Mechanized reasoningを用いたアクセスログの統合と解析の自動化

安藤類央† 門林雄基† 三輪信介† 篠田陽一††

†情報通信研究機構
〒184-8795 東京都小金井市貫井北町 4-2-1
††北陸先端科学技術大学院大学
〒９２３-１２９２　石川県能美市旭台１丁目１番地
ruo@nict.go.jp

あらまし
本論文では、モニタリング、フィルタリング技術の発達により可能になった多様なログを統合し、セキュリティインシデントに関する情報を抽出するための解析手法を提案する。適用手法としては、述語論理、導出法などを用いる自動推論 (mechanized reasoning) を用いて、各種ログに共通な観測項目によってログの統合を行い、検出するイベントとの間で節矛盾を生成することで関連情報を推論過程として抽出する。適用例として、トラフィック生成元のプロセス ID やライブラリ、ファイルなどの情報を抽出する。データセットは MARS Dataset を用い、２種類の導出手法を用いて、検出過程で生じた節数を比較し、評価実験を行った。

An automated integration and analysis of access log using mechanized reasoning

Ruo Ando† Youki Kadobayashi † Shinsuke Miwa† Yoichi Shinoda ††

†National Institute of Information and Communications Technology,
4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795 Japan
††Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292 Japan

Abstract With the rapid advance of monitoring and filtering technology, malware analysis need to synthesize a variety of access log and extract information to detect security incident. In this paper we an automated integration and analysis of access log using mechanized reasoning. In proposed system, we resolve several kinds of access log into uniform clausal representation. Then, automated deduction system generates unit conflict to detect malware's behavior. We apply our system for coping with MARS dataset to evaluate numerical outputs. Also, we compare the result of two kinds of resolutions: binary and hyper resolution.

1　はじめに

近年のモニタリング、フィルタリング技術の進展により、ネットワークトラフィックだけでなく各種リソースアクセスのログが採取可能になった。しかしながら、これらのログを統合し、イベントに関する情報を抽出するための解析手法はまだ提案されていない。本論文では、単一化、導出、置換などの論理演算を含む自動推論（Mechanized reasoning）を用いて、マルウェア

によるソケット、ファイル、メモリなどの多様なアクセスログを統合し、情報抽出、解析の自動化を行う手法を提案する。

2　関連研究

最近の定理証明系、宣言型言語には、Alloy, Issabelle, XSB などがある。Haskell や Erlang は、その型安全性や堅牢性から、WEB やネットワークプログラミングに、積極的に適用されている。Ocaml は、仮想マシンモニタ XEN のプロジェクト XenAPI に一部適用されている。

3　提案手法

本節では、ログの統合と自動解析に必要な、自動推論の中心的手法である resolution（導出）演算と、問題の定式化について述べる。

3.1　導出法

節 Cls1 と Cls2 がリテラル L1,L2 を持つ場合、導出節 C_R は下記によって得られる。

$C_R = (C_1\sigma \setminus L_1\sigma) \cup (C_2\sigma \setminus \bar{L_2}\sigma)$

ここで、σ は、リテラル L1 と L2 を等しくする単一化演算子である。σ は、最汎単一化子（most general unifier）の場合もある。

$Lit_1 \in Cls_1$ は $Lit_n \in Cls_n$ でも可能であり、複数の節から導出する方法を超導出という。二項導出、超導出の計算コストの実験結果については５節で述べる。

3.2　問題の定式化

ログ L からイベント E を発見することを、節集合 S から論理式 P が恒真であることを導出するとする。これは節集合 S に P の否定 $\neg P$ を付加して空節を導くことと同義になる。

検体 x のイベントの集合 S(x) と T(x) があり、R(a) が起ったことで S(x) が生じ、P(x) の結果になった場合、

$\forall x((S(x) \vee T(x))) -> P(x))$

$\forall x((S(x) \vee R(x)))$

$\neg R(a)$

以上３つの節に $\neg P(a)$

を付加して導出を行うと、空集合が得られることになる。これをプログラムの形式で表現すると、

```
#set 1
S(x). T(x). R(a).
#set 2
-S(x) | T(x) -> P(x).
-S(x) | -R(x).
```

となる。その他本提案手法では、項書き換えや包摂処理を行い、ログからイベントと関連情報を抽出する。

4　適用アルゴリズム

4.1　支持集合戦略

支持集合戦略 は、１９６５年に Wos らによって提案されたものである。この計算戦略は制限戦略の１つで、自動推論プログラムに目標とする解空間に関係ないところを探索せずに、対象としている問題に集中させるようにする。節集合 S、T があり、S-T が充足可能であるとき、T は S の支持集合である。このとき、支持集合に属さない節同士では導出を行わず、支持集合に属する節との間で、導出を行う方針を支持集合戦略という。

4.2　超導出

超導出 は１９６５年に Robinson らによって提唱された手法で、通常の導出系の手法では１対の節から順次導出を行うのに対して、２個以上の節に対して導出を行う。超導出の意味は、何段階もの２項導出にあたる作業を１つにまとめたもので、通常の２項導出に比べて、多くの導出が起こるという事を指す。

4.3　包摂

定理証明を用いた推論プロセスでは、目標とする節を導出する過程で、いくつかの節が保持

```
given clause #16351: (wt=5) 18983 [hyper,16373,15734,8] port_match(pid(756),port(1038)).
** KEPT (pick-wt=1): 18984 [hyper,18983,15739,15763,demod,propositional] ok_2.

----> UNIT CONFLICT at  12.41 sec ----> 18985 [binary,18984.1,15741.1] $F.

Length of proof is 6.  Level of proof is 3.

---------- PROOF ----------

2 [] socket(pid(1232),port(1900),proto(17)).
8 [] socket(pid(756),port(1038),proto(6)).
2819 [] packet(src(sip1(10),sip2(0),sip3(0),sip4(1),sport(mtqp)),dst(dip1(da),dip2(69),dip3(2d),sip4(static),dport(xlhost))).
14992 [] library(pid(1232),dll(_windows_0_system32_wshtcpip_dll)).
15576 [] file(pid(1232),file(_windows_0_system32)).
15729 [] sport(mtqp)=sport(1038).
15734 [] -packet(src(sip1(w1),sip2(w2),sip3(w3),sip4(w4),sport(x2)),dst(dip1(w6),dip2(w7),dip3(w8),sip4(w9),dport(w10)))|
          -socket(pid(x1),port(x2),proto(x3))|por\_match(pid(x1),port(x2)).
15735 [] -socket(pid(x1),port(x2),proto(x3))| -library(pid(x1),dll(y1))|pid_match_1(x1).
15736 [] -library(pid(x1),dll(y1))| -file(pid(x1),file(z1))|pid_match_2(x1).
15737 [] -pid_match_1(x1)| -pid_match_2(x1)|pid_match_3(x3).
15739 [] -port_match(pid(x1),port(x2))| -pid_match_3(x2)|$EQ(x2,22)|ok_2.
15741 [] -ok_2.
15750 [hyper,15576,15736,14992] pid_match_2(1232).
15762 [hyper,2,15735,14992] pid_match_1(1232).
15763 [hyper,15762,15737,15750] pid_match_3(x).
16373 [2819,demod,15729] packet(src(sip1(10),sip2(0),sip3(0),sip4(1),sport(1038)),
        dst(dip1(da),dip2(69),dip3(2d),sip4(static),dport(x))).
18983 [hyper,16373,15734,8] port_match(pid(756),port(1038)).
18984 [hyper,18983,15739,15763,demod,propositional] ok_2.
18985 [binary,18984.1,15741.1] $F.

------------ end of proof -------------


Search stopped by max_proofs option.
```

図 1: 定理証明系の出力結果。pcap ログ、ファイルアクセス、メモリダンプログを統合し、トラフィックの送受信の IP アドレス、送信元のプロセス ID、送信元のプロセスがロードしている DLL などを検出した。

され、新しい節が生成された時点で、過去に保持された節との間で、改めて定理が適用される。この保持されている節のうち、より一般的な節を残す処理を包摂という。

4.4 デモジュレーション

デモジュレーションとは、あらかじめ等価代入を行うための節を定理証明系に加えて、処理節群の簡略化あるいは正準化を行う処理である。本論文ではデモジュレーションを pcap データのポート番号の処理に適用した。

5 解析結果

図１は検体の解析時の生成節数と超導出による生成節数を示した。計算コストのうち主要な操作である導出によるコストが大部分をを占めることが明らかになった。

図２は、検体ごとの項書き換え数を示したものである。節の書き換えの操作は、デモジュレーションやパラモジュレーションがあり、本論文ではデモジュレーションをポート番号の処理に用いた。置換処理コストは、検体ごとの利用ポート状況を大きく反映する結果になった。図３は包摂処理の計算コストを示したものである。前向き包摂数は、導出あるいは全体コストとほぼ一致するが、支持集合による包摂コストは検体によって大きく異なる結果になった。

図４は２項導出と超導出の生成節数の比較を示したものである。概ね超導出の適用により計算コストは削減されるが、３．１０．１１に関しては２手法の効果はあまり変わらない結果になった。また、検体１７に関しては２項導出の計算コストの方が低い結果になった。

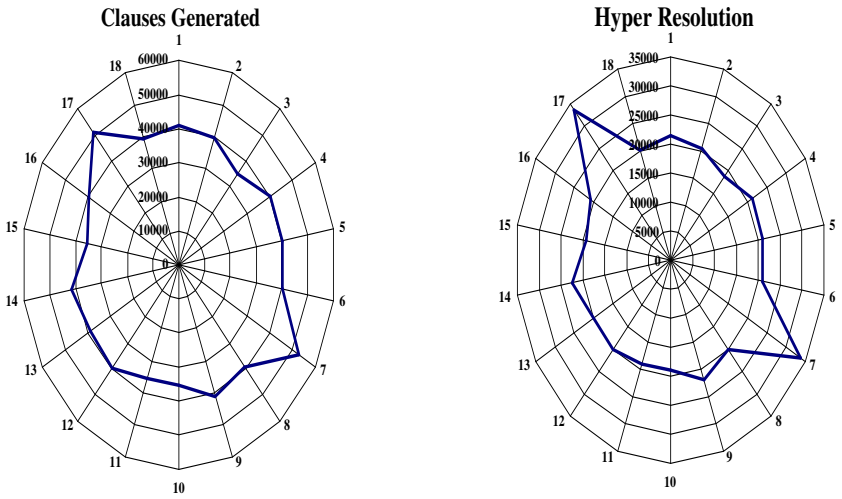

図 2: 各検体の解析時の生成節数と超導出による生成節数。計算コストのうち主要な操作である導出によるコストが大部分ををを占めることが明らかになった。

```
Search stopped by max_proofs option.

============ end of search ============

-------------- statistics test ! --------------
clauses given                16351
clauses generated            35070
  hyper_res generated        18720
  demod_inf generated        16350
demod & eval rewrites        14471
clauses wt,lit,sk delete         0
tautologies deleted              1
clauses forward subsumed     31827
  (subsumed by sos)          12535
unit deletions                   0
factor simplifications           0
clauses kept                 3242
new demodulators                 0
empty clauses                    2
clauses back demodulated         0
clauses back subsumed            2
usable size                  16355
sos size                     2607
demodulators size               17
passive size                     2
hot size                         0
Kbytes malloced              41015

----------- times (seconds) -----------
user CPU time          12.41          (0 hr, 0 min, 12 sec)
system CPU time         0.13          (0 hr, 0 min, 0 sec)
wall-clock time        12             (0 hr, 0 min, 12 sec)
```

図 6: 定理証明系の各種演算の計算コストの出力例。図２から図５に比較を図示した。

6　まとめと今後の課題

現在、マルウェアの実装の洗練による振る舞いの多様化と、モニタリング、フィルタリング技術の発達により、マルウェア解析者は多様なログを統合し、解析する必要になった。本論文では、pcap データだけでなく、ファイル、ソケットアクセス、メモリダンプなどの各種アクセスログを統合し、セキュリティインシデントに関する情報を抽出するための解析手法を提案した。適用手法としては、述語論理、導出法などを用いる自動推論 (Mechanized Reasoning) を用いて、各種ログに共通な観測項目によってログの統合を行い、検出するイベントとの間で節矛盾を生成することで関連情報を推論過程として抽出する手法を提案し、各検体の情報抽出、振る舞い検知に適用した。適用例として、トラフィック生成元のプロセス ID やライブラリ、ファイルなどの情報を抽出し、出力例と演算時の計算コストなどを示した。また、二項導出、超導出２種類の導出手法を用いて検出過程で生じた節数を比較した。

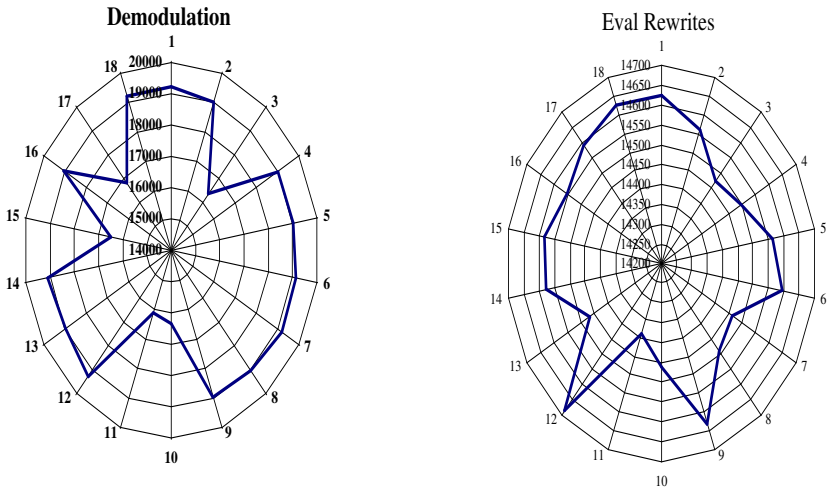

図 3: デモジュレーションは節の書き換えを行う操作で、本論文ではポート番号の処理に用いた。置換処理コストは、検体ごとの利用ポート状況を大きく反映する結果になった。

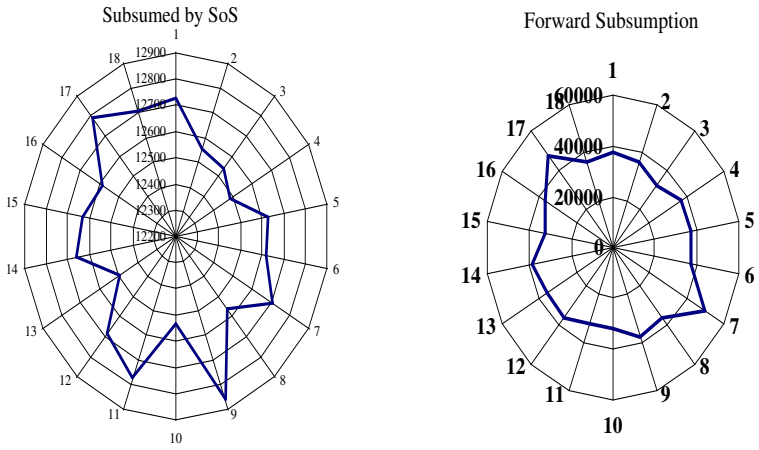

図 4: 包摂処理の計算コスト。前向き包摂数は、導出あるいは全体コストとほぼ一致するが、支持集合による包摂コストは検体によって大きく異なる結果になった。

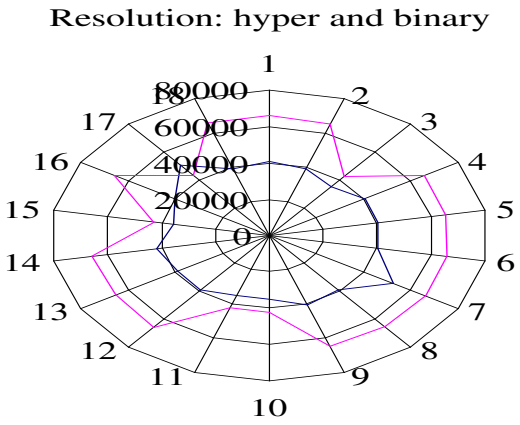

図 5: ２項導出と超導出の生成節数の比較。概ね超導出の適用により計算コストは削減されるが、３．１０．１１に関しては２手法の効果はあまり変わらない結果になった。また、検体１７に関しては２項導出の計算コストの方が低い結果になった。

参考文献

[1] Larry Wos, George A. Robinson, Daniel F. Carsonm ”Efficiency and Completeness of the Set of Support Strategy in Theorem Proving”, Journal of Automated Reasoning, 1965.

[2] Larry Wos: The Problem of Explaining the Disparate Performance of Hyperresolution and Paramodulation. J. Autom. Reasoning 4(2): 215-217 (1988)

[3] Larry Wos: The Problem of Choosing the Type of Subsumption to Use. J. Autom. Reasoning 7(3): 435-438 (1991)

[4] Larry Wos, George A. Robinson, Daniel F. Carson, Leon Shalla, ”The Concept of Demodulation in Theorem Proving”, Journal of Automated Reasoning, 1967.

[5] Diomidis Spinellis, ”Reliable identification of bounded-length viruses is NP-complete”, IEEE Transactions on Information Theory, 2000.

[6] Ruo Ando, Yoshiyasu Takefuji, ”Faster resolution based metamorphic virus detection using ATP control strategy”, WSEAS TRANSACTIONS ON INFORMATION SCIENCE AND APPLICATIONS, Issue 2, Volume 3, February 2006 ISSN 1709-0832,pp260-2266,February 2006

[7] Ruo Ando, ”Automated Log Analysis of Infected Windows OS Using Mechanized Reasoning”, ICONIP 2009,Neural Information Processing, 16thInternational Conference, ICONIP 2009, Bangkok, Thailand, December 1-5,2009

[8] 安藤 類央, 門林 雄基, 篠田 陽一, 「Automated deduction system を用いたマルウェア外部観測ログ解析の自動化」情報処理学会 コンピュータセキュリティシンポジウム ２００９ ２００９年１０月

[9] alloy
http://alloy.mit.edu/community/

[10] isabelle
http://isabelle.in.tum.de/

[11] XSB
http://xsb.sourceforge.net/