Vol.2019-DPS-178 No.9
Vol.2019-CSEC-84 No.9
2019/3/4

Processing 言語によるセキュア OS/侵入防御効果の可視化手法

小倉有花[†1]　辻秀典[†1]　橋本正樹[†1]

概要：本研究は，グラフィック機能に特化したプログラミング言語である Processing を用いて，システムが攻撃を受けた際の侵入防御に対してセキュア OS が発揮する効果を可視化する手法を提案する．提案手法ではまず，部分的なメモリ破壊とローカル権限昇格を同時に実行する脆弱性を利用した攻撃を TOMOYO Linux がインストールされたシステムに対して行い，サイバーキルチェーンを基にした脅威モデルを作成して攻撃の進捗段階を%で定義する．その後，使用頻度が高いと考えられるコマンドを中心に合計 4 つのディレクトリから 132 個のコマンドを抽出し，各進捗段階において TOMOYO Linux の強制アクセス制御機能が無効と有効の場合それぞれで実行可能なコマンド数を調べ，その結果を Processing 言語を使って手動で 1 枚の可視化図に表現した．可視化図には，実行可能なコマンドを色付きの丸印で表し，実行不可能なコマンド数を灰色の丸印で表示したところ，攻撃の段階が進行しても，TOMOYO Linux の強制アクセス制御機能が有効の場合には調査した 132 個のコマンドのうち攻撃の影響を受けたコマンドが 3 個に抑えられたことを視覚的に示すことができた．今後は，定性的な自己評価のみに頼っている可視化図の評価について客観的な評価を得るべく改善を行っていく．

キーワード：TOMOYO Linux，効果，Processing 3，可視化

A Visualization Method for Understanding the Effect of Intrusion Detection using the Processing 3

Yuka OGURA[†1] Hidenori TSUJI[†1]
Masaki HASHIMOTO[†1]

Abstract : This research proposes a method for understanding the effect of TOMOYO Linux to intrusion detection with the aid of the visualization image written in the programming language called the Processing 3. In the proposed method, we use a PoC program for a vulnerability that allows local attackers to execute kernel memory corruption and privilege escalation simultaneously, so as to define a progress status of the experimental attack in %, along with a virtual threat model based on the cyber kill chain. In the experiment, we selected frequently used commands from 4 different directories such as /sbin, to investigate the number of executable commands between when TOMOYO Linux is in disabled mode and enforcing mode at each status of the experimental attack. The result of the experiment shows that 129 commands were successfully protected from attackers by TOMOYO Linux when the status of the attack progressed to the next level and we manually visualized the results in Processing 3. We find improving the evaluation of the visualized image to be the future work.

Keywords: TOMOYO Linux, effect, Processing 3, visualization

1. はじめに

1.1 研究の背景と目的

現代社会では情報システムが重要インフラとして浸透しており，人間の日常生活は情報システムなしでは成り立たない．情報システムの爆発的な普及と進化によって私たちが享受できる利便性が飛躍的に向上した一方で，情報システムへのアクセスは容易化・一般化し，コンピュータやサーバ等機器同士の連携もより密になった．それに伴い，特定人物の個人情報から組織固有の営業秘密に至るまでありとあらゆる類の機微な情報が情報システム上で日常的にやり取りされているが，これらの情報資産は常に世界各地からのサイバー攻撃やヒューマンエラーによる漏洩等の危険に晒されているのが現状である．

このような状況においては大切な情報を確実に保護することは困難で，絶対に侵入されないような情報システムを設計することも不可能であると言える．そのため，サイバー攻撃等に遭遇した場合でも攻撃者の意のままにコントロールされることなく，様々な破壊・搾取活動を抑制できるようにすることが大切であり，この目的のために，最小特権の原則を実現するための強制アクセス制御機能を実装した，所謂セキュア OS が開発されている．強制アクセス制御機能の Linux に対する代表的な実装例としては，SELinux，AppArmor，Smack，TOMOYO Linux[1] 等が知られているが，その有用性がある程度認知されているにも関わらず，普及が進んでいるとは言い難い．

セキュア OS は，「出来ないことが増える」という不自由さがあるだけでなく，利用に伴ってポリシーの理解や修正などの管理作業が必要となるため，使いにくいという難点がある．この他にも，例えば，細粒度のアクセス制御でシ

†1 情報セキュリティ大学院大学
Institute of Information Security

Vol.2019-DPS-178 No.9
Vol.2019-CSEC-84 No.9
2019/3/4

ステムに制限をかけるため正しく設定しない限りはシステム停止等のトラブルを誘発する可能性があること，現時点での使用事例が少なく効果も不明確なことなどがあり，これらをまとめると,「ポリシー管理業務の手間を引き受けてまで使用することに果たしてメリットがあるのか」という疑念をユーザに想起させていることが容易に推測できる．このような背景の下，セキュア OS の使いにくさに関しては，ポリシーを見易くしたり解析を容易にしたりするためのインターフェイスの開発などを始めとする，使いにくさの改善に向けてアプローチを行った先行研究が既に存在している一方で， セキュア OS の効果やメリットに焦点を当てて明確にすることに焦点を当てた研究は，現状例が少なく不十分である．

本研究では，セキュア OS の中でも特に TOMOYO Linux に焦点を当て，その効果の可視化を目指す．本研究で可視化の対象とする「効果」については，「TOMOYO Linux の強制アクセス制御機能の無効時と有効時に，システムに侵入した攻撃者が管理者として実行可能であるコマンド数を比較した時の差分」と定義し， この差分の示す値が大きければ大きいほど，攻撃者によるシステム内部の破壊活動を食い止める効果が大きいと考える．また，本研究における可視化とは， 模擬攻撃の進捗段階が 20%， 40%， 60%である場合と TOMOYO Linux の強制アクセス制御機能が無効時と有効時の場合という合計 6 種類の異なる条件下において実施した模擬攻撃の結果を，豊富なグラフィック機能と記述の容易性を両立する Processing 言語を用いて作成した図に表すことにより， 本項で定義する「効果」を視覚的に認識可能となるようにすることである．

1.2 本研究の貢献

本研究の貢献は，セキュア OS による侵入防御の様子を可視化することで，セキュア OS の効果を認識しやすくしたことにある．従来，セキュア OS の課題として取り組まれてきた研究は，使いにくさの改善や機能そのものの強化を目指すものがほとんどであったが，本研究においては，普及が進まない別の原因として，効果が分かりづらいことを仮定し，それをわかりやすくすることを試みたものである．提案した視覚化手法は，プログレスバーや色，記号を用いて，侵入防御の様子を視覚的に把握しやすく検討・工夫を重ねたものであり，当初の目的をある程度は達成できたと考えているが，その評価についてはあくまでも主観的な自己評価のみに頼っているところであり，これについては今後の大きな課題である．本研究を足がかりに，客観的・定量的に評価可能な人間に認識しやすい侵入防御の視覚化手法へと繋がり，それをベースとしたセキュア OS の普及へと繋がることを期待したい．

1.3 本稿の構成

本稿の構成は以下の通りである．すなわち，第 1 章で背景と課題，本研究の目的と貢献について説明し，第 2 章で関連研究を整理する．第 3 章では，提案手法について説明する．その後，第 4 章では模擬攻撃について述べた上で、可視化実験で得られた可視化図を具体的に示す．続いて第 5 章において提案手法に関する評価と考察を述べてから，最後に第 6 章で本稿全体のまとめと今後の課題を述べる．

2. 関連研究

本章では， OS 全般およびセキュア OS に関する研究，可視化に関する研究に分けて関連研究を整理する．

2.1 OS 全般およびセキュア OS に関する研究

橋本らの研究[2]では， 情報セキュリティを担保するための最も基礎的なソフトウェアとして OS を位置づけ， 参照モニタの要件と対応付けながら， 仮想化技術や OS プログラムの検証技術，アクセス制御技術に焦点を当てて近年の研究動向を分類・整理して紹介し，個々の技術に関する今後の研究についての展望や課題を整理している．

原田らの研究[3]では，従来のアクセス制御方式の課題であった，アクセス要求の可否判断のシステムやアプリケーションへの影響および可否判断時点で客体に保存されていた情報の使われ方が考慮されていない問題に対する解決策として，アプリケーションの実行状況を考慮した新たなアクセス制御方式を提案し，その手法の Linux 上での実装である TOMOYO Linux における評価結果を報告している．従来のアクセス制御方式では，アクセス主体であるアプリケーションとアプリケーションがアクセスしようとするファイル等の客体の組み合わせによるアクセス要求の可否判断を行っていたが，原田らの提案手法では，システムが起動されてからアプリケーションが実行されるまでの履歴とアプリケーションのコマンドライン引数やアクセス要求発生時のコマンドライン引数等の様々な情報からアプリケーションの実行状況を解釈し，これらの情報を条件として利用することによってアクセス可否の判断している．提案手法については，不正アクセスや誤操作に伴うリスクを軽減できるだけでなく，典型的な不正アクセス手法の多くに対する効果があることも考察されている．しかし，不正なアクセス要求が発生してから提案手法によって拒否されるに至るまでの具体的な流れに関しては検討対象としていない．

品川の研究[4]では，インターネットを経由して試られる不正アクセスを対象として,研究や開発が行われている OS による不正アクセス防止技術を複数紹介し，これらの不正アクセス防止技術がどのような種類の攻撃に対してどの程度の有効性を持つのかを分類・評価している．既存のいくつかの不正アクセスを受けた際にシステムが被る被害の大

きさを計測する指標を導入してそれを基に評価を行い，これらの不正アクセス技術が，論文で導入した指標のどのような点に対して効果を発揮するかについて報告している．しかし，個々の不正アクセス防止技術が具体的にはどのような情報資産を保護することが可能であるかといったような，不正アクセス防止技術が持つ効果に関する検証実験は実施していない．

2.2 可視化に関する研究

白山の研究[5]では可視化を行う対象と方法に応じて既存の可視化手法を，データの可視化(Data Visualization)・情報の可視化(Information Visualization)・対話型の可視化(Interactive Visualization)の 3 カテゴリに分けて整理し，膨大量のデータを処理することが求められるビッグデータ時代において，人間の手作業による可視化作業を自動化し，効率化するための可視化エージェントを提案･設計している．本研究は，可視化という行為や可視化結果から何が分かるのかという問題意識を起点として，インタラクティブに(人間とコンピュータとの双方向的に進められる可視化プロセスにおいて，可視化した結果 (例えば画像等) がどのように認知されるのかという問題と，可視化プロセス自体の情報化と構造化によってプロセスを半自動化できるかという問題の 2 つを考察 (解決) しようと試みるものである．可視化から分かることは対象依存であり，可視化を必要としないものもあるだけでなく，解釈は個々人に任せるべきと指摘される場合もあること等から，「『可視化から何が分かるのか』という問には，何かが分かるという以外に答えられることはない」と本論文で白山は述べている． 今後の課題としては，文中で紹介した様々な可視化手法やエージェントへの展開は研究が始まったばかりのものも多く，可視化研究の発展のため更に多くの研究が望まれるとしている．

3. 提案手法

本章では，提案手法の概要と可視化に利用する Processing 言語について述べ，具体的な可視化手法について詳細に説明する．

3.1 提案手法の概要

本研究では TOMOYO Linux の制御モードを disabled(無効)の状態に設定した場合と，enforcing(強制)の状態に設定した場合のそれぞれに対して攻撃を行い，それぞれの場合において実行可能なコマンドを比較した結果をプログラミング言語の一つでグラフィック機能に注力した Processing を用いて可視化する．

TOMOYO Linux の強制アクセス制御機能を無効にする場合(disabled)と，有効にする場合(enforcing)の場合において，実行可能であったコマンド/実行不可能であったコマンドを具体的に調べた上で， TOMOYO Linux の強制アクセス制御機能によって実行を制限することに成功したコマンドの数と実行可能であったコマンドの数を可視化結果の図に反映させることによって，攻撃が行われた後の被害拡大の防止効果が一目でわかるようにする．

3.2 Processing 言語

Processing は Java をベースとするビジュアルデザイン等の描画機能に特化したオープンソースのプログラミング言語かつ統合開発環境であり，キャセイ・レアスとベンジャミン・フライによって 2001 年から MIT メディアラボで開発が始まった[6]．当初はプログラミングの基礎を初心者に指導しやすくするための描画ソフトとして活用できるよう開発されたが，その後は利用者層が拡大し，データ可視化，ネットワーク，3 次元の物体の描画等様々な用途向けに Processing コミュニティにより数千を超える大量のライブラリが作られている．また，Mac，Windows，Linux，Android 等の主要な OS に対応しており，記述の容易さという大きな特性も兼ね備えていることから，これまでに多くの利用者によって非常に多岐に亘る作品が作られている．2019 年 2 月現在最新のバージョンは 3．4(2018 年 7 月リリース)である．尚，Processing には，Windows 等一般的な OS 上で動作する Processing ソフトウェアの他，p5．js，Processing for Android，processing．py の大きく分けて 3 種類の派生実装が存在する．Processing のその他の大きな特長として，ドキュメントの豊富さも挙げられる．

3.3 侵入防御効果の可視化主法

1) サイバーキルチェーンと攻撃シナリオ

脆弱性を利用した攻撃に対する TOMOYO Linux の効果を可視化するにあたり，攻撃の進捗段階を定義するため，サイバーキルチェーンをベースとして仮想的な脅威モデルを作成する．サイバーキルチェーンとは，組織が標的型攻撃の脅威に備えるために攻撃者の考え方を分析して多層防御の概念を取り入れて階層化したもので，Lockheed Martin 社の Mike Cleppert により提唱されたフレームワークである[7]．サイバーキルチェーンの各階層はどの組織でも全く同じ構成で一律に決まっているとは言えないが，概ね 7〜8 の段階で構成されている．

本研究では，マクニカ・ネットワークス社のサイバーキルチェーン[8]を参考に，下記の攻撃シナリオを設定する．

① 偵察：ネットワークスキャン・脆弱性情報収集

被害端末 (今回実験を行う端末)と同一のネットワーク内にある別の端末から，ネットワークスキャン等を行うことにより，端末のシステム情報などを入手する．

② デリバリ：USB 接続による不正プログラムの送付

脆弱性を利用した攻撃プログラを USB 接続で， 標的とする端末内に設置する．

③　エクスプロイト：添付ファイルの実行 or 脆弱性を突いた攻撃

カーネルの脆弱性を利用した攻撃プログラムを実行する． 実行に伴って， ローカル権限昇格とメモリの部分的な破壊が同時に行われる．

④　ローカル環境の侵害：ローカル環境の情報収集 or ローカル環境の権限昇格

③で行われた攻撃に伴ってローカル権限昇格を行うことができたため，様々なコマンドを駆使してローカル環境内部のファイルを参照する等して可能な限りの情報を集める．

⑤　永続性の確立：アンチウイルスの無効化　or　ログの削除

今後を想定してアンチウィルスソフトが走っていれば無効化し，その際にファイアウォール等の機能もオフにしておく．また，ログの削除もこの段階で行われる．

⑥　目的の達成：ファイル/データ摂取・外部へのメール送信

目的の情報を探し出し，外部に持ち出す．

2)　可視化図中に表示するプログレスバーの定義

本研究では，攻撃の進捗段階と深刻度を示すプログレスバーを作成する． 特に，ネットワーク内部の動き回りや C&C サーバへの接続等の段階を含む，実際の攻撃を模擬したサイバーキルチェーンをベースとして，そのうちのいくつが達成されたかを以って攻撃の進捗段階(%)を定義することとする．模擬攻撃におけるサイバーキルチェーンは，偵察・デリバリ・エクスプロイト・インストール・C&C・ローカル環境の侵害・内部偵察・感染拡大・永続性確立・目的実行の計 10 段階の階層を有している．

【プログレスバーの定義】

・プログレスバー1：偵察・デリバリが行われた状態．今回の実験で検討した仮想的な脅威モデルでは，PoC プログラム exploit．c のデリバリは USB で行われたものとする．サイバーキルチェーン全体の段階数 10 段階のうち，偵察とデリバリが行われたため，バーの表示は 20%とする．

・プログレスバー2：エクスプロイト・ローカル環境の侵害が行われた状態．今回の実験では，PoC プログラム exploit．c が実行されて部分的なメモリ破壊とローカル権限昇格とが行われた直後の状態．サイバーキルチェーン全体の段階数 10 段階のうち，偵察とデリバリに加えエクスプロイトとローカル権限昇格も行われた状態のため，バーの表示は 40%とする．

・プログレスバー3：永続性の確立・目的の達成が行われた状態．今回の脅威モデルでは， 攻撃者が侵入の痕跡を残すことのないようログの削除を行い，今後の侵入に備えてアンチウィルスソフト(本実験の環境では Clam Antivirus がインストール済) も削除すると想定する．また，攻撃者が目的とするファイルは/etc/shadow および /etc/passwd の 2 つであるとする．偵察・デリバリ・エクスプロイト・ローカル権限の侵害に加えて，永続性の確立と目的の達成も行われた状態であるため，バーの表示は 60%とする．

【プログレスバーの色の濃度】

攻撃の進捗が最終目的 (目的の達成) にどのくらい近いかを示すためバーの色を利用する．バーの色が薄いほど目的から遠く，濃くなるほど目的達成に近い)

【可視化図中に表示する記号の定義とプログレスバーおよびそれぞれの記号への配色の決定】

可視化図中に表示する記号は，シンプルな可視化図を作成する目的で〇のみを使い，〇の 1 個分はそのままコマンド 1 つを表現すると定義する．また，異なる 4 つのディレクトリからコマンドを選び出して調査しているため，ディレクトリが異なるコマンドは異なる色で表現する必要があり，プログレスバーの色彩に関しても進捗段階が進むほど色を濃くすることとし，下記のように色彩を決定した．

色彩の決定にあたっては，認知科学や色彩を扱う学問の観点から一目で識別しやすい色の組み合わせを選ぶため，社会における使い易い色彩環境を目指す特定非営利活動法人カラーユニバーサルデザイン機構や関係分野の研究者らで構成されるカラーユニバーサルデザイン配色セット製作委員会が発行しているカラーユニバーサルデザイン推奨配色セットガイドブック[9]に掲載されている印刷用の見やすい配色に関するページを参考にしている．

4. 模擬攻撃と可視化実験

本章では， 模擬攻撃とそれを受けて実施した可視化実験に関して詳細に述べる．

4.1 実験の概要

本実験では，はじめに，TOMOYO Linux をインストールした仮想マシンに対して，ローカル権限昇格とメモリ破壊を同時に行う模擬攻撃を，強制アクセス制御が無効時と有効時に分けて実施する．その際には，予め決めた 132 個のコマンドをそれぞれの場合に実行し，TOMOYO Linux の強制アクセス制御機能有効時に実行を防ぐことができたコマンドと攻撃後も実行を防止できなかったコマンド(メモリ破壊に影響を受け実行できなかったコマンドも含める)に分類する．その後，分類した結果を，提案手法を用いて手動で可視化する．

尚，本実験で実行対象とするコマンドは， システム管理

Vol.2019-DPS-178 No.9
Vol.2019-CSEC-84 No.9
2019/3/4

者が利用する基本的なコマンドが配置されている/sbin と /usr/sbin，その他，基本的なコマンドが配置されている/bin と/usr/bin の 4 つのディレクトリから使用頻度が高いと考えられるコマンドを予め 33 個ずつ選び，実行対象とした．

4.2 可視化対象とする攻撃

本研究では，模擬攻撃シナリオの内で，「外部からシステムに侵入後，攻撃者が管理者へのローカル権限昇格とカーネルの部分的なメモリ破壊の 2 つを同時に行う PoC プログラムをシステム内部に設置し，同プログラムを実行して管理者となり内部の破壊活動を試みる」を対象に可視化を試みる．そのために，CVE2017-1000111 および CVE2017-1000112 に対応するローカル権限昇格・部分的なメモリ破壊の可能性がある Linux カーネル内のパケットソケットの実装に存在している脆弱性を利用した攻撃を行うプログラムを使用する．同プログラムはその実行と同時にローカル権限昇格とメモリ破壊が行われるプログラムになっており [10]，2017 年 8 月 13 日に Exploit Database に登録された KASLR / SMEP (Linux Kernel < 4．4．0-83 / < 4．8．0-58 Ubuntu14．04 / 16．04) で Andrey Konovalov によって前述の脆弱性を再現できるようにするために作成された Proof of Concept(PoC)の C 言語プログラム 43418．c である[11]．

本実験では TOMOYO Linux が予めインストールされている Ubuntu 14．04 に対して前述のプログラムを用いた攻撃を実施することにより，TOMOYO Linux の強制アクセス制御機能が無効の場合(disabled モード)と有効の場合(enforcing モード)において，図 1 に示すように予め調査対象として選んだ 132 個のコマンドをそれぞれの場合で実行し，実際に実行が可能であったコマンド数の差を比較することで TOMOYO Linux の効果を可視化することを試みる．

	/sbin (33)	/bin (33)	/usr/sbin (33)	/usr/bin (33)
1	blockdev	bash	a2enmod (apache)	apt-get
2	depmod	bzip2	a2dismod (apache)	arch
3	ethtlient	cat	adduser	cd
4	halt	chgrp	addgroup	curl
5	ifquery	chmod	apparmor_status	clear
6	ifconfig	chown	chat	diff
7	ifdown	cp	chroot	dpkg
8	ifup	date	cpgr	find
9	init	dd	cppw	gcc
10	initctl	dir	cron	head
11	insmod	dmesg	deluser	history
12	ip	echo	delgroup	id
13	iptables	egrep	groupadd	lastlog
14	isosize	findmnt	groupdel	last
15	iw	fuser	groupmod	locate
16	logsave	gunzip	grub-reboot	make
17	lsmod	hostname	guest-account	man
18	mdfs	kill	mkdat+found	mlocklop
19	mkmfs	ls	NetworkManager	nmcli
20	modinfo	pidof	newusers	printenv
21	modprobe	mkdir	ownership	pstree
22	mountall	more	service	python
23	poweroff	mount	sshd	rename
24	resolvconf	mv	tcpdump	reset
25	route	netstat	tomoyo-editpolicy	service
26	runlevel	ps	ufw	ssh
27	shadowconfig	pwd	update-grub	service
28	shutdown	ping	update-passwd	top
29	start	rm	update-initramfs	unzip
30	status	sleep	useradd	vi
31	stop	tar	userdel	wget
32	sulogin	touch	usermod	which
33	sysctl	uname	visudo	whoami

図 1　調査対象のコマンド一覧
Figure 1　list of the selected commands

4.3 攻撃実験の結果

模擬攻撃の進捗状況が 20%，40%，60%と異なる 3 つの場合において，TOMOYO Linux の強制アクセス制御機能が無効な場合(disabled モード)と有効である状態(enforcing モード)である時それぞれに対して実験 1～実験 3 を行った．

実験 1 では，模擬攻撃の進捗状況が 20%となっており，すなわち同一ネットワーク内の他端末(本実験では Kali Linux を利用)から実験対象の端末に対する偵察と攻撃に用いる PoC プログラムのデリバリ(USB デバイス経由) が行われた直後である．実験 2 は，模擬攻撃が進み，PoC プログラムが実験対象のシステム状で実行されることによって攻撃者による部分的なメモリ破壊と権限昇格が行われた直後の状態であり，実験 3 では，攻撃者が自身のシステム侵入を示す痕跡を削除して今後の侵入を行いやすくした上で目的とするファイルを探索する段階である．実験 1～実験 3 のそれぞれの場合において，TOMOYO Linux のモードが disabled と enforcing に分けて図 1 に列挙した合計 132 個のコマンドが実行可能であるか不可能であるかを調査した．

実験 1～3 で調査を行った結果，攻撃者によるローカル環境への侵入を受けた場合であっても，TOMOYO Linux による強制アクセス制御機能が有効となっている状態では，ほとんどのコマンドを攻撃者から実行されないように守ることができたことが分かった．また，攻撃の進捗段階が 40%から 60%に進行した場合であっても，TOMOYO Linux が無効・有効のいずれの状態においても攻撃者が実行できるコマンド数に変化はなかった．表 2 は，実験 1～3 で得た調査結果の一覧を整理してまとめたものである．

実験 1 の場合では，攻撃の進捗段階が本研究の定義において 20%であり，この時点で行われた攻撃は偵察と攻撃に利用する PoC プログラムのデリバリのみのため，TOMOYO Linux が無効である場合・有効である場合のいずれであっても調査対象のコマンド(実行ファイル) 132 個は全て実行することが可能であった．また，同じ 20%の進捗段階である場合は TOMOYO Linux の強制アクセス制御機能が有効であっても，132 個のコマンドが全て実行可能であることに変化はなかった．攻撃の進捗段階が 40%の場合，PoC プログラムによるエクスプロイトとローカル環境の侵害が行われたため部分的なメモリ破壊の影響を受けて 6 個のコマンドが実行不可能になったと考えられるが，TOMOYO Linux が無効の場合ではそれらのコマンドを除く合計 126 個が正常に実行できた．一方で，TOMOYO Linux の強制アクセス制御機能が有効の場合は，メモリ破壊を受けた影響で正常に実行できなかったと考えられる 3 個のコ

Vol.2019-DPS-178 No.9
Vol.2019-CSEC-84 No.9
2019/3/4

マンドを除く 129 個のコマンドが TOMOYO Linux の強制アクセス制御機能によって攻撃者が実行できないよう保護することができたことが確認できた．続いて攻撃の進捗段階が 60%の場合にはログの削除や目的のファイルへの操作などが行われた状態であるが，TOMOYO Linux の強制アクセス制御機能が無効・有効のいずれであっても実行可能なコマンド数には変化がなく，攻撃の進捗段階が 40%の時と同じであった．

攻撃進捗 (%)	TOMOYO Linux の強制アクセス制御なし		TOMOYO Linux の強制アクセス制御機能あり	
	実行可能コマンド数	実行不可コマンド数	実行可能コマンド数	実行不可コマンド数
20% (実験 1)	132 個	0 個	132 個	0 個
40% (実験 2)	126 個	6 個	3 個	129 個
60% (実験 3)	126 個	6 個	3 個	129 個

表 2　模擬攻撃の結果
Table 2　results of the experimental attack

4.4 攻撃結果の可視化

前項で得られた実験の結果に基づき，プログレスバーの進捗段階が 20%, 40%, 60%の場合，および TOMOYO Linux が無効の状態と有効の状態の場合の合計 6 つの異なる段階において，「色付き＝実行可能コマンド」，「白色＝攻撃の影響を受けた(と考えられる)コマンド」，「灰色＝TOMOYO Linux の強制アクセス制御機能で攻撃者から保護できたコマンド」の 3 つのパターンに分類し，Processing 言語を利用した可視化図を作成した．尚，可視化図において，どの図が何の状態を示しているかに関しては，図 2 に記す通りである．

図 2　可視化図のレイアウト
Figure 2　Layout of the visualization image

実行可能なコマンドと不可能なコマンドに関しては，計 132 のコマンドをそれぞれ実験 1 から実験 3 において実行して調べることによって得られた 実験 1〜3 の実験結果の表 2 に基づいている．

上記で検討したレイアウトおよび実験結果を記した表に沿って，着色部分 = 実行可能なコマンド・灰色部分 = メモリ破壊の影響もしくは TOMOYO Linux の強制アクセス制御機能など何かしらの理由で) 実行不可能となった(保護された)コマンドという定義の下，Processing 言語を用いて実験結果を可視化したところ，図 3 に示すような可視化図が得られた．

図 3　Processing 言語による可視化図
Figure 3　visualization image by Processing 3

5. 評価と考察

5.1 評価

本研究では，視覚化の効果を測定するために，表 3 に示す評価軸を設定した．表 3 は，この評価軸を用いて，主観的な評価を行った結果である．

評価軸	評価内容
図は直感的に理解しやすく作られているか	△
セキュア OS・TOMOYO Linux に馴染みの薄い人でも理解しやすい図になっているか	○
保護された情報資産の数は分かるか	◎
保護された資産の重要性は分かるか	×
プログレスバーの表示は理解しやすいか	○
図中の記号の意味(形や色など)は直感的に理解できるか	△

表 3　可視化図の評価結果
Table 3　the evaluation of the visualization image

評価は 4 段階で行い，各々，◎(大変分かり易い)・○(まあまあ分かり易い)・△(あまり分からない)・×(分からない)の表記で定性的に示している．

図の全体像に関しては，構成としてはシンプルに作られている一方で，可視化図単体のみでは左右のどちらが TOMOYO Linux の強制アクセス制御機能が無効の場合を表しているのかなどが分からず，別紙の可視化図のレイアウトを参照しなければすぐには分からないため，今後の改善が必要と考えて△の評価とした．続いて図の分かり易さに関する 2 つの項目に関しては，セキュア OS に関する特殊な用語等は図中に登場しておらず複雑な構造の図となっていないため，「まあまあ分かり易い」を意味する○と記した．また，3 番目と 4 番目の評価項目については，具体的に保護された資産数は直感的に理解可能であると考え◎の評価とした一方で，個々のコマンドの中でも特に攻撃者に使用されやすいものなどの「保護された資産の重要性」については表現することができなかったため×とした．プログレスバーに関しては，利用した攻撃の特性から進捗段階が細かく 3 段階に分かれており，色彩の違いもはっきりと区別できるため攻撃の進捗段階が一目で判断可能であると考えたため，評価内容は上から 2 段階目に良い評価である○とした．図中の記号の意味の分かり易さに関しては，コマンド 1 個が○1 個分を表すという簡単なルールのみに従ったという観点では分かり易いと言えるが，図中にその旨の説明を記載できていないことから評価を△とした．尚，今回行ったこれらの評価は自己評価の域を出ず，定量的な評価を行うことができていないため，定性的な評価手法以外の評価方法も検討し確立していく必要がある．

5.2 考察

本研究で行った模擬攻撃では，ローカル権限昇格とメモリ破壊の 2 つを同時に行う攻撃を利用し，TOMOYO Linux による強制アクセス制御機能が無い場合とある場合の 2 つの状況において，具体的にどのようなコマンドが実行できてどのコマンドが実行できなくなるのかの差分を比較し，可視化結果の図を作成した．可視化結果の作成にあたっては，利用する攻撃のプログラムに合わせて攻撃進捗度を定義し，進捗度に合わせて TOMOYO Linux の無効と有効の際にそれぞれどのコマンドが実行可能であるか，132 のコマンドを実際に実行し調べた結果を反映した．可視化を行ったことにより，TOMOYO Linux の利用によってどんなコマンドの実行を防止できたのかを具体的に知ることができ，TOMOYO Linux の強制アクセス制御機能を利用していない場合と比べ，利用した場合には攻撃で権限昇格を行った攻撃者が様々なコマンドを実行してしまうことを阻止できたことから，管理者権限を取得されてもその後に破壊活動を拡大されないよう食い止める効果があったことを可視化結果の図から視覚的に確認することができた．しかし，調査したコマンドの数が現時点では数少ないことや，図の構成が非常にシンプルであるため図の内容を直感的に理解するには情報量が足りない等の問題があり，特に可視化結果の図に対して，他の図や手法を利用してみる等の検討も含めた大幅な改善が急務である．

6. まとめと今後の課題

6.1 まとめ

本研究は，セキュア OS および TOMOYO Linux が実際の攻撃に対して発揮する効果が不明瞭であるという課題に対し，ローカル権限昇格とシステムの部分的なメモリ破壊を行う脆弱性を突く模擬攻撃を TOMOYO Linux インストール済システムに実施した結果を，Processing 言語で可視化する手法を提案した．

可視化にあたっては，可能な限り現実的な攻撃に近い状況において TOMOYO Linux の効果検証を行うべく，実際に報告されたローカル権限昇格 / メモリ破壊脆弱性を使った攻撃を再現可能なプログラムを使い，攻撃の進捗段階をサイバーキルチェーンに従って本研究独自の攻撃想定シナリオを検討した上でシナリオに基づいて 20%, 40%, 60%という 3 種類の進捗段階を定義した．その後，各々の進捗段階において TOMOYO Linux の強制アクセス制御機能が無効と有効の場合に/sbin，/bin，/usr/sbin，/usr/bin の 4 つのディレクトリから使用頻度が高いと考えられるコマンド 33 個ずつ合計 132 個を調査対象として選定し，実行可能か不可能かを判定する実験を行ったところ，攻撃が 20%から 40%に進み実験対象システムが部分的なメモリ破壊とローカル環境の侵害を受けても，TOMOYO Linux が無効の場合は調査した全 132 個のコマンドのうちメモリ破壊に影響を受けたと考えられる 6 個を除く 126 個は管理者権限を取得した攻撃者によって正常に実行可能だったのに対し，有効の場合では調査した全 132 個のうち攻撃者が正常に(メモリ破壊の影響を受けず)実行することができたコマンドは僅か 3 個であった．このことから，攻撃者がシステムに侵入し実際に攻撃を受けた場合でも，TOMOYO Linux の強制アクセス制御機能が有効の場合では攻撃者にコマンドを実行されないよう保護することに有意な効果があることが分かった．可視化を行った祭には実験で検証した効果について Processing 言語を用い，実行可能なコマンドと不可能なコマンドに色分けして可視化図に示すことによって，TOMOYO Linux の強制アクセス制御機能が攻撃者の更なる内部破壊活動の進行に効果を示すことを視覚的に示した．

6.2 今後の課題

①利用する脆弱性の再検討

本研究では，システムの部分的なメモリ破壊とローカル権限昇格を同時に行う脆弱性を利用して攻撃実験を行った

ものの，ネットワーク内を攻撃者が動き回ったりマルウェアをインストールさせて C&C サーバに接続させたりするといったような所謂典型的な標的型攻撃などの現実的な脅威に近い状況を再現することができなかった．そのため，Linux システムを狙うマルウェアを利用するなど，現実的なサイバー攻撃の脅威に対して TOMOYO Linux が持つ効果を可視化できるような模擬攻撃を行うために，利用する脆弱性を再考する必要がある．具体的には，リモート端末からバックドアを作成して侵入する脆弱性など，攻撃が行われた際の被害がより大きなものを使って，模擬攻撃の質を向上させることが今後の課題である．

②可視化図の改善

今回行った可視化では，静的な可視化図を作成し，図中の文字は攻撃の進捗段階を示す%をプログレスバーの横に示すのみとするなど可視化図に表示する文字形態での情報は必要最小限に留めた．その結果，シンプルな可視化図を作成することができた一方で，作成した可視化図 1 枚のみでは，どのコマンドが攻撃の影響を受けた可能性があったり攻撃者から保護できなかったりして逆にどのコマンドであれば確実に保護することができたのかを完全に理解することができないという課題が残っている．本研究で作成した可視化図の内容を理解するには，模擬攻撃の項目で記した「調査コマンド一覧表」が必要となり，「分かり易い可視化」にはなっていない状況であるため，例えば同じ Processing 言語の中でも p5．js などの Java Script と併せて利用できるライブラリを使って動的かつインタラクティブな可視化を試みるなど，より分かり易い可視化を目指していく必要がある．

③評価手法の抜本的な見直し

今回の可視化図の評価は， 定性的かつ主観的な自己評価に留まった．今後は客観的な評価を行えるようにするため，アンケートを作成して多くの人に可視化図を客観的に評価してもらえるようにするために評価軸を更に細かく設定するなど，可能な限り多くの客観的意見を収集するための抜本的な見直しを行っていくことが必要である．また，定量的評価も今回の評価では実施することができなかったことから,定量的な評価手法も併せて確立する必要がある．

④可視化時のコマンドの色分け等の作業の自動化

今回の可視化図の作成にあたっては，模擬攻撃について調査したそれぞれの攻撃の進捗段階と TOMOYO Linux による強制アクセス制御機能が無効/有効時のコマンド数を全て手動で可視化図に反映した．そのため，可視化図の作成過程は非効率的であり，今後本研究を更に発展させていくことを考えた場合，更に多くのコマンドを調べたり，今回は調査したりすることができなかった種類のファイルも調査することを考えた場合は，手動での可視化図作成は膨大な時間を要することとなり，現実的とは言い難い．今後は，可視化図作成を自動化して効率的な可視化を行える方法にも注力して検討する必要がある．

参考文献

[1] “TOMOYO Linux プロジェクト　公式サイト” http://tomoyo．osdn．jp/index．html．ja (参照 2016-02-20).
[2] 橋本正樹・安藤類央・前田俊行・田中英彦， 「情報セキュリティ向上に向けた OS 研究の動向」2012、情報処理学論文誌 コンピューティングシステム Vol．5，No．2，pp．51-62，(Mar． 2012)
[3] 原田季栄・半田哲夫・橋本正樹・田中英彦， 「アプリケーションの実行状況に基づく強制アクセス制御方式」, Vol53、No．9， pp．1-18， 情報処理学会論文誌， 2012
[4] 品川高廣,「オペレーティングシステムによる不正アクセス防止技術」, コンピュータソフトウェア， Vol． 21， No． 6， pp． 482‐493 (2004)． 桜井貴文． 直観主義論理と型理論． 情報処理， 1999， vol． 30， no． 6， p． 626-634．
[5] 白山晋,「可視化から何が分かるのか What can we extract from the visualization?」, システム創成学，第二回学術講演会
[6] “Processing 公式サイト” https://processing．org (2018-12-30 参照)
[7] lockheed martin サイバーキルチェーン“https://www．lockheedmartin．com/en-us/capabilities/cyber/cyber-kill- chain．html” (2018-09-04 参照)
[8] マクニカネットワーク社　サイバーキルチェーン “https://www．macnica．net/solution/security_apt．html/” (2018-11-7 参照)
[9] カラーユニバーサルデザイン推奨配色セットガイドブック第 2 版
[10] “サイオス脆弱性情報” “https://security．sios．com/vulnerability/kernel-security-vulnerability-20170901．html” (2018-08-10 参照)
[11] “Linux Kernel < 4．4．0-83 / < 4．8．0-58 (Ubuntu 14．04/16．04) - Local Privilege Escalation (KASLR / SMEP)”, “https://www．exploit-db．com/exploits/43418” (2018-8-19 参照)