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 １８ 日現在

機関番号：３４３１５
研究種目：基盤研究(C)（一般）
研究期間：2015～2019
課題番号：１５Ｋ００１１２

研究課題名（和文）コード書換攻撃等による情報漏洩を連携し抑止するセキュアなコンパイラとOSの開発

研究課題名（英文）Developing such a secure system with cooperation of a compiler and an operating system that can deter Information leakage caused by code rewriting attacks

研究代表者
国枝　義敏（KUNIEDA, Yoshitoshi）

立命館大学・情報理工学部・教授

研究者番号：９０１５３３１１

交付決定額（研究期間全体）：（直接経費）　　3,600,000円

研究成果の概要（和文）：　本研究では、任意のユーザプログラムの攻撃耐性を強化し、一言で言えば、攻撃者が目指す情報漏洩に対する最終防衛ラインの形成を図る。具体的には、(1)OSによるユーザプログラムの実行監視を格段に強化させ、システムコールの発行時に、その実行可否を元のプログラムの意味とファイルの権限に遡って検査する機構、(2)機密データには、アクセス権限情報を付加し、同データのコピー時にはアクセス権限情報付きで伝播させ、システムが機密情報を必ず保護できるようにする機構、この2種の機構を実際に具体化し、どう実現するかを研究、提案した。

研究成果の学術的意義や社会的意義
　クラウドコンピューティングの例を出すまでもなく、企業経営、個人情報等にかかる貴重なデータは、コンピュータをネットワークに接続した途端、悪意ある他者にさらされる危険性がある。こうした社会基盤そのものを揺るがすような危険な攻撃に対し、本研究の成果である提案手法は、コンパイラとOSとの連携により、これまでに類を見ないレベルの強力な安全性を獲得することへの具体的な道筋を、その実現可能性も含め示した。これにより、現在のコンピュータシステムの基盤であるシステムソフトウェア群から攻撃耐性を向上させることができ、社会的意義は計り知れないと考えられる。

研究成果の概要（英文）：Through this research, we intended to construct a final defense line against attacks intruding into user application programs. Specifically, we have proposed two types of feasible defense approach, (1) how to detect any spoofing system calls when issued by using the semantics and file permission level of the application, and (2) how to protect confidential information by putting each own access permission on creating data and propagating it on copying the data.

研究分野：セキュアシステムコンパイラ

キーワード：対攻撃耐性向上　セキュアシステム　ディペンダブルシステム　コンパイラ　オペレーティングシステム　システムコールトレース手法　保護ポリシ

様　式　C－１９、F－１９－１、Z－１９（共通）

１．研究開始当初の背景

クラウドコンピューティングの例を出すまでもなく、企業経営、個人情報等にかかる貴重なデータは、コンピュータをネットワークに接続した途端、悪意ある他者にさらされる危険性がある。それとは別にネットワークを介さずとも、不用意な、あるいは、悪意あるユーザにより、USB メモリなどを介して流出するリスクもある。こうした危険性に対処すべく、各種セキュアシステム、ディペンダブルシステムに関する研究が国内外で進められている。科研費の特定領域研究でも過去に「情報爆発時代における安全・安心 IT システム基盤」として重点的に推進されている研究分野である。海外を含む先駆的な具体例としては、セキュア OS に SELinux がある。これは、細分化されたアクセス制限や、ユーザごとの操作権限を事前に設定し実行時に検査する。また、プロセスをサンドボックスと呼ばれる隔離された実行環境で動作させることで、攻撃から防御する OS がある（SoftwarePot など）。この場合隔離されるが故に、たいていの場合実用的でない。一方、コンパイラがセキュアなコード生成を目指す提案も複数ある。例えば、文部科学省リーディングプロジェクト「安全なシステム記述言語および高信頼 OS 記述言語」(リーダー 米澤 明憲; 2003～2007)で開発された高安全 C コンパイラがある。このアプローチは、通常の C 言語の脆弱性を、言語仕様から見直し、高度な型理論を用いた型チェック機能により安全性向上を図る。

本研究のメンバーは、これらとは全く別のアプローチとして、既存プログラムに対する書き換え攻撃に端を発するシステムコールを悪用した乗っ取り等への攻撃耐性向上について、これまで基本ソフトウェアに関する研究を重ねてきており(研究業績欄[2]が中間的まとめで、他[1][3]以外すべて関連実績。例外的な[1][3]は研究代表者がコンパイラ作成実績)、各自の立場から、根本的な対策を議論した。われわれの見る限り、従来のセキュアシステム関連の研究提案は、残念ながら、いずれも OS レベル、 ネットワーク技術レベル、コンパイラが生成する目的コードレベルなどサブシステム単体としての改善策でしか無く、必然的に技術的な限界を内包すると言わざるを得ない。そこで、本研究では、メンバーが協力し合うことで、コンパイラと OS との連携(この着想は古く[4]に依る)により、これまでに類を見ないレベルの強力な安全性を獲得できると着想した。

２．研究の目的

前章の考察・着眼点に基づき、以下の目標を設定した。
任意のユーザプログラムの攻撃耐性を強化し、一言で言えば、攻撃者が目指す情報漏洩に対する最終防衛ラインの形成を図る。具体的には、大きく次の二種類の脆弱性への対抗策としての機能を研究提案し、実証的なプロトタイプを 5 年で Linux を改変する(図 1)。

[機能 1] ユーザプログラムの OS による実行監視を格段に強化させ、特に情報漏洩に関するシステムコールの発行時に、その可否を元のプログラムの意味とファイルの権限に遡って検査する機構を設ける。

[機能 2] 機密データには、アクセス権限情報を付加し、同データのコピー時にはアクセス権限情報付きで伝播させ、システムが機密情報を必ず保護できるようにする。

従って、本研究の対象、主目的は「ユーザプロセスのシステムコールに着目し、不審/不正な挙動を検出し、それに起因する情報漏洩やプロセス乗っ取り、すなわち攻撃者による exec 系コマンドの実行を防止すること」である。

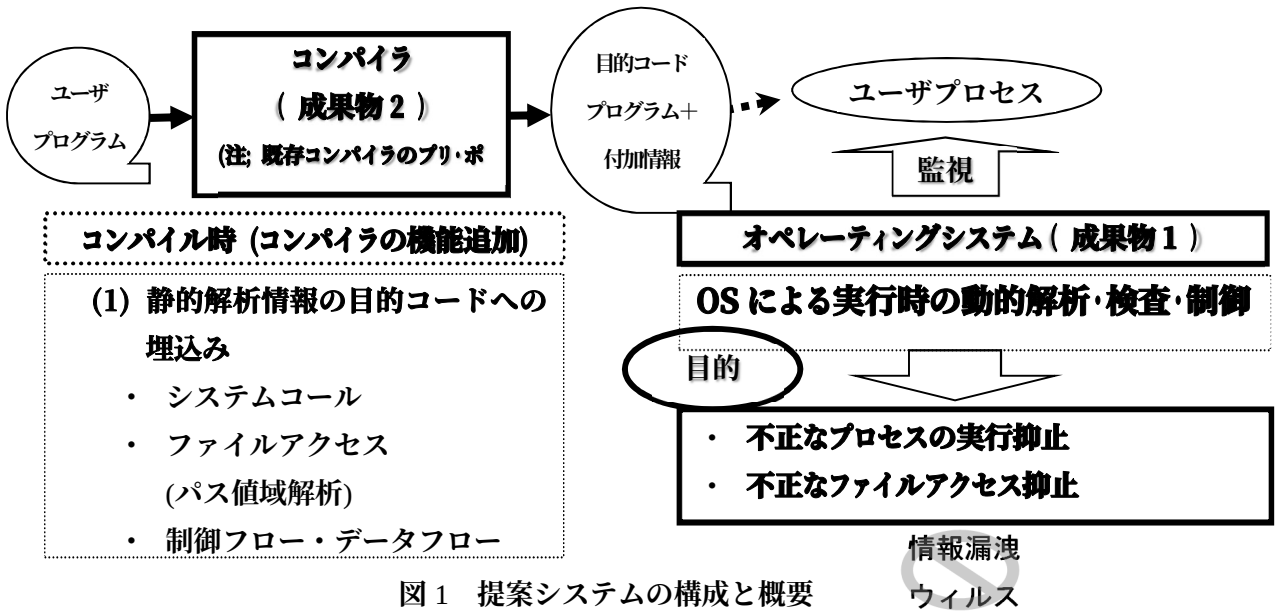

図 1　提案システムの構成と概要

言い換えると､｢元々ソースプログラムに記述された挙動のみしか許さないようにしばりをかけること」である。以下、前述の二つの機能について詳述する。
[機能 1]システムコール監視機能：コンパイラが、ソースプログラムおよび標準ライブラリを解析し、1)呼び出されるシステムコールの種類一覧、2)各システムコール呼び出し時の実引数の値域解析(取り得る値の範囲を解析:環境変数、argv 変数にまで遡る)情報を目的コードとともに出力(OS, 実行系に渡す)。これらの解析は、従来のコールグラフ作成、各種フロー解析を必要な変数に限って重点的に精緻に行うことで可能。OS,実行系は、これらに加え、3)戻り番地による呼び出し元のチェック(return-lib 攻撃に対抗)の三方向から厳重に実行時検査する。こうして、コード書き換え攻撃による本来許されないシステムコール(情報漏洩を図る send, write 系)を遮断。
[機能 2]ファイルアクセス監視機能: 上記コンパイラによるファイル変数やパスに関するフロー解析+値域解析結果を上記同様 OS に渡し、実行時に必ず機密データには、当該データに関するアクセス権限を付随伝播させ厳密に検査する。
本研究の特色は、ウィルスのような組織外からの攻撃から悪意を有する組織内部者の持出し行為までを総合的かつ網羅的に防止できる点である。ここで、既存のセキュア技術、例えばネットワーク侵入検知システム､ウィルススキャナなどは､本研究と組み合わせることが可能で、そうすることでより高い耐攻撃性/信頼性を得ることができる。ファイルパスの値域解析は、全く新規である。
学術的/技術面での特色は、OS とコンパイラが直接連携する点である。すなわち、システムソフトウェアが強力に連携し合う点で、従来にない全く新規かつ独自の発想といえる。こうすることで、本研究提案方式により耐攻撃性の高いセキュアなプログラム実行環境が実現でき、クラウドコンピューティングなど高度に安全性が要求される分野に大きな恩恵をもたらす。特に組込システムでは、一度導入した応用ソフトウェアをバージョンアップすることが容易ではないので、脆弱性が改修されない。今次提案の高信頼性基本ソフトウェアを導入することで、組込システム全般で有用性は顕著である。さらに、今回実装される個々の要素技術は、各検査のオーバーヘッドを低く抑えられる[2]ことと相まって、より一般の OS,実行形に適用可能な学術的知見である。本手法が一般に採用されれば、既存の任意のプログラムが、図 1 のコンパイラで再コンパイルするだけで、簡単に図 1 の OS,実行系の保護対象となって、耐攻撃性が格段に向上する便益が得られる。

３．研究の方法

まずこれまで国枝、毛利が独自に開発してきたプロトタイプシステムの仕様を持ち寄り、すりあわせを行った。コンパイラのコード生成は、gcc のコンパイルオプションでほぼ対応可能であるので、例えば gcc 内部を直接改変する必要は無いことも明かである。期間の制約から、フルコンパイラではなく、必要な情報を抽出・追加するプリ・ポストプロセッサの形とする設計方針を立てた。0S 側の改変は、主にシステムコールの監視部分を追加するだけであるから、通常の unix 系 0S では特に大きな改変は必要無かった。年度ごとに進捗状況を見直し、開発期間の短縮の可能性を追求し、必要に応じ連携研究者、研究協力者を増やしていった。結果的に、下表の当初メンバーに加え、2018 年度まで、研究分担者の毛利公一とともに、研究を行っていた瀧本栄二を加えて増強を図った(「６．研究組織」に既述のとおり)。そして、より柔軟で発展的な開発をめざし、既存研究で組み合わせ可能な手法を取り込んでいく可能性も研究・考察した。
平成 30 年度までは、システムの全体設計・詳細設計を実施し、その設計方針・計画に従って平成 31 年度まで、下記に示すモジュール作成・実装を行い、可能なモジュールから単体 テストも実施する計画であった。1.コンパイラ部(1)手続き・関数コールグラフ … グラフ構造の具体的設計と生成部を実装進行中、部分的に完成。(2)システムコールの解析部 … プログラムのどこでどの種のシステムコールが使われるかを解析するモジュールを実装進行中、部分的に完成。 (3)データフロー解析部 … 既存のモジュールより解析精度を上げる方策を検討し一部アルゴリズム設計し適用可能性を検討したがデータの粒度に関する実装上困難な部分があることが判明。この困難さを解消するために、次章で後述する発展的な展開へ道を拓いていった。(4)解析結果伝達データ形式 … 実行ファイルへ格納するデータの形式を別ファイルの形とすることで開発中。解析結果は、論理的には関数やループを 単位としたグラフになる。これを、小さいサイズかつ 0S が実行時に高速に参照可能とする形式について検討した。2．0S 部(1)システムコールの動的検査部 … システムコールを実行直前でフックし、プロセスの挙動が許可リストに沿っているか否かを検査する機構を実装した試作版一部完成。この機能部分については、研究分担者の毛利公一が、0S の挙動解析の様々な新しい研究分野を開拓し、その実証実験にも使用している。なお、実行ファイル形式の拡張が完了するまでは、目的コードと別ファイルで許可リストを与える手法で進めていた。(2)保護ポリシ形式 … データの伝播範囲を定義する保護ポリシについて、そのポリシ記述法について検討できたので、その検討結果を試作機に実装している。この試作版では、当初計画通り、そのポリシの管理法として、ファイルシステムの i-node のようなファイル管理ブロック内へ格納する方法を検討・実装している。この部分機能に関しても、研究分担者の毛利公一が、0S による動的なデータフロー解析技術とも呼べる新しい研究分野を開拓し、その実証実験にも使用している。

４．研究成果

本研究の二本柱のひとつである OS 部に関して言えば、上述のように、本研究により OS の挙動解析の様々な新しい研究分野を開拓に結びつき、本研究で具体化した様々な要素技術が、それら研究分野での様々な実証実験にも使用できている。また、OS による動的なデータフロー解析技術とも呼べる新しい研究分野へと発展し、本研究で具体化した様々な要素技術が、その実証実験にも利用されている。

一方、本研究の残りの柱であるコンパイラ部に目を移すと、以下の発展へつながっている。

本研究を実施していく段階で、新たなセキュアシステムに関する着想を得ることもできた。具体的には、ユーザプログラムの中で使われるデータに機密度の概念を記述できるようにプログラミング言語を拡張すること。次に、機密度の高いセキュアなデータが、漏洩していないかを、情報流解析によりコンパイラが静的にチェックすること。これにより、プログラマは、自身の書いたプログラムの安全性を機械的に精査させることが可能となる。すなわち、機密情報が漏洩していないか、プログラマが神経をとがらせ、意識しながらプログラム作成する労力を大幅に軽減でき、ある意味実行時のバグ以上に発見が困難である機密情報の漏洩箇所を容易に検出させることができる手法である。この発展に関する研究成果は、「５．主な発表論文等」に含め列挙する。

＜引用文献・参考文献＞

[1] 「形式的に記述された PDE 解法スキームに基づく分布定数系生体機能モデルシミュレーションコード生成システム」, プンザラン フロレンシオ ラスティ, 山下 義陽, 川端 真成, 嶋吉隆夫, 桑原 寛明, 国枝 義敏, 天野 晃, 生体医工学, 査読有り, Vol.50, No.6, pp.666-674(2013).
[2]「コールスタックの制御データ検査によるスタック偽装攻撃検知」, 冨永 悠生, 樫山 武浩, 瀧本 栄二, 桑原 寛明, 毛利 公一, 齋藤 彰一, 上原 哲太郎, 国枝 義敏, 情報処理学会論文誌, 査読有り, Vol.53, No.9, pp.2075-2085(2012).
[3] “A CellML Simulation Compiler and Code Generator Using ODE Solving Schemes,” Florencio Rusty Punzalan, Yoshiharu Yamashita, Naoki Soejima, Masanari Kawabata, Takao Shimayoshi, Hiroaki Kuwabara, Yoshitoshi Kunieda, and Akira Amano, 査読有り, Vol.7, No.11(2012)
[4] The Implementation of a Compiler Controlled Software Distributed Shared Memory System "Fagus" as a Runtime Support System for Automatic Parallelizing Compilers, 齋藤 彰一, 横手 聡, 上原哲太郎, 国枝 義敏, CSREA press, 査読有り, Int. Conf. on Parallel and Distributed Processing Techniques and Applications(PDPTA'2001), Vol.III, pp. 1186-1192 (2001).

５．主な発表論文等

〔雑誌論文〕　計7件（うち査読付論文　2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 奥野 航平, 内匠 真也, 大月 勇人, 瀧本 栄二, 毛利 公一	４．巻 57
２．論文標題 コンパイラを用いた情報フロー制御による情報漏洩防止機構	５．発行年 2016年
３．雑誌名 情報処理学会論文誌	６．最初と最後の頁 2836-2848
掲載論文のDOI（デジタルオブジェクト識別子） なし	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 内匠 真也, 奥野 航平 , 大月 勇人, 瀧本 栄二, 毛利 公一	４．巻 56
２．論文標題 コンパイラとOSの連携によるデータフロー追跡手法	５．発行年 2015年
３．雑誌名 情報処理学会論文誌	６．最初と最後の頁 2313-2323
掲載論文のDOI（デジタルオブジェクト識別子） なし	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 桑原 寛明, 國枝 義敏	４．巻 36
２．論文標題 情報流解析における制約付き機密度パラメータ	５．発行年 2019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 39-45
掲載論文のDOI（デジタルオブジェクト識別子） 10.11309/jssst.36.4_39	査読の有無 無
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 －

１．著者名 吉田 真也, 桑原 寛明, 國枝 義敏	４．巻 36
２．論文標題 オブジェクト指向言語の情報流解析における機密度のパラメータ化	５．発行年 2019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 48-65
掲載論文のDOI（デジタルオブジェクト識別子） 10.11309/jssst.36.48	査読の有無 無
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 －

１．著者名 吉田 真也，桑原 寛明，國枝 義敏	４．巻 34
２．論文標題 情報流解析のためのJavaアノテーション	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 4_47-4_53
掲載論文のDOI（デジタルオブジェクト識別子） 10.11309/jssst.34.4_47	査読の有無 無
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 －

１．著者名 桑原 寛明，國枝 義敏	４．巻 34
２．論文標題 任意の機密度束を用いた情報流解析における非機密化プリミティブの配置	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 2_28-2_38
掲載論文のDOI（デジタルオブジェクト識別子） 10.11309/jssst.34.2_28	査読の有無 無
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 －

１．著者名 桑原 寛明，國枝 義敏	４．巻 32
２．論文標題 情報流解析におけるDeclassifierの配置手法	５．発行年 2015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 136-146
掲載論文のDOI（デジタルオブジェクト識別子） 10.11309/jssst.32.1_136	査読の有無 無
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12件（うち招待講演　0件／うち国際学会　0件）

１．発表者名 桑原 寛明，國枝 義敏.
２．発表標題 情報流解析における制約付き機密度パラメータ
３．学会等名 ソフトウェア工学の基礎 XXV (FOSE 2018)
４．発表年 2018年

１．発表者名
長谷川 健太，桑原 寛明，國枝 義敏.

２．発表標題
Java Stream APIによるストリーム操作の停止性検査のための型システム

３．学会等名
ソフトウェア工学の基礎 XXV (FOSE 2018)

４．発表年
2018年

１．発表者名
長谷川 健太，吉田 真也，桑原 寛明，上原 哲太郎，國枝 義敏.

２．発表標題
JavaのStream APIによるストリーム操作の停止性を検査する型システム

３．学会等名
第20回プログラミングおよびプログラミング言語ワークショップ（PPL2018）

４．発表年
2018年

１．発表者名
松本 隆志，明田 修平，瀧本 栄二，齋藤 彰一，毛利 公一

２．発表標題
情報漏洩防止のためのTCPによるネットワークワイドなテイント追跡手法

３．学会等名
情報処理学会研究報告 2017-CSEC-79

４．発表年
2017年

１．発表者名
長谷川健太、桑原 寛明、上原 哲太郎、國枝 義敏

２．発表標題
Java Stream API によるストリーム操作の停止性検査のための型システム

３．学会等名
第24回ソフトウェア工学の基礎ワークショップ - FOSE2017

４．発表年
2017年

１．発表者名
内西功一、桑原 寛明、國枝 義敏

２．発表標題
グラフDBを用いたプログラム解析の実現に向けて

３．学会等名
第24回ソフトウェア工学の基礎ワークショップ - FOSE2017

４．発表年
2017年

１．発表者名
松本 隆志, 明田 修平, 瀧本 栄二, 齋藤 彰一, 毛利 公一

２．発表標題
動的テイント解析機能を利用したOSによる細粒度データ出力制御手法

３．学会等名
情報処理学会研究報告 CSEC-75

４．発表年
2016年

１．発表者名
安藤 公希、桑原 寛明、上原 哲太郎、國枝 義敏

２．発表標題
Hybrid MPI/OpenMPによる網羅率100%のレインボーテーブル生成の高速化

３．学会等名
コンピュータセキュリティシンポジウム2016

４．発表年
2016年

１．発表者名
兼松 卓也、桑原 寛明、上原 哲太郎、國枝 義敏

２．発表標題
GPGPUによるレインボーテーブル生成の高速化

３．学会等名
コンピュータセキュリティシンポジウム2016

４．発表年
2016年

１．発表者名
桑原 寛明, 國枝 義敏

２．発表標題
情報量に基づく非機密化プリミティブの記述位置候補の順位付け

３．学会等名
ソフトウェア工学の基礎 XXII (FOSE 2015)

４．発表年
2015年

１．発表者名
安藤 公希, 桑原 寛明, 上原 哲太郎, 國枝 義敏

２．発表標題
MPI並列処理によるレインボーテーブル生成の高速化

３．学会等名
コンピュータセキュリティシンポジウム2015(CSS2015)論文集

４．発表年
2015年

１．発表者名
荒木 良仁, 桑原 寛明, 國枝 義敏

２．発表標題
Stream APIを利用するJavaプログラムにおけるストリーム再利用の静的検出手法

３．学会等名
情報処理学会研究報告 (Vol.2019-SE-201)第201回ソフトウェア工学研究発表会

４．発表年
2019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
研究分担者	上原　哲太郎 (UEHARA Tetsutarou) (20273485)	立命館大学・情報理工学部・教授 (34315)	

6．研究組織（つづき）

	氏名 （研究者番号）	所属研究機関・部局・職 （機関番号）	備考
研究分担者	毛利　公一 (MOURI Kouichi) (90313296)	立命館大学・情報理工学部・教授 (34315)	
研究分担者	瀧本　栄二 (TAKIMOTO Eiji) (90395054)	立命館大学・情報理工学部・助教 (34315)	