64bit Linux ディストリビューションにおける メモリ破損攻撃対策技術の適用状況調査

三浦向平 [1]　　八槇博史 [2]

概要：バッファオーバーフローに代表されるメモリ破損攻撃には，様々な種類の対策技術が存在する．既存研究においては，3 種類，3 バージョン，32bit 版の Linux ディストリビューションに標準で含まれる，バイナリの対策技術の適用状況の調査や，64bit 版を含めたディストリビューションのバイナリについて調査した研究結果が既に存在している．しかし，これらの研究では，2021 年現在においてリリースされている最新のバージョンを調査対象としていない．ディストリビューションの種類や年代が異なれば適用される防御方法もまた異なるため，本研究で最新のバージョンも含めたうえで新たに調査を行い，4 種類の対策技術の適用状況についてさらなる明確化を行った．

Investigation of Application of Countermeasure Technology against Memory Corruption Attack in 64bit Linux Distributions

KOHEI MIURA[1]　　HIROFUMI YAMAKI[2]

1. はじめに

バッファオーバーフローに代表されるメモリ破損攻撃は，今なお様々なサイバー攻撃に利用されている．これらの攻撃に対し，様々な対策技術が数多く考案され，OS やコンパイラなどに組み込まれ，容易に使用することができる．ただし，近年は新たな攻撃手法も含めて攻撃が多様化しており，これらの対策技術を回避する様々な攻撃も出現している．これらのような複数の攻撃を回避・緩和するためには，複数の対策技術を組み合わせて適用することが望ましい．

様々な種類が存在する Linux ディストリビューションの適用状況や，バージョンによる適用状況の変化を明らかにするため，今日までに様々な調査・研究が行われてきた．本研究では 3 種類，3 バージョンの 64bit 版 Linux ディストリビューションに含まれるバイナリに焦点を当てて，4 種類の対策技術（RELRO，SSP，PIE，Automatic Fortification）の適用状況から高いセキュリティを持つディストリビューションを特定することや，各バージョン間での適用状況の変化を明らかにすることを目的に，調査を行った．

2. 調査対象の対策技術

メモリ破損攻撃への対策技術は様々なものが存在し，OS やコンパイラに組み込まれている[1]．これらを複数個組み合わせて実装することで，攻撃への効果をより発揮する．

1 東京電機大学，システムデザイン工学研究科
Tokyo Denki University，Graduate School of System Design and Technology
2 東京電機大学，システムデザイン工学部
Tokyo Denki University，School of System Design and Technology

本項目では，調査対象である 4 種類の対策技術の概要について述べる．

なお，これら 4 種類の対策技術は，後述する先行研究における調査対象であり，今回の調査結果との比較を行うことで，当時の 32bit 版の状況との比較検討が可能である．

2.1 RELRO (RELocation Read-Only)

RELRO は，GCC のコンパイラオプションとして実装されている対策技術である[1]．実行ファイルの起動時に各セクション，主にライブラリ関数のアドレスを保持する GOT 領域を読み取り専用にする．RELRO には 2 種類の状態が存在し，GOT 領域に読み書きが可能な.got.plt セクションが生成される Partial RELRO と，GOT 領域すべてを読み込み専用にする Full RELRO が存在する．両者ともに実行ファイル内のデータ領域を並べ替えることで，バッファオーバーフローが発生した場合に GOT 領域が上書きされないようにするが，Partial RELRO では GOT 領域に読み書きができるセクションが存在するため，そのような攻撃の対象となりうる．Full RELRO ではそのような攻撃を防ぐことができるが，ファイル実行時にパフォーマンスに多少の影響を与えるという欠点がある[2]．

2.2 SSP (Stack Smash Protector)

SSP は，スタックベースのバッファオーバーフローを検知する対策技術であり，コンパイラである GCC において導入されている[1]．SSP が有効な実行ファイルでは関数を呼び出すときに，スタック領域上のローカル変数とリターンアドレスの間に canary と呼ばれる値を挿入し，関数終了

時にその値が改ざんされていないかを確認する．改ざんがあった場合，プログラムを停止する．SSP を適用する場合はソースコードに手を加える必要はないが，コンパイル時にオプションによる指定が必要である．特徴として，実行ファイルの起動ごとに canary の値が変わるというものがある[3]．canary の値を特定して一度攻撃が成功したとしても次回実行時にはもう一度 canary を特定しなくてはいけないため，攻撃の遅延や抑制につながる．ただし，ブルートフォースなどで canary の値を特定したうえでの攻撃[4]や，canary 自体を偽造・上書きする攻撃も存在するため，注意が必要である．

2.3 PIE (Position Independent Executable)

アドレス参照を相対アドレスにすることで，どんなアドレスに配置されても動作する機械語のコードを PIC（Position Independent Code:位置独立コード）と呼ぶ[5]. PIE は，実行ファイルを PIC のコードのみに変換するコンパイルの形式である．PIE と実行ファイルの各要素のメモリ領域へのアドレス配置をランダム化する OS の防御機能である ASLR(Address Space Layout Randomize)を組み合わせることで，PIE 形式の実行ファイルがアドレス空間のどこに存在しても正常に実行できるようになる．これらの対策技術が機能することで特定のアドレスを使用する攻撃を困難にし，攻撃を緩和することができる．欠点として，Linux の x86 アーキテクチャ, すなわち 32bit 版の Linux ディストリビューション上ではパフォーマンスの低下を引き起こす可能性がある[1].

2.4 Automatic Fortification

Automatic Fortification は，プログラムのコンパイル時にバッファオーバーフローの危険性を持つライブラリ関数を，安全な関数に置換する対策技術であり, GCC のバージョン 4.0 から導入されている[6]．コンパイルおよび実行ファイルの起動時に，書き込み先のバッファサイズと書き込むデータサイズのチェックを行い，リターンアドレスやフレームポインタの改ざんを未然に防ぐ．ただし，ある関数ですでに宣言されているバッファを引数として別の関数に渡し，その別の関数内で引数として渡されたバッファに対して，何らかの文字列をコピーする場合は，上記のチェックは発生せず，攻撃が行われる可能性がある[1].

3. 先行研究

本節では，今回の調査結果の比較対象となる先行研究として，2 つの調査の結果について述べる．初めに，32bit 版の Linux ディストリビューションのうち 3 種類，それぞれ 3 バージョンに標準で含まれるバイナリにおける，対策技術の適用状況に関する菅原らの調査[7]について述べる．表 1 に，調査対象のディストリビューションを示す．

表 1　菅原らの調査[7]での調査対象のディストリビューション

系列	バージョン／リリース日／バイナリ数		
Ubuntu (Debian 系)	Ubuntu10.04 2010-04-29 1,131	Ubuntu12.04 2012-04-26 1,094	Ubuntu14.04 2014-04-17 1,160
CentOS (RedHat 系)	CentOS 5.0 2007-04-12 1,700	CentOS 6.0 2010-11-09 1,432	CentOS 7.3 2016-12-12 1,579
openSUSE (Slackware 系)	openSUSE12.1 2011-11-16 2,033	openSUSE13.1 2013-11-19 2,169	openSUSE13.2 2014-11-04 2,194

この研究では32bit版のLinuxディストリビューションを調査対象としており, 64bit 版のディストリビューションを含めたうえで，それらのバイナリを調査した研究結果[8]も存在する．なお，この研究における調査対象のディストリビューションは次ページの表 2 のとおりである．

以上の二つの研究では，調査によって得られた結果として，主に以下の 3 点を挙げている．

1. ディストリビューション毎に対策技術の適用状況が異なっていた．
2. 対策技術が，バイナリにいったん適用された後，のちのバージョンで意図的に非適用・弱化されたケースが散見された．
3. 対策技術を適用するためにコンパイルオプションが指定されていたが，実際には機能していないケースがあった．これは，対策技術が特定の条件下でのみ適用されることに起因する．

また，2018 年の齋藤らの調査[8]ではさらに，32bit 版と 64bit 版において, バイナリ数や対策技術の適用率に大きな差はないという結果を挙げている．

これらの先行研究の調査対象であるディストリビューションは 2021 年現在においてリリースされている最新のバージョンより古いものである．年代が変わると適用される防御方法も異なるため，最新のディストリビューションについて本研究で新たに調査を行い，適用状況の明確化を行うことは有用である．

4. 調査方法

調査対象の 64bit 版の Linux ディストリビューションとそのバージョン，それぞれのリリース日と調査したバイナリ数を次ページの表 3 に示す．

表 2　齋藤らの調査[8]における調査対象のディストリビューション

系列	バージョン／リリース日 ／64bit 版のバイナリ数 (32bit 版のバイナリ数)			
Ubuntu (Debian 系)	Ubuntu12.04 2012-04-26 1,103 (1,104)	Ubuntu13.04 2013-04-25 1,152 (1,152)	Ubuntu14.04 2014-04-17 1,161 (1,161)	Ubuntu17.04 2017-04-13 1,450
CentOS (RedHat 系)	CentOS 6.3 2012-07-09 1,497 (1,499)	CentOS 6.5 2013-12-01 1,485 (1,487)	CentOS 6.6 2014-10-28 1,492 (1,494)	CentOS7.4 2017-09-14 1,588
openSUSE (Slackware 系)	openSUSE12.2 2012-09-05 2,178 (2,181)	openSUSE13.1 2013-11-19 2,199 (2,199)	openSUSE13.2 2014-11-04 2,228 (2,226)	openSUSE42.3 2017-07-26 2,213

表 3　今回の調査対象のディストリビューション

系列	バージョン／リリース日／バイナリ数		
Ubuntu (Debian 系)	Ubuntu16.04 2016-04-21 1,403	Ubuntu18.04 2018-04-26 1,318	Ubuntu20.04 2020-04-23 1,103
CentOS (RedHat 系)	CentOS 6.10 2018-07-03 1,482	CentOS 7.9 2020-11-12 1,704	CentOS 8.3 2020-12-07 1,570
openSUSE (Slackware 系)	openSUSE13.2 2014-11-04 1,812	openSUSE42.3 2017-07-26 1,809	openSUSE15.2 2020-07-02 1,807

調査に使用した環境は以下の通りである.

- ホスト OS：Windows10
- 仮想環境作成ソフトウェア：VirtualBox6.1
- セキュリティ機構調査ツール : checksec(シェルスクリプト)[9]

今回の調査では，GitHub で配布されているシェルスクリプトである checksec を使用した．このスクリプトはオプションでバイナリのファイル名を指定することで，そのバイナリが持つセキュリティ機構を表示する機能を持つ. また，バイナリが含まれるディレクトリをオプションで指定することで，そのディレクトリ内のすべてのバイナリについて調査および結果表示を行うことができる.

調査対象のディストリビューション毎に以下の手順に沿って調査を行った.

① Windows10 上の VirtualBox アプリケーションを使用して，仮想環境を構築し，checksec を配置する.
② 各ディストリビューションにおいてデフォルトで PATH が通っているディレクトリに対し，checksec スクリプトを実行し，どのバイナリに対してどの対策技術が施されているかを検出する．表示された結果を CSV ファイルに出力する.
③ 出力した CSV ファイルをもとに, 対策技術ごとに適用されているバイナリ数を集計し，グラフを作成する.
④ ディストリビューションの各バージョン間で共通するバイナリの数と適用状況が変化したバイナリの数から，強化または弱化されているバイナリの数を集計し，ディストリビューションごとに表を作成する.

5. 調査結果

本項目では，はじめに調査で作成した適用状況のグラフおよびその特徴を，対策技術別に示す．次に対策技術の適用状況の変化についてのデータをまとめた表を，ディストリビューションの種類ごとに示す.

5.1 RELRO の調査結果

図 1 は RELRO の適用状況を表すグラフである.

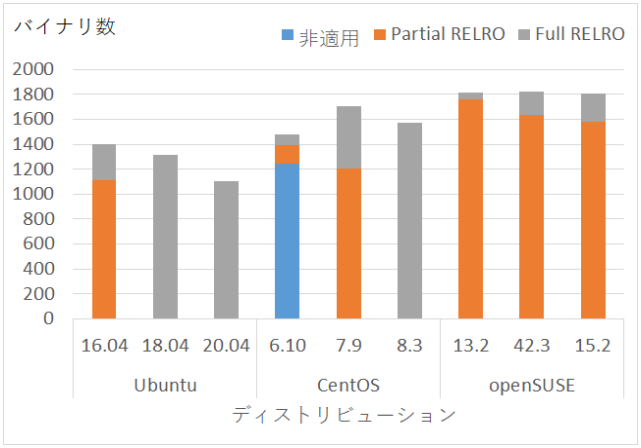

図 1. RELRO の適用状況

Ubuntu16.04 では，Full RELRO が適用されているバイナリは 292 個(20.8%)であったが，Ubuntu18.04 では 1,312 個(99.5%), Ubuntu20.04 では 1,098 個(99.5%)と Full RELRO の割合が大きくなった.

CentOS6.10 では非適用が 1,247 個(84.1%)と大きな割合を占めていたが，CentOS7.9 では 1,203 個(70.6%)が Partial RELRO, CentOS8.3 では 1,569 個(99.9%)が Full RELRO と，バージョンが進むごとに適用率の向上が見られた．

openSUSE は各バージョンにおいて Partial RELRO が適用されているバイナリが多く，13.2 では 1,758 個(97.0%)，42.3 では 1,635 個(89.6%)，15.2 では 1,579 個(87.4%)に Partial RELRO が適用されているという状況であった．こちらは，他のディストリビューションにおけるような，大幅な適用の進捗は調査時点では見られていない．

5.2 SSP の調査結果

図 2 は SSP の適用状況を表すグラフである．

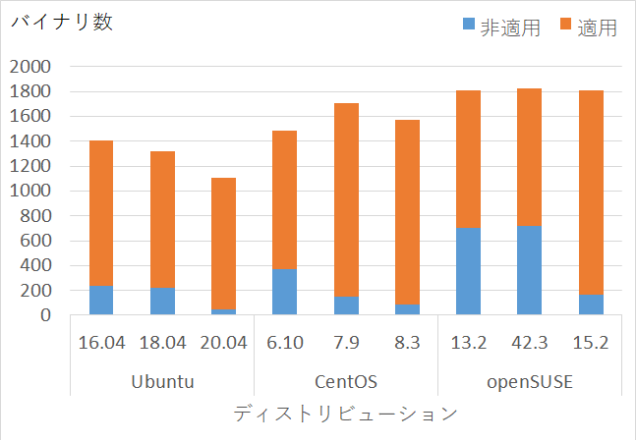

図 2．SSP の適用状況

Ubuntu と CentOS ではすべてのバージョンにおいて適用されているバイナリの数が多く，特に Ubuntu20.04 では 1,054 個(95.6%)，CentOS8.3 では 1,482 個(94.4%)と，両方のディストリビューションともに，一番新しいバージョンにおいて SSP の適用率が最も高かった．

openSUSE においては，13.2 では 1,109 個(61.2%)，42.3 では 1,105 個(60.5%)と，他のディストリビューションと比べて適用率は低かったが，15.2 においては 1,642 個(90.9%)と，ここで適用状況の大きな進展が見られた．

5.3 PIE の調査結果

図 3 は PIE の適用状況を表すグラフである．今回調査した 4 つのグラフのうち，直近での採用が最も進捗した対策技術である．

Ubuntu は 16.04 の時点では適用数は 286 個(20.4%)と非適用の割合が大きかったが，18.04 では 1,313 個(99.6%)，20.04 では 1,099 個(99.6%)と，18.04 において適用率が大幅に伸びている．

CentOS においても 7.9 の時点での適用数は 479 個(28.1%)と多くはなかったが，8.3 では 1,569 個(99.9%)とほぼすべてといってもよい状況に変わっている．

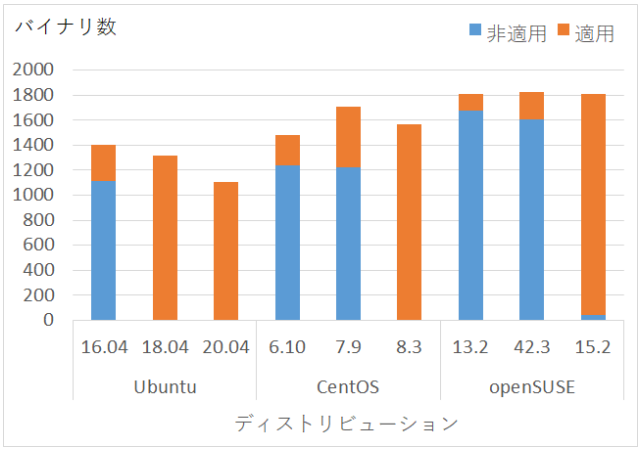

図 3．PIE の適用状況

openSUSE でも同様に，42.3 では 223 個(12.3%)，15.2 では 1,762 個(97.5%)と，15.2 においてはディストリビューションのほぼ全体における適用となった．

5.4 Automatic Fortification の調査結果

図 4 は Automatic Fortification の適用状況を表すグラフである．

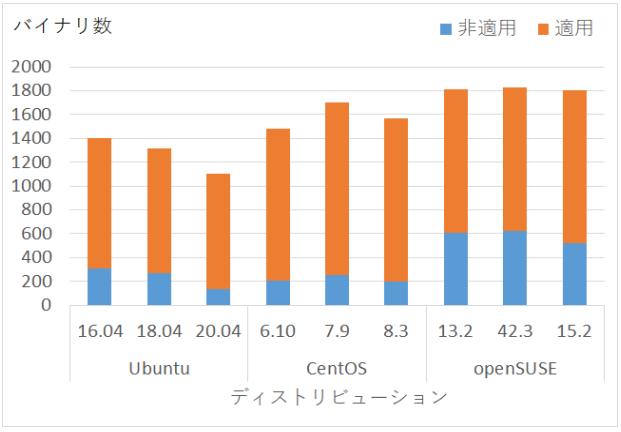

図 4．Automatic Fortification の適用状況

Ubuntu ではどのバージョンでも Automatic Fortification が適用されているバイナリが多く，20.04 の時点では 968 個(87.8%)であった．

CentOS においても 6.10 で 1,274 個(78.2%)，7.9 で 1,447 個(84.9%)，8.3 で 1,375 個(87.6%)と適用状況の大きな変化は見られなかった．

openSUSE では，13.2 と 42.3 の時点ではそれぞれ 1208 個(66.7%)と 1198 個(65.6%)であり，Ubuntu や CentOS と比べると適用率は低いが，15.2 での適用数は 1,289 個(71.3%)と，一定の進捗が見られた．

表 4　Ubuntu における適用状況の変化

	Ubuntu16.04 と Ubuntu18.04				Ubuntu18.04 と Ubuntu20.04			
共通のバイナリの総数	1239				826			
対策技術	RELRO	SSP	PIE	Automatic Fortification	RELRO	SSP	PIE	Automatic Fortification
変化しているバイナリの数	967	17	973	6	0	7	0	7
強化されているバイナリの数	967	14	973	5	0	6	0	3
弱化されているバイナリの数	0	3	0	1	0	1	0	4

表 5　CentOS における適用状況の変化

	CentOS6.10 と CentOS7.9				CentOS7.9 と CentOS8.3			
共通のバイナリの総数	856				1311			
対策技術	RELRO	SSP	PIE	Automatic Fortification	RELRO	SSP	PIE	Automatic Fortification
変化しているバイナリの数	761	149	77	26	957	45	967	27
強化されているバイナリの数	761	146	75	12	957	40	967	20
弱化されているバイナリの数	0	3	2	14	0	5	0	7

5.5 各バージョン間での適用状況の変化

本項目では，各ディストリビューションのバージョン間で共通するバイナリの総数と，対策技術ごとに適用状況が変化したバイナリ数をまとめて，表に示す．以降，非適用から強化または Partial RELRO から Full RELRO への変化を強化，適用から非適用または Full RELRO から Partial RELRO への変化を弱化と呼ぶ．

ただし，Automatic Fortification は 2.4 で述べたような場合や，バイナリのソースコードに脆弱な関数が使用されていない場合には，関数の置換が発生せずに非適用と判断されるため，バイナリが弱化したと判定されてもセキュリティ的に後退したことを明確に示すものではないということに留意する必要がある．

表 4 に，Ubuntu における調査した対策技術の適用状況の変化を示す．

16.04 から 18.04 では，すべての対策技術において強化されたバイナリの数が，弱化されたバイナリの数を上回っていた．特に RELRO と PIE は，適用状況が変化しているすべてのバイナリで強化されている．18.04 から 20.04 では，SSP と Automatic Fortification において適用状況の変化が確認できた．SSP では強化されたバイナリの数が，弱化されたバイナリの数を上回っていたが，Automatic Fortification では反対に，弱化されたバイナリの数が，強化されたバイナリの数よりも多くなっていた．それぞれのバージョン間を比較すると，16.04 から 18.04 のバージョン間のほうが，Automatic Fortification の場合を除き，対策技術の適用状況が変化しているバイナリの数が多かった．

次に，CentOS における調査した対策技術の適用状況の変化を表 5 に示す．

6.10 から 7.9 の場合は Automatic Fortification の場合を除いて，適用状況が変化したバイナリにおいて強化されたバイナリの数が弱化された数を上回っている，特に RELRO においては，変化したバイナリのすべてが強化されており，適用状況の大幅な向上が読み取れた．7.9 から 8.3 の場合は RELRO と PIE において，適用状況が変化したすべてのバイナリで強化されていた．双方の変化したバイナリの数を比較して，6.10 から 7.9 では RELRO が，7.9 から 8.3 では RELRO と PIE が大きく強化されたことがわかる．特に RELRO については，6.10 から 7.9 では非適用から Partial RELRO に，7.9 から 8.3 では Partial RELRO から Full RELRO に強化されたバイナリが多かったために，双方の場合ともに変化した数および強化された数が多くなっている．

最後に，openSUSE における調査した対策技術の適用状況の変化を次ページの表 6 に示す．

13.2 から 42.3 では，RELRO と PIE の場合は強化されたバイナリの数が弱化されたバイナリの数より多く，SSP と Automatic Fortification の場合は弱化されたバイナリの数が強化されたバイナリの数より多かった．特に PIE では適用状況が変化したすべてのバイナリが強化された一方，Automatic Fortification では弱化されたバイナリが強化された数を上回っていた．42.3 から 15.2 においては，Automatic Fortification を除いた対策技術の適用状況において，弱化されたバイナリの数よりも強化されたバイナリの数のほうが多かった．全体を通して，共通しているバイナリ間で PIE が大幅に強化された一方，Automatic Fortification は弱化された数が多かった．ただし，5.4 で述べたようにバイナリ全体での適用率は 13.2 から 42.3 では 1.1%の減少，42.3 から 15.2 では 5.7%の増加となり，大きな変化は発生しなか

表 6　openSUSE における適用状況の変化

	openSUSE13.2 と openSUSE42.3				openSUSE42.3 と openSUSE15.2			
共通のバイナリの総数	1619				1458			
対策技術	RELRO	SSP	PIE	Automatic Fortification	RELRO	SSP	PIE	Automatic Fortification
変化しているバイナリの数	123	33	63	45	16	411	1247	17
強化されているバイナリの数	122	8	63	7	16	409	1247	7
弱化されているバイナリの数	1	25	0	38	0	2	0	10

った．

6. 考察

6.1 PIE の適用状況について

図 3 の PIE の適用状況を見ると，他の対策技術に比べて非適用のバイナリの数が多く，特にそれぞれ最新でないバージョンのディストリビューションにおいて多かった．菅原らの先行研究[7]においては，調査対象の 32bit 版のディストリビューションのすべてで非適用のバイナリが多かった．齋藤らの先行研究[8]では，Ubuntu17.04 を除いた 64bit 版のディストリビューションで非適用の割合が多かった．これらの調査においては，PIE はセキュリティへの効果は高くなく，さらにパフォーマンスの低下につながるので，セキュリティの重要度が高いバイナリにのみ適用されているから全体の PIE の適用率が低くなっているのだと考察されている．文献[1]にも，PIE は Linux の x86 アーキテクチャにおいてパフォーマンスの低下を引き起こすと述べられている．ただし，x86_64 アーキテクチャにおいては顕著なパフォーマンス低下は発生しないともある．

Ubuntu では 16.04，CentOS では 6.10 と 7.9，openSUSE では 13.2 において 32bit 版のディストリビューションが存在する．Ubuntu と CentOS では提供されるディストリビューションが 64bit 版のみとなったタイミングで，PIE の適用状況が大幅に向上している．openSUSE ではそのようなタイミングではないものの，適応状況はディストリビューションが 64bit 版のみになってから向上した．以上より，3 種類のディストリビューションはともに開発を 64bit 版に一本化したのちに，パフォーマンスに及ぶ影響がなくなったために PIE を強化したと考えることができる．

なお，Ubuntu においては Ubuntu17.10 より公式での 32bit 版のディスクイメージの提供が行われなくなった[10]．齋藤らの先行研究[8]においての調査対象である 17.04 の時点では 32bit 版はまだ存在しているが，2016 年の Ubuntu コミュニティのメーリングリスト[11]では 32bit 版のサポートについて議論が行われており，17.04 の時点で開発は 64bit 版のみへの移行を進めていたと考えられる．

6.2 Automatic Fortification の弱化について

図 4 を確認すると，各種ディストリビューションにおける Automatic Fortification の適用率は上昇または 1%程度の減少と，大幅な弱化は見られなかったが，表 4〜6 より，Automatic Fortification に関して強化されたバイナリ数よりも弱化されたバイナリ数の値が上回ったケースが半分以上において見られた．

菅原らの先行研究[7]によると，Automatic Fortification による関数の置換は，書き込み先のバッファサイズおよび書き込むデータサイズによって置換されるケースと置換されないケースがあり，置換されない場合は Automatic Fortification は非適用と判断される．また，今回使用したスクリプト(checksec)では調査するバイナリに関数置換後に追加される_chk を接尾部に持つ安全な関数が存在するかどうかで Automatic Fortification が適用されているかを判断する．

5.4 において既に述べたが，バイナリのソースコードの書き換えによってそのようケースが発生し，弱化したと判定された場合はセキュリティ的に後退したとは言えない．また非適用と判断されたバイナリがどちらかのケースであるか，または Automatic Fortification の適用を指定されなかったかはソースコードを 1 つ 1 つ確認する必要がある．今回の調査範囲からは外れるが，オープンソースであるため検証は可能であり，その点に関しての調査は今後の課題とする．

7. まとめ

本研究では，3 種類の 64bit 版 Linux ディストリビューションのうち 3 つのバージョンを使用して，それぞれに標準で含まれるバイナリについて，メモリ破損攻撃の対策技術の適用状況を調査した．バイナリのセキュリティ機構を調査するシェルスクリプトを使用し，対策技術ごとに適用状況をまとめ，グラフを作成した．また，バージョンの違いによるバイナリの対策技術の適用状況の変化についても調査を行い，ディストリビューションの種類ごとに表にまとめた．

調査の結果として，次ページの 3 つの点を挙げる．

1. ディストリビューション毎に対策技術の適用状況が異なる．
2. それぞれの対策技術およびディストリビューションにおいて，あるバージョン間で適用状況が大幅に向上する傾向がある．
3. 主に Automatic Fortification の適用状況において見られたが，同一のバイナリでも次のバージョンに採用された際に適用されていた対策技術が適用されなくなる場合がある．

調査結果より，バージョンが進むごとに Linux ディストリビューションのセキュリティは高まっているといえるが，バッファオーバーフローをはじめとするメモリ破損攻撃は，以前から継続的に発生しているうえに，今までの攻撃を組み合わせるなどして，様々な攻撃方法も生み出されている．今後新しい対策技術が開発されても，それらを回避する攻撃が生まれると考えられる．我々技術者は，本研究のような現在のセキュリティ機構の適用状況や，攻撃手法などを分析して理解したうえで，使用するディストリビューションの選択や，よりセキュアなプログラムやシステムの構築などを行うことが重要である．

参考文献

[1] 齋藤孝道，鈴木舞音，上原崇史，金子洋平，角田佳史，馬場隆彰，”メモリ破損攻撃への対策技術の調査と分類”，2014，(参照:2021 年 2 月 10 日)
[2] Huzaifa Sidhpurwala ，“Hardening ELF binaries using Relocation Read-Only (RELRO)”，2019，(参照:2021 年 4 月 27 日)
[3] GmS944y，“セキュリティ機構 SSP について（自分用メモ）”，2019，(参照:2021 年 4 月 26 日)
[4] ももいろテクノロジー，“byte-by-byte bruteforce による SSP 回避”，2014，(参照:2021 年 4 月 24 日)
[5] Tanakmura，“PIC,PIE,shellcode,ASLR”，2018，(参照:2021 年 4 月 26 日)
[6] 明治大学情報セキュリティ研究室，“バッファオーバーフローへの対策技術入門”，2017，(参照:2021 年 4 月 22 日)
[7] 菅原捷汰，近藤秀太，渡辺亮平，横山雅展，中村慈愛，須崎有康，齋藤孝道，”Linux ディストリビューションのバイナリにおけるメモリ破壊攻撃の対策技術の適用状況の調査”，2017，(参照:2021 年 1 月 26 日)
[8] 斎藤孝道，菅原捷汰，横山雅展，鈴木嵩人，石渡聖也，須崎有康，”Linux ディストリビューション間での ELF バイナリにおけるセキュリティ対策機構の適用状況の比較”，2018，(参照:2021 年 4 月 30 日)
[9] Brian Davis，”slimm609/checksec.sh: Checksec.sh”，<https://github.com/slimm609/checksec.sh>，(参照:2021 年 1 月 26 日)
[10] kledgeb，”Ubuntu 17.10 その 84 - Ubuntu Desktop 32bit 版のディスクイメージ提供終了へ”，<https://kledgeb.blogspot.com/2017/09/ubuntu-1710-84-ubuntu-desktop-32bit.html>，(参照:2021 年 5 月 1 日)
[11] Canonical，” Installation Media and supportability of i386 in 18.04 LTS Re: Ubuntu Desktop on i386”，<https://lists.ubuntu.com/archives/ubuntu-devel/2016-June/039420.html>，(参照:2021 年 5 月 2 日)