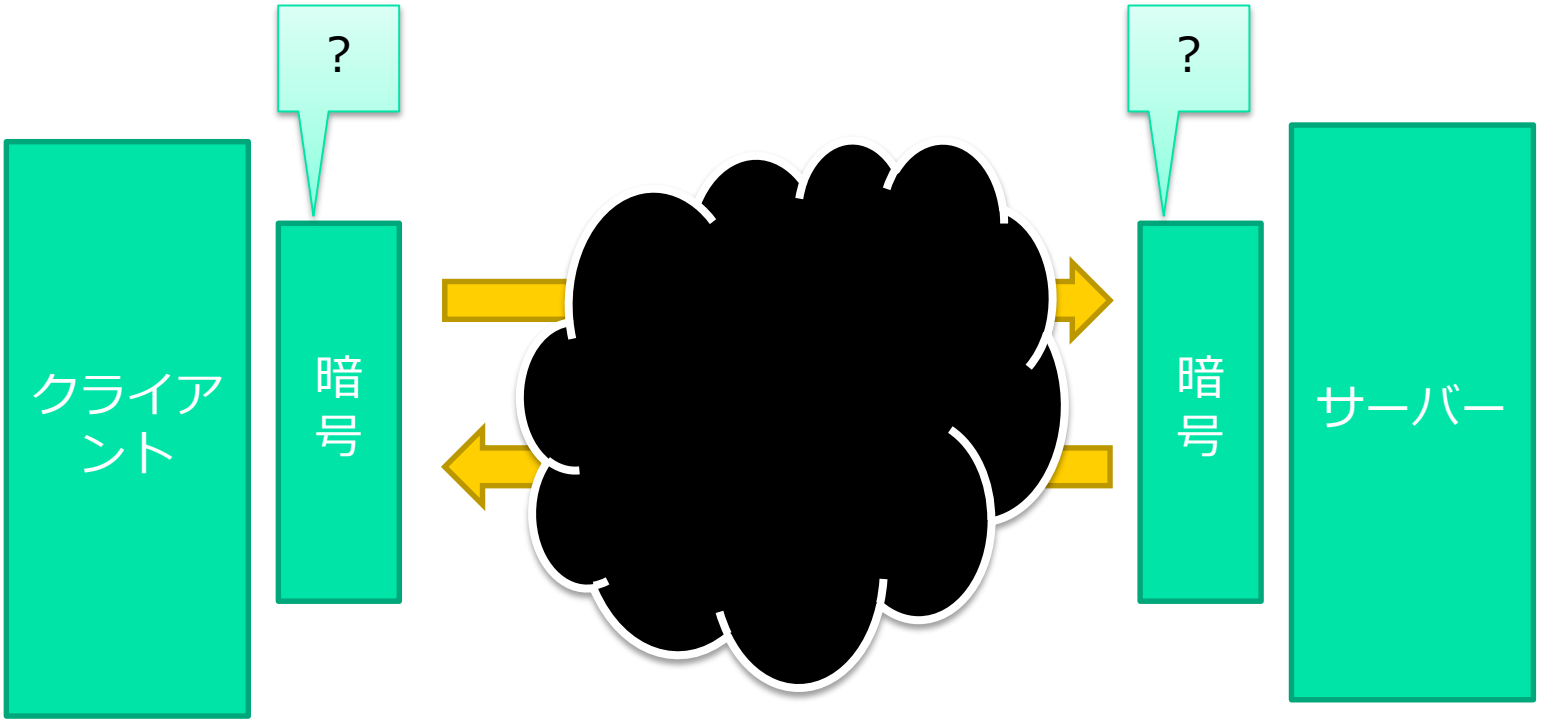
CCS Injection攻撃(2)

- 黒いところにいる攻撃者がccsを送る



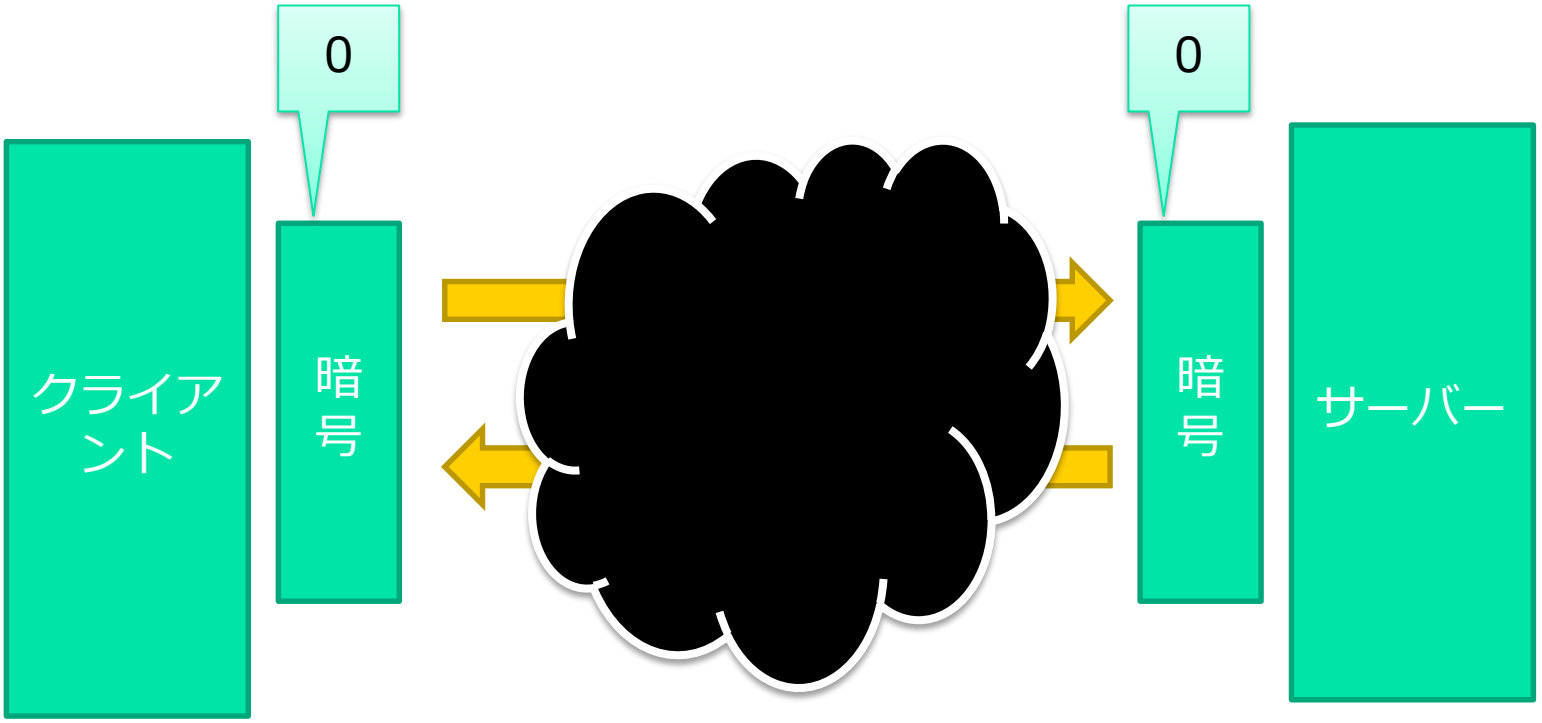
CCS Injection攻撃(3)

- 謎の鍵を使って暗号通信を開始してしまう



CCS Injection攻撃(4)

- 謎の鍵は初期値0から作られた自明なもの



CCS Injection攻撃(5)

- 実際はこの後にPKIの処理が走ったり、鍵交換の結果を検証するメッセージ等がある
- それらの辻褄をうまく合わせるように改竄することで自明な鍵を維持したままハンドシェークを完了させることが出来た
  - 謎の鍵等を含めOpenSSLが本来想定している内部状態が異常になっているので、つじつま合わせには技術的な苦労が

発見の背景(1)

- 元々、TLS/SSLの安全性に関するソフトウェア的な研究を行っていた
  - NICT「通信プロトコルとその実装の安全性評価に関する研究開発 -形式手法による プロトコル実装の検証技術と形式仕様に基づく網羅的ブラックボックス検査技術の開発-」
  - 平成22～24年度 (産業技術総合研究所/レピダム)
  - 定理証明支援系 "Coq" 等を利用

発見の背景(2)

- Internet Engineering Task Force (IETF)でのHTTPやTLSに関係する標準化活動の中で、Beast AttackやCRIME Attackの対応・影響を間近に体験
  - 実際に、CRIME Attack公開時にはHTTP Basic認証への影響に気づいた為、PoCデモの作成を行い、発見者との情報交換等も

発見の経緯

社内のエンジニア(発見者の菊池)が

状態機械の一番複雑な部分が
ChangeCipherSpecにおける遷移

だと考え、

既存実装(OpenSSL)の動作を確認したところ
検査が不適切であることに気づいた

発見者の初期モチベーション

- 「みんながバグ探し競争してるので効率的に探したい」
- 「Coqもどこかで使いたい」
- 「バグのありそうなモジュールを経験的に決めうちする」
- 「意味不明なコードを理解する足がかりが欲しい」

発見者の初期モチベーション

- 「みんながバグ探し競争してるので効率的に探したい」
- 「Coqも
- 「バグの 経験的に決めうち
- 「意味不 がかりが欲しい」

しかし…
Coqを使うまでもなく
脆弱性発覚

Coqとは？

- 定理証明支援系
  (formal proof management system)
  - フランス国立情報学自動制御研究所(INRIA)で開発されている(OCaml等を開発している)
  - 命題と証明を記述すると、"**証明が正しいこと**"を**自動**で検証してくれる
    - 一部は自動での証明も行う
- "プログラミング Coq"
  - http://www.iij-ii.co.jp/lab/techdoc/coqt/

脆弱性実証コードの作成

- rubyで300行くらい
- 暗号プリミティブは**openssl**ライブラリを使用
- 二日ほどで完成
- 壊れた状態でハンドシェークを完了させるために、細かな工夫(試行錯誤)が必要

『結局Coqのコードは
100行ぐらいしかできなかった』
(**継続したいのでスポンサーを募集しております**)

CCS Injection Vulnerabilityの課題1

- このバグはOpenSSLの最初のリリースから存在しており、該当コードは

16年そのままだった

- しかも、修正のチャンスも3回ほどあった

(1) CVE-2004-0079

- new_cipherが設定されているときだけChangeCipherSpecを受理するように
- しかしnew_cipherはServerHelloの段階で設定される為、正しい実装にならなかった

CCS Injection Vulnerabilityの課題2

(2) CVE-2009-1386

- DTLSで、いきなりChangeCipherSpecを受け取った場合に落ちる問題が修正されたが、やはり正しい実装にならなかった

(3) DTLS fragment retransmission bug (OpenSSL bug #1950)

- 更にDTLSにおいて、運悪くパケットの順番が入れ替わってChangeCipherSpecを受け取った場合の検査が追加され、DTLSでは正しい実装に
- 未計算のマスターシークレットが暗号に使われた結果、ランダムなメモリを読み出してハンドシェークが失敗すると分析されているが、実はランダムなメモリではなく空のバイト列だった
- これに気付いていれば**攻撃の可能性が分かったはず…**

プロトコルの形式検証と脆弱性発見の現実

- Case of CCS Injection -

脆弱性発見とその報告・公開

lepidum

https://lepidum.co.jp/

どこに報告すべきか？

- 日本語？英語？
- OpenSSL？CERT？
- 正しく影響範囲が伝わるか？
- そもそも自分達の解析は正しいのか？
- 検証と社内統制

報告した後…

- 0-day攻撃の可能性も含め危機管理・情報管理が必要に
  - 基本的には連絡を待つことしか出来ない
  - 他社・他組織との相談等は事実上行えない
  - 箝口令を引かれているスタッフの危機感
  - 特にレスポンスがないと「本当に報告プロセスは正しかったのか？」等の焦燥感
  - 影響範囲が大きい(すぎる)と適切な報告先の選択すら難しい

報告した後…

- 0-day攻撃の可能性も含め危機管理・情報管理が必要に
  - 基本的 ない
  - 他社・ えない
  - 箝口令 機感
  - 特にレ 告プロセスは正し
  - 影響範囲が大きい(すぎる)と適切な報告先の選択すら難しい

本当につらいです

何があると良いのか？

各組織で身を切って
やる意味はない！

情報公開・共有の基盤が必要

レスポンスの早い専門組織とその枠組

適切なガイドライン

国内だけでなく国際的にも

(この状況でInternet of Things…？！)

情報公開(Blogサイト準備)

- ドメイン取得検討 (ドメインドロップ対策も)
- 広告の非掲載を徹底 (信頼の為)
  - ⇒ 是非の議論
- 負荷対策
  - CDN検討・検証 (CloudFront VS CloudFlare)
  - キャッシュが有効なBlog構成
  - CDNとBlogの動作検証
- サイト更新手順の確認
- 情報収集・管理体制の構築、徹底
- etc...

こういう反省点を
共有したい (1)

「正確な」情報公開のポリシーは？

- 誰かが明確に提示してくれるわけではない
  - 対外発表の機会と日程調整

直後にInterop Tokyoで暗号通信に関する登壇が

- ネーミングが直接的すぎてDNS更新すら危ぶむ羽目に

こういう反省点を共有したい (2)

- ルールやガイドはない
  - 良識で判断 ⇒ ？良識？

公開当日

- 公開時刻は不明なので手動確認！
  - CERTを含め、誰も発見者に教えてくれるわけではない
- 取材・お問い合わせ等
  - メディア、英語対応、お客様、ご相談、SNS…
    - The Guardian, New York Times, etc...
    - 適切な取材とそうでない取材
    - お取引先への調整
  - Blogを公開してなかったら耐えられなかった可能性が高い
- 情報更新
  - 情報の更新などのキャッチアップ
  - 有識者からの連絡とそうでないものの分別

公開当日

- 公開時刻は不明なので手動確認！
  - CERTを含め、誰も発見者に教えてくれるわけではない
- 取材・お問
  - メディア
    - The Gu
    - 適切な
    - お取引
  - Blogを公
- 情報更新
  - 情報の更新などのキャッチアップ
  - 有識者からの連絡とそうでないものの分別

全社体制！
(業務停止)

FAQ・その他

- なぜロゴを作ったのか？
- 『どのくらい儲かりました？』
- エンジニア達のケア
- 経営者的な苦悩

情報管理の重要性

- 不要な危機感は煽りたくない
- でも必要なところには『適切に』情報が届いて欲しい
- 『準備が出来たら』広域に対策をアナウンスしたい

脆弱性公開の厳しさ

- 協力依頼も出来ないし、
  もちろん誰も助けてなんてくれません

- 変な会社だから出来たし、
  企業組織だから耐えられた自負はあります

脆弱性公開の厳しさ

- 協力依頼も出来ないし、
  もちろん
- 変な会社
  企業組織

せん

あります

それでも
やってよかったと
思っています

プロトコルの形式検証と脆弱性発見の現実

- Case of CCS Injection -

暗号技術基盤の脆弱性と現実

lepidum

https://lepidum.co.jp/

CCS Injection発見を踏まえて

CCS Injectionは
偶然見つかったに
等しい

基盤技術の応用研究に
もっと力を入れる
必要がある

国内には貢献できる
能力を持つエンジニ
ア・研究者が山程いる

信頼と安全のバランス
が崩れている事例

CELLOS

Cryptographic protocol Evaluation toward Long-Lived Outstanding Security

Publication

■ 公開情報

本コンソーシアムのアウトプットは、暗号プロトコルの安全性に関する、専門家による幅広く信頼性の高い情報であり、以下の3つからなる。

- 暗号プロトコルへの攻撃手法に関する最新情報（速報）
- 攻撃手法の実際のICTシステムへの影響と対処方法（報告書）
- 速報と報告書の検索システム（Protocol zoo）

速報

今後、学会等で発表された暗号プロトコルへの攻撃手法に関する情報を、事実関係を確認した上で速報的に提示します。内容は、エグゼクティブサマリ、攻撃手法の概要（詳細な手法は除く）、情報のソースへのリンクからなります。

- [2015/3/04] SSL/TLSへの弱い暗号を利用したFREAK Attackについて
- [2014/12/12] TLS 実装上の問題による POODLE attack の影響について
- [2014/10/15] SSLv3仕様そのものに対する POODLE attack について
- [2014/8/08] Black Hat 2014 USAで発表されたSSL/TLSへの複数の新たな攻撃手法について
- [2014/6/06] OpenSSLに関する7つの問題点とその対策について
- [2014/6/05] GnuTLSの暗号ライブラリにおける問題について
- [2014/4/18] OpenSSLにおける Heartbleed Bugと呼ばれる脆弱性について
- [2014/3/13] TLSにおける新たな攻撃の報告

報告書

CELLOS

Cryptographic protocol Evaluation toward
Long-Lived Outstanding Security

HOME

HOME
CELLO
What's
概要
活動内容
発起人
体制
構成
規約
役員
会員
公開情報
速報
報告書
Protocol zoo
シンポジウム
2014
調査対象
イベント
アーティクル
関連サイト

Publication

による幅広

TLS/SSL関係のみ！

係を確認し

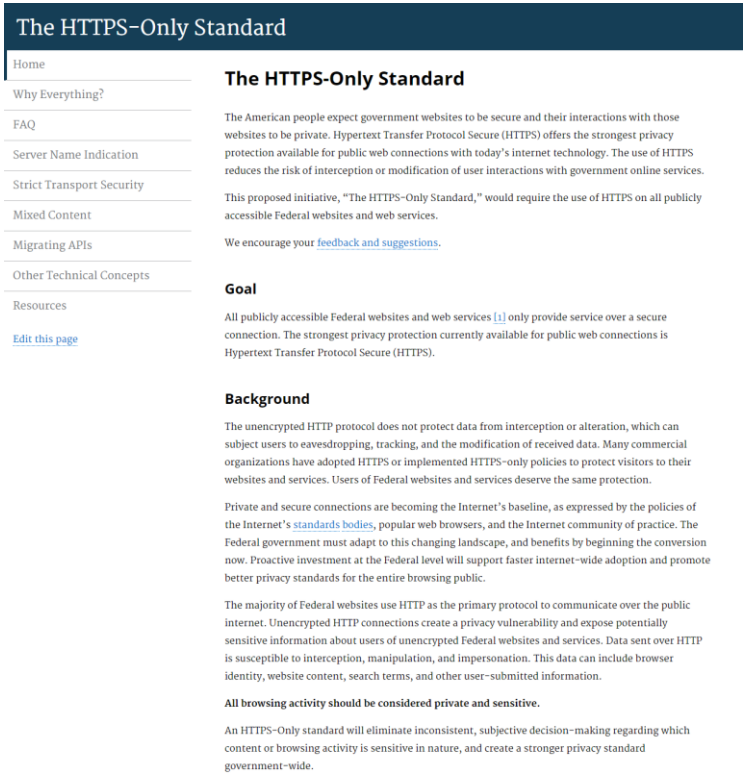
た上で速報的に提示します。内容は、エグゼクティブサマリ、攻撃手法の概要（詳細な手法は除く）、情報のソースへのリンクからなります。

- [2015/3/04] SSL/TLSへの弱い暗号を利用したFREAK Attackについて
- [2014/12/12] TLS 実装上の問題による POODLE attack の影響について
- [2014/10/15] SSLv3仕様そのものに対する POODLE attack について
- [2014/8/08] Black Hat 2014 USAで発表されたSSL/TLSへの複数の新たな攻撃手法について
- [2014/6/06] OpenSSLに関する7つの問題点とその対策について
- [2014/6/05] GnuTLSの暗号ライブラリにおける問題について
- [2014/4/18] OpenSSLにおける Heartbleed Bugと呼ばれる脆弱性について
- [2014/3/13] TLSにおける新たな攻撃の報告

報告書

"The HTTPS-Only Standard"

- "All publicly accessible Federal websites and web services [1] only provide service over a secure connection. The strongest privacy protection currently available for public web connections is Hypertext Transfer Protocol Secure (HTTPS)."
  - https://https.cio.gov/



The HTTPS-Only Standard

Home
Why Everything?
FAQ
Server Name Indication
Strict Transport Security
Mixed Content
Migrating APIs
Other Technical Concepts
Resources
Edit this page

The HTTPS-Only Standard

The American people expect government websites to be secure and their interactions with those websites to be private. Hypertext Transfer Protocol Secure (HTTPS) offers the strongest privacy protection available for public web connections with today's internet technology. The use of HTTPS reduces the risk of interception or modification of user interactions with government online services.

This proposed initiative, "The HTTPS-Only Standard," would require the use of HTTPS on all publicly accessible Federal websites and web services.

We encourage your feedback and suggestions.

Goal

All publicly accessible Federal websites and web services [1] only provide service over a secure connection. The strongest privacy protection currently available for public web connections is Hypertext Transfer Protocol Secure (HTTPS).

Background

The unencrypted HTTP protocol does not protect data from interception or alteration, which can subject users to eavesdropping, tracking, and the modification of received data. Many commercial organizations have adopted HTTPS or implemented HTTPS-only policies to protect visitors to their websites and services. Users of Federal websites and services deserve the same protection.

Private and secure connections are becoming the Internet's baseline, as expressed by the policies of the Internet's standards bodies, popular web browsers, and the Internet community of practice. The Federal government must adapt to this changing landscape, and benefits by beginning the conversion now. Proactive investment at the Federal level will support faster internet-wide adoption and promote better privacy standards for the entire browsing public.

The majority of Federal websites use HTTP as the primary protocol to communicate over the public internet. Unencrypted HTTP connections create a privacy vulnerability and expose potentially sensitive information about users of unencrypted Federal websites and services. Data sent over HTTP is susceptible to interception, manipulation, and impersonation. This data can include browser identity, website content, search terms, and other user-submitted information.

All browsing activity should be considered private and sensitive.

An HTTPS-Only standard will eliminate inconsistent, subjective decision-making regarding which content or browsing activity is sensitive in nature, and create a stronger privacy standard government-wide.

国内の政府機関サイト

- 「日本政府機関Webサイト(.go.jp)のTLS対応状況について」
  (Next Internet Technology and Society Project)
  - "httpsで接続出来た機関は調査を行った35機関中19機関でした。しかし接続出来た機関のうち10機関は404や403エラーなどで正しく表示出来ませんでした。"
    (https://awa.sfc.keio.ac.jp/2015/03/04/survey-of-supporting-tls-at-japanese-governemnts-website/)

- 「go.jpドメインのHTTPSサイトの状況について私もみてみました(2015年3月4日時点)」
  - "イベントで一時的に立ち上がってたサイトや停止したサーバーなどがあり、インターネットからアクセス可能なgo.jpドメインのHTTPSサイトは882中、722であり、 今回は722のgo.jpドメインサイトを調査対象としました。"
    (http://blog.livedoor.jp/k_urushima/archives/1763144.html)

Secure by Default (ex. Google Chrome)

- "Marking HTTP As Non-Secure"
  (https://www.chromium.org/Home/chromium-security/marking-http-as-non-secure)
  - "gradually change their UX to display non-secure origins as affirmatively non-secure. "
  - (まだ試験版(Canary)の話で、さらに設定も必要)
- "Gradually sunsetting SHA-1"
  (http://googleonlinesecurity.blogspot.jp/2014/09/gradually-sunsetting-sha-1.html)

この接続ではプライバシーが保護されません

攻撃者が、[redacted]**.go.jp** 上のあなたの情報（パスワード、メッセージ、クレジット カード情報など）を不正に取得しようとしている可能性があります。

詳細情報を表示しない

セキュリティで保護されたページに戻る

このサーバーが[redacted]**go.jp** であることを確認できませんでした。このサーバーのセキュリティ証明書は[redacted]から発行されています。原因としては、不適切な設定や、悪意のあるユーザーによる接続妨害が考えられます。

[redacted].go.jp にアクセスする（安全ではありません）

NET::ERR_CERT_COMMON_NAME_INVALID

インターネット基盤技術の課題

信頼できる基盤技術とは？

- 我々は
  なにを信頼していたのか？

なぜバラバラにやってるのか？

- サイロ化が一番いけない分野なのに…

脆弱性発見時の課題

"世界平和活動"すぎる

非効率すぎる

コストがかかりすぎる

あまりにもプロセスや体制が成熟していない

形式検証の重要性

形式検証というプロセスそのものが重要	• 頭を少し使うだけで見つかる
最初から検証されていることが望ましい	• ダメなサイクル • 仕様⇒ 実装⇒ 普及(制御不能)⇒ パッチ⇒ 仕様⇒... • 人間はスケールしない
しかし、後からでも手を動かすべき	• 安全な領域を少しでも増やし未来に繋ぐ

実装の重要性

「実装してから仕様を書いて欲しい」

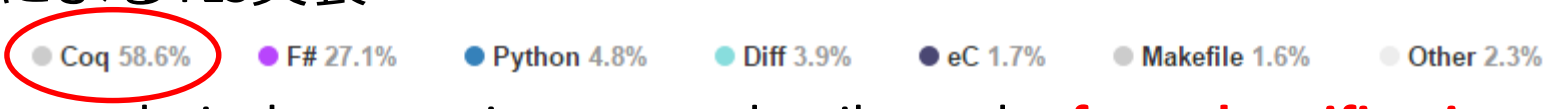
「そうすれば実装が複雑で不具合が出そうな部分がわかるはず」

「TLSぐらいの複雑さのプロトコルをCoqでいじるには先に普通に実装した経験がないとつらいと思う」

FREAK Attackの注目点

技術的・工学的な
アプローチで
安全性向上が可能！

- 発見者：miTLS Team
  - "miTLS is a joint project between Inria and Microsoft Research."
- miTLS "A verified reference TLS implementation"
  (https://www.mitls.org/)
  - F#によるTLS実装



  - "The technical report gives more details on the **formal verification** of miTLS and its cryptographic model."
  - "State-machine attacks [2015]
    Early CCS Attack and SMACK Attack are prevented as the **type-based proof** for miTLS guarantees that its state machine conforms to its logical specification."
    (https://www.mitls.org:2443/wsgi/tls-attacks)

今後の暗号通信の変化

TLS 1.3 (IETF TLS WG)

"IP Stack Evolution Program" (IAB)

"QUIC:Quick UDP Internet Connections"

tcpcrypt (IETF TCPINC WG)

Heartbleedはひとつの例にすぎない

今後も同じようなことは高い確率で起きうる

"OpenSSL is written by monkeys" (2009) (peereboom.us)

Wrote in 2014/6

皆、薄々知っていた

しかし、「皆」とは実際には誰なのか？


CCS

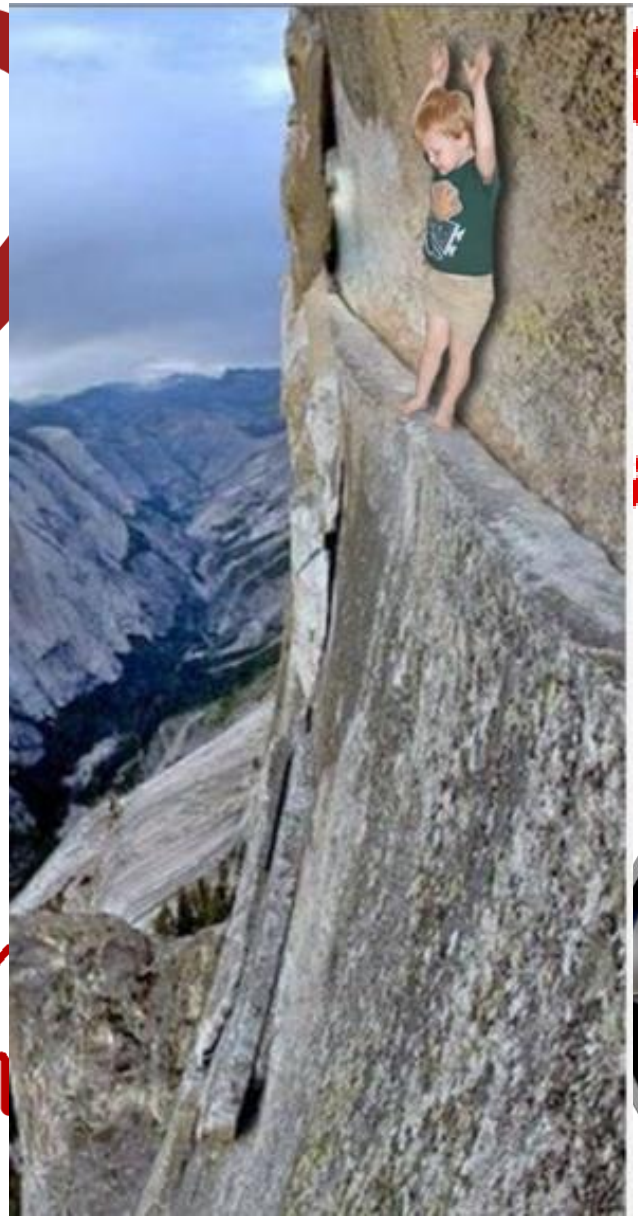

CCS


POODLE SHOULD
KILL SSL 3.0


CCS


POODLE SHOULD
KILL SSL 3.0


PRISM




CCS


POODLE SHOULD
KILL SSL 3.0


PRISM

CCS Injection発見を踏まえて(再掲)

CCS Injectionは
偶然見つかったに
等しい

基盤技術の応用研究に
もっと力を入れる
必要がある

国内には貢献できる
能力を持つエンジニ
ア・研究者が山程いる

信頼と安全のバランス
が崩れている事例

我々はどうするべきか？

インターネットは想定を超えたスケールに

成長してしまったかもしれない

所与のものではなく、

改善し、育て、作り上げていく必要がある

そして、もっと深刻になるべき

その為には、選択と集中、

適切なリソースの投入が必要

我々はどうするべきか？

インターネットは想定を超えたスケールに

成長してしまったかもしれない

改善し　　がある

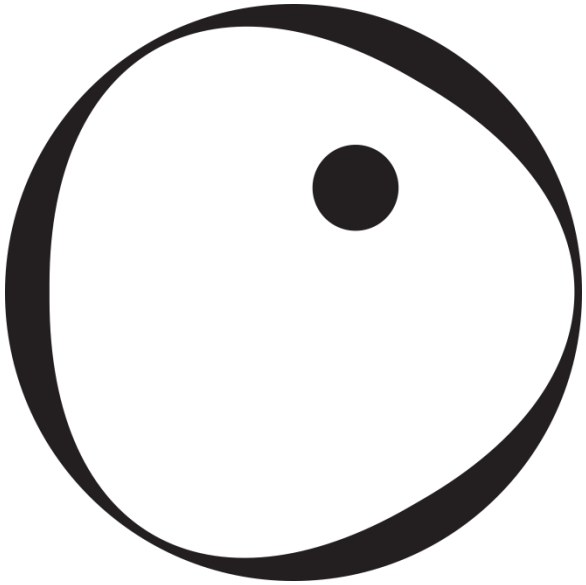
　　を

その為には、選択と集中、

適切なリソースの投入が必要

Let's Work !!

Please contact us



lepidum

https://lepidum.co.jp/ @lepidum

mailto:hayashi@lepidum.co.jp / @lef