様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　２ 年　 ６ 月　 ３ 日現在

機関番号：１２１０２
研究種目：基盤研究(C)（一般）
研究期間：2017～2019
課題番号：１７Ｋ００１７９

研究課題名（和文）IoT環境でマルウェアが実行する耐解析処理の解明

研究課題名（英文）Investigation of anti-analysis operations executed by malware in IoT environments

研究代表者

大山　恵弘（Oyama, Yoshihiro）

筑波大学・システム情報系・准教授

研究者番号：１０３６１５３６

交付決定額（研究期間全体）：（直接経費）　　3,300,000円

研究成果の概要（和文）：主にIoT環境においてマルウェアが自身の解析を妨害するために実行する処理（耐解析処理）を明らかにする研究を行った．第1に，20万以上のマルウェア検体の静的解析により，マルウェアが実行する特定の解析回避処理の傾向を明らかにした．第2に，耐解析処理の1つである長時間のスリープの挙動を分析し，実態を明らかにした．第3に，マルウェアがマルチスレッド実行を用いてサンドボックスや仮想マシンを検出する新しい耐解析処理を構築し，脅威の度合いを示した．第4に，Raspberry Pi上で実行可能な複数の耐解析処理とその脅威の度合いを示した．第5に，例外を発生させるマルウェアのための新しい動的解析手法を開発した．

研究成果の学術的意義や社会的意義
学術的意義は，IoTデバイス上でマルウェアが実行できる，および，実際に実行する耐解析処理が明らかになることである．具体的にはIntel CPUおよびWindows OSが主な対象だった耐解析処理がIoTのハードウェアとソフトウェアからなる環境ではどう変化するかを，人々がより深く理解できるようになる．社会的意義は，マルウェアを効率的に解析，検知，防御するためのセキュリティシステムを企業や研究機関が今後開発する上で大きな助けとなる情報を与えられることである．例えば，耐解析処理の効果を無くすか減じるために，解析システムにどのような機構を組み込むことができるかの指針を与えることが可能になる．

研究成果の概要（英文）：We mainly studied the processing that malware performs to interfere with its own analysis in the IoT environment (anti-analysis processing). First, a static analysis of more than 200,000 malware samples revealed the trend of specific analysis evasion processes performed by the malware. Secondly, we analyzed the behavior of long sleep, which is one of the resistant analysis processes, and clarified the actual situation. Third, we construct a new anti-analysis process in which the malware uses multithreaded execution to detect sandboxes and virtual machines to show the degree of threat. Fourth, we show the multiple resistant analysis operations that can be performed on the Raspberry Pi and their degree of threat. Fifth, we developed a new dynamic analysis method for malware that raises exceptions.

研究分野：ソフトウェア

キーワード：マルウェア　IoT　耐解析処理　仮想化　サンドボックス

様　式　C－１９、F－１９－１、Z－１９（共通）

１．研究開始当初の背景

Internet-of-Things (IoT) は高い利便性を提供するが，セキュリティ面の問題，例えば IoT のサービスやデバイスに存在する脆弱性に対する攻撃の問題も内包している．過去の多くの文献で，インターネットから制御可能な電球や監視カメラなどの各種 IoT デバイスなどについて，攻撃実験の結果などを通じてこの問題が深刻であることが報告されている．

しかし，IoT のセキュリティについてはまだ十分に理解されていない．理解が難しい理由の 1 つは，IoT では多様なハードウェアとソフトウェアが利用されることである．IoT では PC とは異なり，独特なハードウェアやソフトウェアが用いられることが多い．その結果，PC 向けに開発されたセキュリティ技術の多くはそのまま IoT に適用できない．例えば，ARM 系の CPU を用いた IoT デバイスにおいては，Intel x86 の機械語コードを対象に開発されたコード解析技術は使えない．また，IoT デバイスに搭載されるメモリやストレージは一般に貧弱であるため，IoT デバイス上でセキュリティのための機構を動作させることが難しい．PC と多くの点で異なる IoT デバイスにおいて生じるセキュリティ面の問題は，現在も活発に議論されている open problem である．

IoT デバイス向けのマルウェアが出現したら，セキュリティ分野の技術者や研究者はそれらを解析する必要がある．この際には，静的解析に加え動的解析をマルウェアに適用して，動作を詳しく調べることが重要となる．動的解析はマルウェアを実際に実行する解析手法である．しかし，マルウェアが動作対象とする IoT デバイスを解析者が所有しているとは限らない．さらに，解析用ソフトウェアをそのデバイス上で動かせるとは限らないので，必ずしも解析は容易ではない．

より現実的な解析方法の 1 つは，IoT システムを PC 上にエミュレータと呼ばれるソフトウェアによって模擬的に実現し，その上でマルウェアを実行するというものである．この方法であれば，様々な IoT システムを PC 上に仮想的に実現可能であり，解析用ソフトウェアも導入しやすい．しかし，PC 向けマルウェアの中には，解析を妨害するための処理（以下では耐解析処理と呼ぶ）を実行するものが多数存在することが知られている．耐解析処理を組み込んだマルウェアの解析には一般に手間がかかり，耐解析処理に対抗するための機構を導入する必要が生じる．IoT 向けマルウェアにおいても，作者が耐解析処理を組み込むようになる可能性は高いと考える．

Intel CPU で動く Windows OS を対象に作られたマルウェアが実行する耐解析処理については，理解が急速に進んでおり，解析の方法や解析用ソフトウェアも整ってきている．一方，IoT デバイス上でマルウェアが実行する耐解析処理やその効果の度合いについては，ほとんど理解されていない．IoT デバイスを狙うマルウェアに関する知見を急いで集める必要がある．

２．研究の目的

本研究では，IoT デバイス上でマルウェアが実行できる耐解析処理および実際に実行する耐解析処理を明らかにすることを目的とする．IoT デバイス上および IoT デバイスを仮想的に実現するソフトウェア上での実験を通じて，マルウェアが実行可能である耐解析処理，実際に実行する耐解析処理，それらの処理がもたらす効果を明らかにする．具体的には，以下の研究上の問いに答えることを目指す．

- IoT デバイス上で動くマルウェアはどんな耐解析処理を実行可能であり，それによってどのような効果を得るか．
- IoT デバイス上で動く実際のマルウェアはどんな耐解析処理を実行するか．
- IoT デバイス上でマルウェアが実行する耐解析処理の効果を無くすか減じるために，解析システムにどのような機構を組み込むことができるか．

主に対象とする耐解析処理は，動的解析システムの検出および動的解析の困難化とする．そのような処理の例は，ハイパバイザ検出，デバッガ検出，サンドボックス検出，デバイスや OS の特定，長時間のスリープ，挙動の複雑化，挙動のランダム化，ダミー処理の実行である．その結果を元に，耐解析処理の効果を無くすか減じるためにマルウェア解析システムに組み込める機構についても探求し，有用なマルウェア解析機構の実装方式を提案する．

本研究により，実践的見地からは，マルウェアを効率的に解析，検知，防御するためのセキュリティシステムの開発を支援する．学術的見地からは，Intel CPU および Windows OS が主な対象とされてきた耐解析処理が，IoT で用いられる組み込み系のハードウェアとソフトウェアからなる環境ではどう変化するかを明らかにする．

３．研究の方法

まず，研究対象とする分野の IoT デバイスと IoT デバイスを仮想的に実現するソフトウェア（IoT エミュレータ）を入手する．各 IoT エミュレータを研究遂行の観点からの有用性に基づいて比較する．まず候補となるのは QEMU をベースにしたシステムである．並行して，IoT エミュレータで模擬できる実際の IoT デバイスを調達する．まずは CPU は ARM ベース，OS は Linux ベースのデバイスを想定する．

次に，それらの上で動作させて実験で検証することができる耐解析処理を洗い出す．そのため

に，実験対象の IoT デバイスの CPU，その他ハードウェアデバイス，OS，ライブラリの仕様を調査する．例えば CPU については，割り込み機構，レジスタセット，仮想化機構，計時機構，性能カウンタなどの仕様に多様性があることが予想される．OS についても，デバッグのためのライブラリ関数は多様である．これらのような耐解析処理に関係する仕様を，各 IoT デバイスについて調査していく．

その後，それらの耐解析処理を実行するプログラムを開発し，実際に動作させて実行結果や効果を調べる．入手したマルウェアに静的解析と動的解析を適用し，それらが耐解析処理を実行するかどうか，実行するならばどんな種類の耐解析処理であるかを調べる．具体的には，ハイパバイザを検出するプログラム，デバッガを検出するプログラム，サンドボックスを検出するプログラム，デバイスや OS を特定する fingerprinting 処理を実行するプログラムの開発を検討する．加えて，動的解析を困難化する機構が含まれたプログラムを開発も検討する．そのような機構としては，例えば，長時間のスリープ，挙動の複雑化，挙動のランダム化，ダミー処理の実行がある．

並行して，研究者がマルウェア情報を交換する Web サイトなどを通じて，実在のマルウェア検体を収集し，実験に用いる．

ここまでの結果を踏まえ，実際のマルウェアが実行する耐解析処理を解明する．入手したマルウェアに静的解析と動的解析を適用し，それらが耐解析処理を実行するかどうか，実行するならばどんな種類の耐解析処理を実行するかを調べる．静的解析では，逆アセンブル結果やメタ情報をもとに，アクセスされる資源や実行される処理などを特定することが可能である．動的解析では，エミュレータ上でマルウェアを実行して API コール列や CPU 命令の実行ログを取得する．その結果を元に，各種の検出処理の実行の有無や，プログラムの各部分の実行にかかった時間などの情報を集める．

４．研究成果

主に IoT 環境においてマルウェアが自身の解析を妨害するために実行する処理（耐解析処理）を明らかにする研究を遂行し，多くの研究成果を得た．

第 1 に，数千のマルウェア検体の動的解析結果を分析し，マルウェアが実行する耐解析処理の種類をわかりやすく示した．この分析では，多くのマルウェアが検知を試みるハイパバイザや，サンドボックスを検出するために多くのマルウェアが用いる手段についての興味深い知見を得た．

第 2 に，耐解析処理の 1 つである長時間のスリープの挙動を分析し，マルウェアのスリープ挙動に関する実態を明らかにした．さらに，スリープ挙動を用いてマルウェアを分類する新しい手法を構築し，実験でその有効性を実証した．これまでマルウェアのスリープ挙動についてはほとんど理解されておらず，その詳細を示したのは本研究が最初である．

第 3 に，マルウェアがマルチスレッド実行を用いてサンドボックスや仮想マシンを検出する新しい耐解析処理を構築し，その脅威の度合いを示した．同時に，その耐解析処理を備えたマルウェアに対する対策技術の設計と実装を行った．その技術は，マルウェアの耐解析処理による解析時間の長大化を極力防ぎつつ，マルウェアが取得する時間情報を修正することにより，マルウェアによるサンドボックスの有無についての判断を誤らせるものである．マルチスレッド実行を用いた耐解析処理の研究は過去に極めて少なく，その意味で本研究の独自性は高い．

第 4 に，IoT デバイスとして広く用いられている Raspberry Pi 上で実行可能な耐解析処理を複数示し，それらの脅威の度合いについての評価を行った．PC 上に Raspberry Pi 環境を仮想的に実現するソフトウェアを利用し，仮想的な環境と実際の環境におけるプログラムの挙動の違いを観測した．その結果をもとに，ハイパバイザ検出やデバッガ検出を実行するプログラムを実装し，実験を行った．それは OS の基本操作に要する時間の計測によって仮想マシンなどを検出するものである．IoT 環境での耐解析処理についてはこれまで研究が極めて少なく，その意味で本研究の価値は高い．

第 5 に，20 万以上のマルウェア検体に静的解析を適用して，マルウェアが実行する特定の解析回避処理の傾向を明らかにした．具体的にはそれは RDTSC という高精度時刻を取得する CPU 命令を用いる処理である．この分析により，マルウェアが実行時間を計測する処理の多くを解明することができた．また，耐解析処理の効果を減じるために解析システムに組み込める機能も評価した．

他には，OS の見かけ上の時間の流れを速くまたは遅くすることを可能にするハイパバイザの構築手法，動的ライブラリ関数の解決に関する挙動を利用してマルウェアの検知や分類を行う手法，例外を発生させるマルウェアのための新しい動的解析手法，マルウェア検体のデータ欠損がアンチウイルスによるマルウェア同定に与える影響，プログラムファイルがマルウェアか善良ソフトウェアかを，ファイルの表層的な解析で得られるメタデータなどの情報を特徴として機械学習によって判定する手法などについて研究を行い，どれについても重要な研究成果を得た．

５．主な発表論文等

〔雑誌論文〕　計3件（うち査読付論文　3件／うち国際共著　0件／うちオープンアクセス　2件）

１．著者名 Oyama Yoshihiro	４．巻 26
２．論文標題 Investigation of the Diverse Sleep Behavior of Malware	５．発行年 2018年
３．雑誌名 Journal of Information Processing	６．最初と最後の頁 461-476
掲載論文のDOI（デジタルオブジェクト識別子） 10.2197/ipsjjip.26.461	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 －

１．著者名 中村 燎太，大山 恵弘	４．巻 34
２．論文標題 ビヘイビアベースマルウェア検知におけるオンライン機械学習アルゴリズムの比較評価	５．発行年 2017年
３．雑誌名 コンピュータ ソフトウェア	６．最初と最後の頁 156-177
掲載論文のDOI（デジタルオブジェクト識別子） 10.11309/jssst.34.4_156	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 －

１．著者名 Yoshihiro Oyama	４．巻 14
２．論文標題 Trends of anti-analysis operations of malwares observed in API call logs	５．発行年 2018年
３．雑誌名 Journal of Computer Virology and Hacking Techniques	６．最初と最後の頁 69-85
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/s11416-017-0290-x	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

〔学会発表〕　計18件（うち招待講演　0件／うち国際学会　2件）

１．発表者名 大山恵弘
２．発表標題 Raspberry Pi環境におけるステルス性の高い仮想マシン検出
３．学会等名 情報処理学会 第81回コンピュータセキュリティ・第41回インターネットと運用技術合同研究発表会，2018年5月.
４．発表年 2018年

１．発表者名
大山恵弘

２．発表標題
Raspberry Pi環境のためのサンドボックス検出ツール

３．学会等名
日本ソフトウェア科学会第35回大会，2018年8月.

４．発表年
2018年

１．発表者名
大山恵弘

２．発表標題
マルウェアによるRDTSC命令の利用方法についての分析

３．学会等名
コンピュータセキュリティシンポジウム 2018，2018年10月.

４．発表年
2018年

１．発表者名
前田優人，大山恵弘

２．発表標題
動的な関数アドレス解決APIの呼び出しログを用いたマルウェア分類

３．学会等名
コンピュータセキュリティシンポジウム 2018，2018年10月.

４．発表年
2018年

１．発表者名
大山恵弘

２．発表標題
BitVisorによるOSの見かけ上10倍速実行

３．学会等名
BitVisor Summit 7，2018年11月.

４．発表年
2018年

１．発表者名
Yoshihiro Oyama

２．発表標題
Skipping Sleeps in Dynamic Analysis of Multithreaded Malware

３．学会等名
Proceedings of the 2018 IEEE Conference on Dependable and Secure Computing, pp. 164-171, Dec 2018.（国際学会）

４．発表年
2018年

１．発表者名
前田優人，大山恵弘

２．発表標題
関数アドレス取得APIの呼び出しログを用いたマルウェア分類

３．学会等名
情報処理学会 第178回マルチメディア通信と分散処理・第84回コンピュータセキュリティ合同研究発表会，2019年3月.

４．発表年
2019年

１．発表者名
大山恵弘

２．発表標題
マルウェアが実行する耐解析処理の定量的傾向

３．学会等名
日本ソフトウェア科学会第34回大会

４．発表年
2017年

１．発表者名
大山恵弘，中井央

２．発表標題
ベアメタルハイパバイザを用いたネットワークブートシステムの性能評価

３．学会等名
第21回学術情報処理研究集会

４．発表年
2017年

１．発表者名
大山恵弘

２．発表標題
動的マルウェア解析においてスリープ時間を短縮する方式

３．学会等名
コンピュータセキュリティシンポジウム 2017

４．発表年
2017年

１．発表者名
大山恵弘

２．発表標題
大学の教育研究用端末上でのベアメタルハイパバイザの運用

３．学会等名
BitVisor Summit 6

４．発表年
2017年

１．発表者名
大山恵弘

２．発表標題
Raspberry Pi環境におけるステルス性の高い仮想マシン検出

３．学会等名
情報処理学会第81回コンピュータセキュリティ・第41回インターネットと運用技術合同研究発表会

４．発表年
2018年

１．発表者名
Yoshihiro Oyama

２．発表標題
How Does Malware Use RDTSC? A Study on Operations Executed by Malware with CPU Cycle Measurement

３．学会等名
The 16th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2019)（国際学会）

４．発表年
2019年

１．発表者名
小久保博崇，大山恵弘

２．発表標題
マルウェア検体のデータ欠損がマルウェア同定に与える影響の調査

３．学会等名
コンピュータセキュリティシンポジウム 2019，2019年10月．

４．発表年
2019年

１．発表者名
大山恵弘，小久保博崇

２．発表標題
例外を発生させるマルウェアのための動的解析手法

３．学会等名
コンピュータセキュリティシンポジウム 2019，2019年10月．

４．発表年
2019年

１．発表者名
イボットアリジャン，大山恵弘

２．発表標題
動的シンボル情報を用いたLinuxマルウェアの検知

３．学会等名
コンピュータセキュリティシンポジウム 2019，2019年10月．

４．発表年
2019年

１．発表者名
Yoshihiro Oyama, Takumi Miyashita, Hirotaka Kokubo

２．発表標題
Identifying Useful Features for Malware Detection in the Ember Dataset

３．学会等名
The 6th International Workshop on Information and Communication Security (WICS 2019)

４．発表年
2019年

１．発表者名 大山恵弘
２．発表標題 BitVisorによるOSの見かけ上10倍速実行
３．学会等名 BitVisor Summit 7
４．発表年 2018年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考