資源情報の特徴抽出によるモデル化手法と攻撃検知法の提案

清水 裕基 †　　菅谷 みどり ‡　　秋岡 明香 ††　　吉永 努 ††

† 電気通信大学 電気通信学部 情報通信工学科

‡ 独立行政法人 科学技術振興機構 ディペンダブル組み込みOS研究開発センター

†† 電気通信大学 大学院情報システム学研究科

1　はじめに

従来からの攻撃検知手法として，パターンマッチングやヒューリスティックが提案されている [1]．しかし，これらは0-day攻撃に対して無効である．一方でシステムコール系列を学習し，学習系列から外れた場合異常として検知する手法も提案されている [2]．この手法では0-day攻撃を検知できる可能性もあるが，学習した系列の増加とともにオーバーヘッドも増加する．これはリアルタイムに異常を検知する際に問題となる．しかし攻撃の増加は著しく，またノンストップのシステム要求が増えつつある現在，軽量でかつリアルタイムでの攻撃検知手法の要求が高まっている．

本稿ではコンピュータの資源情報を用いた攻撃検知法を提案する．従来ツールでは情報の精度や収集間隔に問題がある．そこでAyaka[3]を用いることで，資源情報を安定した周期，高い精度で収集を行う [4]．システムは収集した情報をモデル作成フェーズと異常検知フェーズで用いる．モデル作成フェーズでは，平均値と標準偏差をとる手法及びN-Gram法を用いて正常モデルを作成する．異常検知フェーズでは作成した正常モデルと比較することで異常の検知を行う．

2　異常検知手法の提案と実装及び実験

本稿では，Ayakaから得られる資源情報のうち，プロセスのCPU利用時間 [ns] とメモリのページファイル利用数から異常を検知する [4]．実験で用いたマシン構成を表1に示す．

モデル構成時は，プロセスを複数回稼働させることでデータを収集する．資源情報を異常検知に用いるとき，スケジューラやハードウェアの省電力機能などに影響を受けるという問題がある．本稿ではこの影響を受けにくいモデルの構成法と異常判定式を提案する．

†Hiroki SHIMIZU ‡Midori SUGAYA ††Sayaka AKIOKA ††Tsutomu YOSHINAGA

†Department of Information and Communication Engineering, The Univercity of Electro-Communications.

‡Dependable Embedded OS Center, Japan Science and Technology Agency(JST)

††Graduate School of Information Systems, The Univercity of Electro-Communications.

表1: マシン構成

Model	Kohjinsha SH6
CPU	Intel A100 600MHz
MEM	1GB
Kernel+Ayaka	2.6.24+ayaka-v0.1(20090707)patch

2.1　CPU資源情報の統計法と異常検知法

CPU資源情報はある範囲の総和を取るとほぼ一定である．本稿では総和を取る範囲を，メモリ利用量が設定した閾値 (CPU_THRESHOLD) を超えるまでとした．この手法で，スケジューラなどによる影響を受けにくい範囲を設定できる．ばらつきに対して柔軟な検知を行う必要があるが，これは正規分布に従っているため，次のようなモデル化を行う．

1. Ayaka出力値を閾値 (CPU_THRESHOLD) を超えるまで総和を取る．これをCSUMとする．
2. CSUM結果を複数回そろえ平均値を取る．これをAVEとする．
3. AVEでCSUMを割る．
4. 3の標準偏差を算出する．これをSTDとする．

AVEとSTDを用いて検知式をモデル化する..

$$\text{if not}\left\{\left(1-\text{STD}<\frac{\text{CSUM}}{\text{AVE}}\right)\middle|\left(\frac{\text{CSUM}}{\text{AVE}}<1+\text{STD}\right)\right\}$$

then"Anomaly"

本稿手法の利点は，常にO(1)の計算量で異常判定が行えることである．

2.2　メモリ資源情報の統計法と異常検知法

メモリ資源情報はその遷移系列がプロセスに固有である [4]．そこで，次のようなモデル化を行う．

1. Ayakaの出力値の前後差分をとり，プロセスの稼働時間が閾値 (MEM_THRESHOLD) を超えるまでの総和をとる．これをMSUMとする．

2. 1の結果をN-Gramの一要素とし，系列モデル化する．

系列モデルMMODEL[1..N]を用いて次の式で異常を判定する．ここでMDATA[1..N]は異常検知フェーズでのMSUM値N回分である．

if not(MDATA[1..N] == MMODEL[1..N])

then"Anomaly"

2.3 実装および実験

2.1節及び2.2節で提案した手法を実装し，評価実験を行った．検証プログラムには，Linux-kernel2.6.24に存在するvmsplineの脆弱性攻撃コードを用いた[5]．このプログラムを実際に攻撃を行う(プログラム改変なし)のものと，攻撃を行う命令のみをコメントアウトしたものの2種類用意した．まず攻撃を行わないものを10回学習しモデルを生成した．次に攻撃を実行し，異常判定が行えるか観察した．実験のパラメタ値(表2)とCPU利用時間を用いた異常検知結果(表3)を以下に示す．

表2: 各パラメタ値の設定

CPU_THRESHOLD(pages)	10
MEM_THRESHOLD[ns]	1000000
Ayaka_PERIOD[ns]	1000
N	3

表3: CPU利用時間を用いた異常検知結果

NO	AVE[ns]	STD	DATA[ns]	JUDGE
1	66959.2	1.70	11961	
2	452393.0	0.01	564136	Anomaly
3	18830.4	0.00	4137983	Anomaly
4	12349.0	0.21	19522	
5	43500.0	0.08	10400	
6	2321169.4	0.08	44768	
7	7358.6	0.30	2333742	

1回のプロセス稼働で7回の判定があり，2回目及び3回目の判定で異常を検知した．この実験を複数回行い，表3と同様の結果を再現することができたため，提案手法の有効性が示せた．

同様にメモリ利用量に関しても実験を行った．学習で生成された系列データは16個であった．判定回数は7回で，7回すべて異常系列を検知した．

よってCPU利用時間及びメモリ利用量の両変化から異常を検知でき，本稿手法の有効性が確認できた．

3 まとめと課題

本稿で示した手法の問題点として，次のような点があげられる．

適切な閾値やパラメタの設定法

閾値やパラメタの設定は検知精度とオーバーヘッドに影響を与える．現状これは経験から設定されている．またプロセスに対し常に一定の閾値となっているが，これを流動的に変化させる手法も検討する必要がある．

モデル作成時の試行回数

モデル作成時，試行回数を増やすとより多くのパターンが網羅される．しかしどの程度の回数で網羅できるのか，また過学習の問題も考慮しなければならない．

まだ課題点はあるが、単純な手法で異常を判定する手法の有効性を示すことができた．今後研究を続けることでパラメタの推定法などの改良を続けていきたい．

謝辞 本研究の一部は，文部科学省研究費補助金特定領域研究「情報爆発時代に向けた新しいIT基盤技術の研究」（計画研究A02-00-04，情報爆発に対応する高度にスケーラブルなモニタリングアーキテクチャ）による助成を受けた．

参考文献

[1] 菅原 千石ら:未知ウイルス検知技術に関する一考察 The 2004 Symposium on Criptography and Information Security (SCIS 2004)

[2] 春山 中里ら:権限の悪用を考慮したシステムコールモニタリングによる侵入検知手法 ソフトウェア科学会プログラミングおよび応用のシステムに関するワークショップ2005

[3] Midori Sugaya Yuki Ohno and Tatsuo Nakajima : Lightweight Anomaly Detection System with HMM Resouece Model International Journal of Security and Its Applications, IJSIA Vol.3 No.3 2009 (to appear)

[4] 清水 菅谷ら:細粒度リソース監視による攻撃検知法の提案と考察 ソフトウェア科学会第26回大会発表 (CD-ROM)

[5] milw0rm:Linux Kernel 2.6.17 - 2.6.24.1 vmsplice Local Root Exploit http://www.milw0rm.com/exploits/5092