システムコールに基づく異常検知システムの設計と評価

鑪 講平[†] 田端利宏[††,☆] 櫻井幸一[††]

バッファオーバフローを利用した侵入行為を防ぐ目的で，計算機上でプログラムが正常に動作していることをシステムコールの発行履歴から調べる研究がさかんに行われている．本論文では，プログラムの制御フローの性質を考慮して適切な異常検知を行うべく，実験と考察を行う．具体的にはベイジアンネットワークを用いてシステムコールの履歴をモデル化する確率論的手法の有効性を検証する．また，誤検知をなくすための要件について考察する．

The Design and Evaluation of Anomaly Detection System Based on System Call

Kohei Tatara,[†] Toshihiro Tabata[††,☆] and Kouichi Sakurai[††]

In order to prevent attacks exploiting buffer overflow vulnerabilities, there are many researches of checking programs for abnormal behaviors based on history of system calls emitted by them. In this paper, the authors take into account control flow of the programs, and prove an efficiency of a method for modeling history of system calls in a Bayesian Network. We also consider a method for appropriate anomaly detection without false positives.

1. はじめに

今日，計算機の不正利用の事例が数多く報告されている．計算機が乗っ取られると，個人情報の漏洩につながったり，さらなる攻撃への踏み台に利用される恐れもあるため，侵入行為を監視する侵入検知システムの重要性は高い．侵入行為の多くは，バッファオーバフローを誘発させてプロセスの制御を奪う過程を含む[1]．そのため，制御の主体が代わる前後で劇的に変化する事象を，効率良く観測することが解決策となる．

近年，プログラムの制御フローをシステムコールシーケンスとして表現する試みが，異常検知システムの研究で広く行われてきた[2)~6),8)~10)]．異常検知では，訓練期間に監視対象から得られるデータを用いて，監視対象が正常に動作しているかどうかを監視する．すなわち，対象の正常な動作に基づいた特徴抽出が肝要である．そこで，利用者に供与されるプログラムに関する情報の量が，対象の正確なモデル化と検知精度とに大きな影響を与える．また，異常と判断された動作と特定の原因とを関連付けることは難しく，誤検知の要因となっている．システムコールシーケンスはオペレーティングシステム側で容易に利用でき，プログラムのソースコードも必要としない．また，システムコールシーケンスの異常は，バッファオーバフローを利用した侵入行為の発生ととらえることができる．

Forrest らは，システムコールシーケンスを N-gram として扱った[2),3)]．ここで，N-gram とはシステムコール番号を文字とおいた長さ N の文字列を示す．既存の研究では，最適な N の値を求めたり[7)]，N の値を可変にしたりする試みがなされている[5)]．また，確率的手法を用いることにより，N-gram が失う情報量を抑える取り組みもある[6)]．スタックなど，新たな情報を付加して精度を高める研究もある[10)]．

これらは，総じてアルゴリズムの特徴と検知精度との関連を調査することに主眼を置いている．Forrest らの手法では，観測データにおける N 個の連続したシステムコールの局所性を示した．また，N の値を可変にして検知精度を改善する試みは，N-gram ではプログラムの制御フローを完全に表現できない可能性を示唆している．

† 九州大学大学院システム情報科学府
Graduate School of Information Science and Electrical Engineering, Kyushu University
†† 九州大学大学院システム情報科学研究院
Faculty of Information Science and Electrical Engineering, Kyushu University
☆ 現在，岡山大学大学院自然科学研究科
Presently with Graduate School of Natural Science and Technology, Okayama University

Lee らは，RIPPER と呼ばれるルール学習型プログラムを用いた異常検知の精度を評価した[8]．“正常” もしくは “異常” とラベル付けされた N-gram を RIPPER の入力として，If-then 型のルールセットを生成する．ルールセットは “if $p_2 = 104$ and $p_7 = 112$ then the sequence is *normal*”（ここで，$p_i = j$ は N-gram における i 番目のシステムコール番号が j であることを意味する）のような形式をとる．Lee らはまた，同時に N-gram における N 番目と $(N+1)/2$ 番目のシステムコールを予測するという試みを行った．Lee らが行った実験では，University of New Mexico のデータセット[4] を用いて，異常検知が可能なことを示した．彼らは，N-gram における個々のシステムコールは特定の位置にある他のシステムコールとの間に相関性を持つことを発見した．

Eskin らは，sparse prediction tree に基づく sparse Markov transducers を用いたモデル化を提案した[6]．彼らの手法では，N-gram における重複する部分をワイルドカードに置き換え，枝の数を減らしている．そうして得られた sparse prediction tree と，葉の部分に対応する条件付き確率により異常検知を行う．結果的に，Forrest らの手法[2),3)] に比べて精度が高いことを示し，確率的な閾値を用いた手法の有効性を証明した．すなわち，Lee らの成果と同様に確率的な相関性を説明している．

本論文では，Lee らや Eskin らによって明らかになったシステムコール間の相関性の解明に取り組む．具体的には，N-gram における N 番目のシステムコールの決定に，それ以前の $(N-1)$-gram が寄与する割合を確率として評価し，その情報を用いた異常検知の精度を評価する．この結果により，プログラムのコーディングに起因する N-gram の局所性や $(N-1)$-gram の非順序性について理解し，N-gram に基づく異常検知において誤検知が発生する理由について分析する．本論文では，N-gram を用いる異常検知手法の設計と評価を主たる提案としている．また，実験結果の考察を補足する意味で，最後に誤検知をなくす異常検知システムの実現可能性についても言及する．

本論文の構成は以下のとおりである．2 章で，システムコール間の相関性に基づいた異常検知手法について述べる．3 章では，2 章で述べる異常検知手法を用いて実験を行った結果について述べる．4 章では，実験結果の分析と誤検知が発生する原因の分析に基づき，誤検知をなくした異常検知手法について考察する．そして，5 章でまとめとする．

2. N-gram におけるシステムコール間の相関性の定量的な評価

2.1 概　　要

初めに，訓練期間に監視対象のプログラムから得られたシステムコールシーケンスの履歴から N-gram を生成し，その N-gram の情報を用いてベイジアンネットワークを形成する．システムコールは，アプリケーションプログラムが OS の提供するサービスを利用するために用意された関数であり，それぞれにユニークな番号が割り振られている．以後，システムコール S_i という表現は，システムコール番号が i のシステムコールを表す．

ベイジアンネットワーク[12)] は変数間の定性的な依存関係を非循環有向グラフで表したもので，不確実性を含んだ事象や相関関係の表現に適している．変数 X_i と X_j の間の関係を，ベイジアンネットワークでは有向リンクで $X_i \rightarrow X_j$ と表し，X_i は親ノード，X_j は子ノードと呼ばれる．X_i と X_j の定量的相関関係は条件付き確率 $P(X_j|X_i)$ で表される．また，親ノードが複数あるとき，子ノード X_j の親ノードの集合を $\pi(X_j)$ と表す．子ノード X_j について，親ノードのすべての値を条件とする条件付き確率 $P(X_j|\pi(X_j))$ を求めたものは条件付き確率表（Conditional Probability Table: CPT）と呼ばれる．対象の正常な動作は CPT によって表現され，プログラムが発行するシステムコールシーケンスから生成した N-gram の正当性を検証するために用いられる．

監視期間にプログラムがシステムコール X_i を発行した場合，訓練期間に得られた CPT を用いて条件付き確率 $P(X_i|\pi(X_i))$ を求める．一方で，$P(S_j|\pi(X_i))$ が最大になるような j を選ぶ．これを，N-gram のそれぞれのシステムコールに対して行う．最後に，得られた 2 つの条件付き確率の集合に対して，マン・ホイットニーの U 検定と呼ばれる統計的手法を適用して正常か異常かを判断する．マン・ホイットニーの U 検定は，2 群の代表値に差があるかどうかを検定するうえで有用な手法である[13)]．

2.2 訓練期間におけるベイジアンネットワークの形成

(1) システムコールシーケンス $X = \{X_1, \ldots, X_j, \ldots\}$ において，j 番目のシステムコールと，過去 D 回にわたり発行されたシステムコールの集合 $\{X_{j-D}, \ldots, X_{j-1}\}$ の間に依存関係があると仮定する．また，$\pi(X_j) = \{X_{j-D}, \ldots, X_{j-1}\}$ とおく．D の値は，依存度と呼ばれ，システム

child node	parent node
stat	close, open, read
mmap	open, read, stat
close	read, stat, mmap
mprotect	stat, mmap, close
munmap	mmap, close, mprotect
socketcall	close, mprotect, munmap
gettimeofday	mprotect, munmap, socketcall
mprotect	munmap, socketcall, gettimeofday
gettimeofday	socketcall, gettimeofday, close
socketcall	socketcall, munmap, close

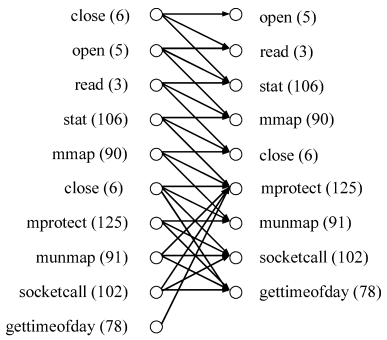

図 1 依存度 $D = 3$ の場合におけるベイジアンネットワークの例（括弧内の数字はシステムコール番号）
Fig. 1 Example of Bayesian network when the degree of dependency D is 3 (Number in the parenthesis expresses system call number).

パラメータとして任意に設定可能である．

(2) すべての $\pi(X_j)$ をベイジアンネットワークのノードとおき，依存関係が認められたノード間を有向リンクで結ぶ．

(3) (1) と (2) の手続きを，すべてのシステムコールシーケンスに対して適用する．

(4) すべての X_j に対して，$P(X_j|\pi(X_j))$ を計算する．

図 1 は ftp クライアントプログラムが発行したシステムコールシーケンスに，上記の手続きを適用してベイジアンネットワークを形成した例である．図 1 における括弧内の数字はシステムコール番号を表している．条件付き確率の値はすべて，これら N-gram の出現頻度から計算される．socketcall システムコールと gettimeofday システムコールの親ノード集合は同じであるため，両者の条件付き確率の値は訓練データにおいて，親ノード集合が現れた回数に比例する．

2.3 監視期間における異常検知の手続き

(1) システムコールシーケンス $X = \{X_1, \ldots, X_i, \ldots\}$ において，i 番目のシステムコールと，過去 D 回にわたり発行されたシステムコールの集合 $\{X_{i-D}, \ldots, X_{i-1}\}$ の間に依存関係があると仮定する．また，$\pi(X_i) = \{X_{i-D}, \ldots, X_{i-1}\}$ とおく．

(2) 訓練期間に得た CPT から $A_i = P(X_i|\pi(X_i))$ を計算する．また，$B_i = P(S_j|\pi(X_i))$ が最大の値をとるような j を求める．これらを，$i, \ldots, i-I+1$ について求め，2 群 $\{A_i, \ldots, A_{i-I+1}\}$ と $\{B_i, \ldots, B_{i-I+1}\}$ とを求める．ここで，I はシステムパラメータとして任意に設定可能である．

(3) 2 群 $\{A_i, \ldots, A_{i-I+1}\}$ と $\{B_i, \ldots, B_{i-I+1}\}$ とに対して，マン・ホイットニーの U 検定[13] を適用する．ここで，帰無仮説は 2 群の代表値に差はないとし，有意水準は異常かどうかを判断するための閾値として任意に設定可能である．

(4) (3) で計算された U 統計量が棄却限界値を下回った場合には，帰無仮説を棄却する．すなわち，システムコールシーケンスを異常と判定する．

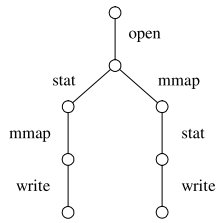

図 2 sparse Markov tree の例（4-gram の場合）
Fig. 2 Example of sparse Markov tree (4-gram).

(2) では，訓練期間に得られた CPT を用いて，以下のように $P(X_i|\pi(X_i))$ を計算する．初めに，$\pi(X_i)$ とのハミング距離が最も小さくなるような $\pi(X)$ の集合を選び，それらが与えられた下での X_i の条件付き確率をそれぞれ求める．次に，その中から最も高い値を持つものを $P(X_i|\pi(X_i))$ として選ぶ．この 2 つの $(N-1)$-gram 間のハミング距離が検知能力に影響を及ぼす．

2.4 関連研究との相違点

Eskin らの手法では，N-gram における N 番目より前の $(N-1)$-gram を逐次的なものとして扱う[6]．たとえば，4-gram として，{open mmap stat write} と {open stat mmap write} が与えられたとき，Eskin らの手法では，**図 2** のような sparse Markov tree が生成される．提案手法では，2 つの 4-gram を同一のものとして扱うため Eskin らの手法に対して異常検知に用いる情報が少ない．一方で，確率論に基づく手法が他の N-gram に基づく手法と異なる点は，N-gram

表 1 実験に用いたデータセット
Table 1 Details of data sets.

Program	# of seq.	# of seq. for training	# of seq. for testing	# of proc.
ftp	181,663	10,000	171,663 (1,358)	5
xlock	9,230,067	895,924	8,334,143 (937)	2
ps	8,589	4,112	4,477 (2,458)	11
login	13,739	6,128	7,611 (4,847)	13
sendmail	143,109	73,491	69,618 (8,286)	34

や状態の決定に確率値を用いるという点である．プログラムが発行したシステムコールシーケンスから，N-gram として特徴を抽出する場合，システムコールの “順序” と “種類” という要素が用いられる．提案手法では，この “種類” という要素のみに焦点を絞って評価を行う．また，確率論に基づく手法は，これにシステムコールが発行される “頻度” という要素を加えることにより異常検知を行うものである．

提案手法は，プログラムの実行によって得られるシステムコールシーケンスの性質を明らかにすることを目的とする．提案手法では，確率値 $P(X_i|\pi(X_i))$ を用いて検定を行う．$\pi(X_i) = \{X_{i-D}, \ldots, X_{i-1}\}$（$D$：依存度）であるため，この D 個のシステムコールからなる D-gram がいかなる順序で配列されても，同一の $\pi(X_i)$ と見なす．$D = N - 1$ とすると，提案手法は N-gram における $(N-1)$-gram を非順序的に扱うことを意味する．3 章では，この操作が異常検知の精度に与える影響を見るための実験を行う．この研究の背景には，$(N-1)$-gram はプログラムにおける分岐構造やループ構造に起因する非順序的な性質を持つという考えがある．

3. 実　　験

本章では，2 章で述べた手法を用いて実験を行った結果を述べる．目的は，他の手法との精度を比較しつつ，提案手法がバッファオーバフローを利用した侵入行為を検知できることを示すことである．

3.1 データセット

Forrest らは，プログラムが正常に動作する際と侵入行為が行われる際におけるシステムコールシーケンスが異なることを実証した[2),3)]．また，それまでに提案された様々な手法を用いて，異常検知の精度を比較することも行った[4)]．この研究において用いられたデータセットは，Web 上にだれでも利用可能な形で公開されている☆．既存の研究との比較のために，我々はこれらのデータセットを用いて実験を行った．

☆ http://www.cs.unm.edu/~immsec/data-sets.htm

データセットはプログラムを正常に利用した際に記録された “live data” と，プログラム実行に際しオプションを注意深く選択することによって得られた “synthetic data” からなる．実験では，これらのデータすべてから適当な数のシステムコールシーケンスを無作為に選び，それぞれ訓練と監視に割り当てた．**表 1** は実験に用いたデータセットの詳細を示す．表にある訓練に用いたデータは，侵入行為が行われる際のデータを含まない．また，監視に用いたデータのうち，括弧内の数字は侵入行為により観測されたシステムコールシーケンスから得られる N-gram の数を表す．ここで，N の値は 6 とした[7)]．

3.2 比較対象

N-gram に基づく異常検知手法は多く提案されている．tide，stide[4)]，や Hofmyer らの手法[3)] は良い成果をあげている．なかでも，我々は検知精度の比較対象として，Hofmyer らの手法[3)] を選択した．Hofmyer らの手法では，N-gram どうしのハミング距離を計算して，その値が閾値を上回っていた場合に異常と判断する．彼らの手法では，プログラムはシステムコールの順序と種類によって特徴付けられている．我々は，システムコールシーケンスを非順序的に扱った提案手法でも同等の結果が得られるかどうかを調査する．

3.3 実験結果

異常検知の精度を比較するために，提案手法と Hofmyer らの手法とで ROC 曲線を描いた．ROC 曲線は，縦軸を True positive rate，横軸を False positive rate として，異常検知システムの検知精度の評価に適している．ここで，True positive rate とは，侵入行為が行われる際に発行されるシステムコールシーケンスの集合のうち，異常と判断される数の割合を表す．また，False positive rate は，正常な動作の際に発行されるシステムコールシーケンスの集合において，異常と判断される数の割合を表す．N の値は 6 として[7)]，提案手法と Hofmyer らの手法におけるパラメータを変化させることにより ROC 曲線を描いた．また，これらの割合は，“live data” と “synthetic data” とを 1 つに集めたものからのデータセットの無作為抽出

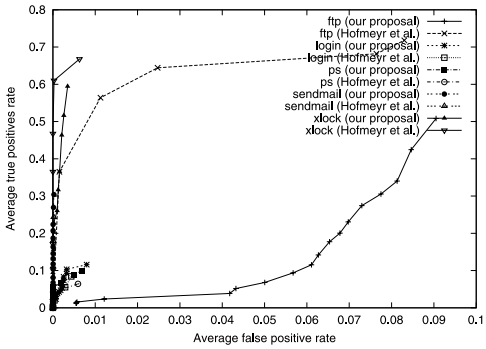

図 3 実験結果（すべてのプログラム）（$N = 6$）
Fig. 3 Result of test (all of the programs) ($N = 6$).

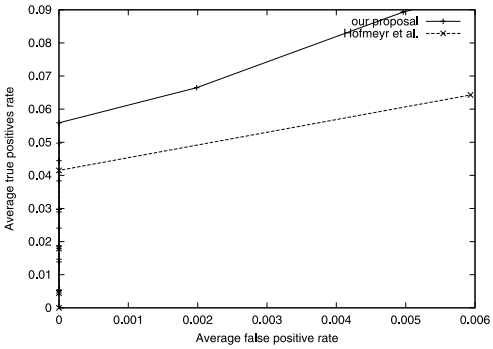

図 5 実験結果（ps）（$N = 6$）
Fig. 5 Result of ps test ($N = 6$).

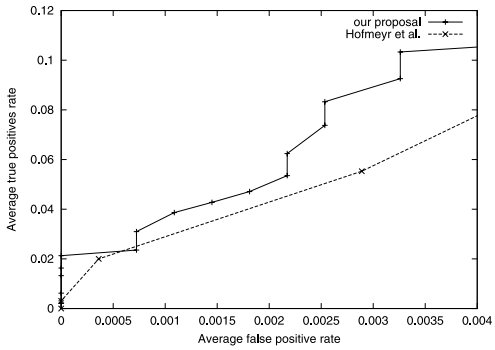

図 4 実験結果（login）（$N = 6$）
Fig. 4 Result of login test ($N = 6$).

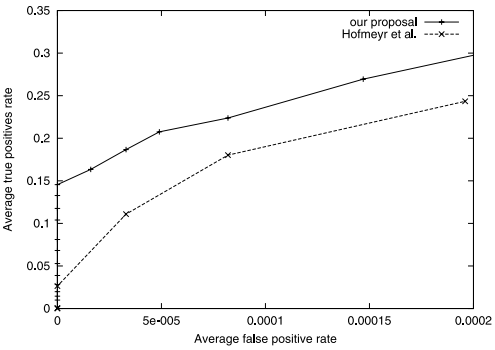

図 6 実験結果（sendmail）（$N = 6$）
Fig. 6 Result of sendmail test ($N = 6$).

を 10 回繰り返して実験を行い，値の平均をとったものを採用した．その結果，**図 3** のような曲線が得られた．図 3 において，ftp 以外のプログラムに関する実験結果を見やすいようにしたものが，**図 4**，**図 5**，**図 6**，**図 7** である．グラフにおける縦軸は，表 1 における侵入行為が行われる際に得られたシステムコールシーケンスを監視した際に，正しく異常と判断された割合を示す．横軸は，正常な実行におけるシステムコールシーケンスを監視した際に，誤って異常と判断してしまった割合を示す．

図 3，7 に見られるように ftp と xlock において，提案手法は Hofmyer らの手法より低い精度を示した．検知精度が下回った原因が，システムコールシーケンスを非順序的に扱ったことによる情報量の減少と，確率を用いた統計的手法を導入したこととのどちらによるものなのかについてはさらなる解析が必要と思われる．その他のプログラムに関しては，適切な閾値の選択により Hofmyer らの手法と同等の検知精度を保つことが分かった．ここで述べるパラメータとは，Hofmyer らの手法における最小のハミング距離を決める閾値 C の値や，提案手法では，D，I，T の値，U 検定における棄却限界値などである．また，N の値を決定するためには，オーバヘッドと検知精度とのトレードオフを考慮する必要がある．実験では，$N = 6$，$D = I = 5$ というパラメータを選択した．グラフは，Hofmyer らの手法と提案手法とにおいて閾値 C と U 検定における棄却限界値を動かすことにより得られたものである．

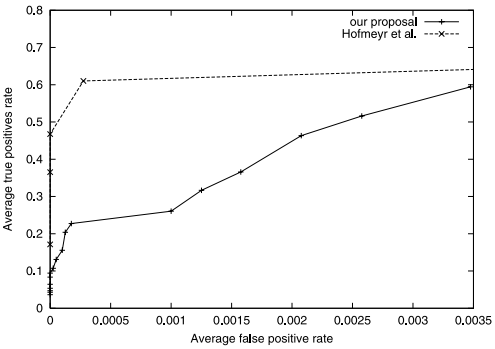

図 7 実験結果（xlock）（$N = 6$）
Fig. 7 Result of xlock test ($N = 6$).

3.4 実験結果の考察

ベイジアンネットワークを用いたモデル化では $(N-1)$-gram におけるシステムコール間の非順序性に着目した．既存の研究のうち，多くは順序性維持の観点から特徴抽出を行う．そのため，OS 上に Σ 種類のシステムコールが存在するとすると，$|\Sigma|^N$ 通りの N-gram を表現できる．提案手法の場合，$N \cdot \frac{(|\Sigma|+N-2)!}{(N-1)!\cdot(|\Sigma|-1)!}$ 通りであり，これは $|\Sigma|^N$ より少ない値である．このことは，異常を検知する割合を下げ，誤検知の割合を上げる要因となりうる一方で，正常な動作データを保存するための領域は少なくて済む．実験からは，非順序性に基づいて異常検知を行っても，検知精度に与える影響は小さいという結果を得た．しかし，システムコールシーケンスを偽装して，検知を回避する可能性もあるため，さらなる評価が必要である[11]．依存度 D が大きい値をとる場合，帰無仮説の棄却限界値の選択範囲が広くなる．すなわち，精度に関して用途に応じた異常検知システムが選択できることを意味する．しかし，オーバヘッドの増加を招くことが考えられる．一方で，依存度 D が小さい場合，選択範囲が狭い．これは，異常検知の精度を下げる恐れがある．

Hofmyer らの手法は，システムコールシーケンスから得られる N-gram の局所性や N-gram におけるシステムコール間の相関性を特徴として用いる．提案手法では，この方針を踏襲する一方で，$(N-1)$-gram を非順序的に扱う点が異なる．しかし，Hofmyer らの手法との比較から，$(N-1)$-gram を非順序的に扱っても，検知精度への影響は小さいことが分かる．本論文では，N-gram を用いる異常検知手法の設計と評価を主たる提案とした．4 章では，これらの性質が観測された原因について調べ，誤検知をなくすための手法の要件について考察する．

4. 能動的な決定性付加に関する考察

4.1 状態遷移の非決定性

訓練期間において得られるデータが，プログラムの制御フローをすべてたどったうえで得られる完全なデータであるという保証はない．また，異常検知システムは分岐処理やループ処理を判断することはできない．このため，特定のシステムコールの発行と，プログラムの制御フローを関連付ける作業は非決定的なものとなり，誤検知を完全になくすことができないと考えられる．

このため，訓練期間に得られるデータに対して，新たな情報を付加する研究が行われている．能動的に付加された情報に基づいて異常検知を行うことによって，誤検知が発生しない異常検知を行うことが可能となる．ここで，付加する情報が満たすべきは，(1) 侵入行為が行われた前後で劇的に変化し，また，検証可能であること，(2) 個々の情報は，プログラムの特定の箇所と関連付けることが可能であること，(3) この情報を用いて生成した N-gram はすべての N-gram 中において，ユニークに識別可能であることなどがある．

Wagner らは，ソースコードを静的解析することにより，プログラムの制御フローを表現する非決定プッシュダウンオートマトンを生成した[9]．Oka らは，Wagner らの方式に，スタックの情報を加えた状態遷移図を生成して，誤検知をなくしている[10]．ほかにも，上記の性質を満足する最も簡単な手法としては，プログラム中に記述されるシステムコールをそれぞれユニークな形に変換し，あらかじめ記録しておくことが考えられる．すなわち，プログラムの他の箇所で発行される同じシステムコールや，訓練データに含まれないシステムコールと区別するために，ユニークな番号を割り振ったり，引数を与えたりする．これにより，変換したシステムコールはプログラムコードの特定の箇所と関連付けることが可能であるため，上記で説明したような非決定性は表れない．異常検知システムは，あらかじめ識別可能であるように変換したシステムコールの発行のみを許可するため，シェルコードを挿入することを前提とした侵入行為を防ぐことが可能となる．

この手法が有効であるためには，プログラムにおける関数やサブルーチン内では，複数のシステムコールが逐次的に発行される箇所が存在することが望ましい．また，バッファオーバフローを利用した侵入行為はプロセスの制御を乗っ取り，攻撃者が意図したシステムコールを発行するものという仮定が必要である．一方で，先の実験によって証明された N-gram の局所性や N-gram におけるシステムコール間の相関性，$(N-1)$-gram における非順序性という性質を利用した異常検知が有効であることから，上記の状況を仮定することは現実的であると考えられる．Oka らはソースコードを静的解析することによってシステムコールの制御フローを決定的に把握したが[10]，ソースコードが手に入らない場合において，(1) (2) (3) の条件を満足するようにシステムコールを変換する方法について説明する．

4.2 提案手法との関連性

提案手法を用いる異常検知 $(N-1)$-gram の非順序性について評価した．これは，システムコールシーケンスにおける “種類” と “頻度” 情報により異常検知を行うことが可能であることを示すものである．すなわ

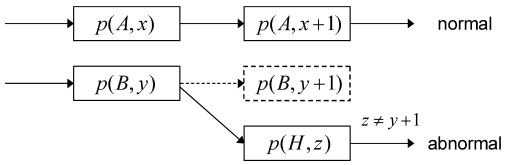

図 8 正常動作時と異常動作時とにおけるプロセスの状態遷移
Fig. 8 State transition of process in normal and abnormal operation.

ち，N-gram における順序的な性質を用いずに，誤検知をなくす可能性について示唆している．このため，実用的な異常検知システム構築のためには，他の性質に基づく異常検知システムの実現可能性について考察することは重要であると考えられる．

4.3 プログラムの実行時の流れ

OS はプログラム実行時に下記のような手順を踏む．

(1) プログラムが発行するシステムコールを，ユニークに識別可能な形に変換する．これは，システムコールの引数にプログラムごとにユニークな乱数やカウンタ値を挿入することによって実現可能である．
(2) 訓練データは，変換したシステムコールに関する情報である．すなわち，異常検知を行うために必要な訓練データはあらかじめ決定されている．
(3) プログラムを実行する．プログラムコードをメモリ上にロードして所定のエントリポイントから実行が始まる．

システムコールの発行時における異常検知の手順は下記のとおりである．**図 8** に提案手法の適用時におけるプロセスの様子を示す．図の $p(i,j)$ は，プロセス i が識別するシステムコールが j であることを表している．ここでは，便宜的に値をインクリメントするにとどめている．$p(A,x)$ の次に発行されるシステムコールは $p(A,x+1)$ となる．

ここで，プロセス B の実行中にバッファオーバフローが発生し，プログラムの制御フローが $p(B,y)$ から $p(B,y+1)$ へと遷移せずに，$p(H,z)$ に遷移した場合を考える．このとき，異常検知システムがあらかじめ登録しておいたプロセス B のシステムコール $y+1$ と，攻撃者がバッファオーバフローによって呼び出したシステムコール z が異なれば，プロセスを異常と判断して終了する．

4.4 実装方法とセキュリティ

システムコールをユニークに識別できるように置き換える操作では，システムコールの引数として新たに乱数を加えることが考えられる．ソースコードが入手可能である場合は，システムコール呼び出しの記述に直接付加することができる．それが叶わない場合，**図 9** のように，システムコールの置き換えをバイナリレベルで可能とするような実装が必要と考えられる．図 9 では，open システムコールの引数に，乱数値（0x12345678）を加えて呼び出している．プログラムの他の箇所で同じシステムコールを呼び出している場合においても，1 つ目の引数の値によってユニークに識別することができる．一方で，OS 側でシステムコールを修正する必要がある．また，ソースコードが手に入らない場合を想定しているため，OS 側でバイナリコードを書き換える必要がある．バイナリコードを書き換える際の注意点として，前後のオペレーションに影響を及ぼさないことがある．すなわち，open システムコールの呼び出しは，my_open 呼び出しに書き換えられる．my_open 呼び出しでは，open システムコールの引数に乱数値（0x12345678）を付加している．

```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $8, %esp
    subl    $8, %esp
    pushl   $0
    pushl   $.LC0
    call    _my_open
    addl    $16, %esp
    movl    $0, %eax
    leave
    ret
    ...
_my_open:
    pushl   $0x12345678
    call    open
    ret
```

図 9 open システムコールの置き換えの例
Fig. 9 Transformation of open systemcall.

攻撃者が，侵入行為を成功させるためには，変換したシステムコールを偽装する必要がある．システムコールの変換方法に，攻撃者が予測できないような乱数的値を加えることによって攻撃が成功する確率は無視できるほど小さい．また，プログラムが発行するシステムコールはすべて，あらかじめ異常検知システムが把握しているため，誤検知は発生しない．

5. 結　　論

本論文では，システムコールに関する N-gram と確率論とに基づき，異常検知手法の提案を行い，その有効性を検証した．提案手法は，Eskin らの手法[6] と同様に，確率論的観点からシステムコールシーケンスに

おける個々のシステムコールの間に相関関係が存在するということに基礎を置いている．具体的には，システムコールシーケンスから生成した N-gram に，ベイジアンネットワークを適用した．その結果，N-gram におけるシステムコール間には，一部非順序的な性質を持つことが明らかになった．

また，異常検知において誤検知が発生する原因について調査を行い，上記の実験結果から得られた事実をもとに，誤検知をなくす方法について言及した．具体的には，プログラムに新たな情報を付加してシステムコールの発行とプログラムの制御フローとを関連付ける作業の非決定性をなくす手法の実現可能性について述べた．

将来の課題としては，本研究によって明らかになった事実や性質に基づき，実用的な異常検知システムを構築することなどがある．

参 考 文 献

1) Beyond-Security's SecuriTeam.com. Writing Buffer Overflow Exploits — a Tutorial for Beginners. http://www.securiteam.com/securityreviews/5OP0B006UQ.html (accessed 2003-09-05).
2) Forrest, S., Hofmeyr, S.A., Somayaji, A. and Longstaff, T.A.: A sense of self for Unix processes, *Proc. 1996 IEEE Symposium on Computer Security and Privacy*, pp.120–128 (1996).
3) Forrest, S., Hofmeyr, S.A. and Somayaji, A.: Intrusion detection using sequences of system calls, *Journal of Computer Security*, Vol.6, pp.151–180 (1998).
4) Warrender, C., Forrest, S. and Pearlmutter, B.: Detecting Intrusions using System Calls: Alternative Data Models, *Proc. IEEE Symposium on Security and Privacy*, No.1, pp.133–145 (1999).
5) Marceau, C.: Characterizing the behavior of a program using multiple-length n-gram, *Proc. 2000 Workshop on New Security Paradigms*, pp.101–110 (2000).
6) Eskin, E., Lee, W. and Stolfo, S.: Modeling System Calls for Intrusion Detection with Dynamic Window Sizes, *Proc.2001 DARPA Information Survivability Conference & Exposition*, pp.165–175 (2001).
7) Tan, K.M.C. and Maxion, R.A.: "Why 6?" Defining the Operational Limits of stide, an Anomaly-Based Intrusion Detector, *Proc.IEEE Symposium on Security & Privacy*, pp.188–201 (2002).
8) Lee, W., Stolfo, S. and Chan, P.: Learning Patterns from Unix Process Execution Traces for Intrusion Detection, *Proc. AAAI97 Workshop on AI Methods in Fraud and Risk Management*, pp.50–56 (1997).
9) Wagner, D. and Dean, D.: Intrusion Detection via Static Analysis, *Proc. 2001 IEEE Symposium on Security and Privacy*, pp.156–169 (2001).
10) Oka, M., Abe, H., Oyama, Y. and Kato, K.: Intrusion Detection System Based on Static Analysis and Dynamic Detection, *Proc. Forum on Information Technology* (*FIT 2003*) (2003).
11) Wagner, D. and Soto, P.: Mimicry Attacks on HostBased Intrusion Detection Systems, *Proc. 9th ACM Conference on Computer and Communications Security* (2002).
12) Motomura, Y. and Hara, I.: User Model Construction System using Probabilistic Networks, http://staff.aist.go.jp/y.motomura/ipa/ (accessed 2003-09-05).
13) Conover, W.J.: *Practical Nonparametric Statistics*, John Wiley & Sons, Inc. (1971). (ISBN 0-471-16068-7)

(平成 16 年 11 月 29 日受付)
(平成 17 年 6 月 9 日採録)

鑓 講平（学生会員）

2004 年九州大学工学部電気情報工学科卒業．同年同大学大学院システム情報科学府に入学．オペレーティングシステム，とりわけ侵入検知システムに関する研究に従事．

田端 利宏（正会員）

1998 年九州大学工学部情報工学科卒業．2000 年同大学大学院システム情報科学研究科修士課程修了．2002 年同大学院システム情報科学府博士後期課程修了．2001 年日本学術振興会特別研究員．2002 年九州大学大学院システム情報科学研究院助手．2005 年から岡山大学大学院自然科学研究科助教授．博士（工学）．オペレーティングシステム，コンピュータセキュリティに興味を持つ．電子情報通信学会，ACM 各会員．

櫻井 幸一（正会員）

1988 年九州大学大学院工学研究科応用物理専攻修士課程修了．同年三菱電機（株）入社．現在，九州大学大学院システム情報科学研究院情報工学部門教授．1997 年 9 月より 1 年間コロンビア大学計算機科学科客員研究員．2004 年 4 月より九州システム情報技術研究所第 2 研究室室長併任．暗号理論・情報セキュリティ・社会情報工学の研究に従事．博士（工学）．2000 年情報処理学会坂井特別記念賞，2000 年・2004 年情報処理学会論文賞，2005 年 IPA 賞授賞．電子情報通信学会，日本数学会，ACM，IEEE 各会員．