シグネチャの H/W 検索を考慮したインライン型 IDS の開発（2）

原田　道明*　　貞包　哲男*　　永嶋　規充*

＊ 三菱電機株式会社 情報技術総合研究所

1. はじめに

インターネットの普及と共にネットワーク不正侵入やウイルス等の脅威が顕在化するなかで、確実・高速な侵入検知・防御技術が必要とされている。

本報告では、先に紹介した CAM（Content-Addressable Memory）による侵入検知の高速化手法[1]における CAM フィルタの算出アルゴリズムを紹介する。また、本手法ではフィルタ展開量の抑制が課題であることを示し、効果的な抑制法を検討する。特に、Source Port／Destination Port の指定の有無や文字列照合の有無によってルール集合を分割し、個別に検知することにより、検知処理の性能を損なうことなくフィルタ展開量を効果的に抑制できることを示す。

2. CAM フィルタの算出アルゴリズム

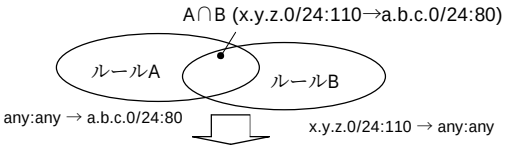
2.1. CAM フィルタ列の性質

図 1 に検知ルールと CAM フィルタの対応関係を示す。CAM は First-Match で使用し、以下の性質を満たすフィルタ列を用意する。

① フィルタはアドレスの包含順に整列される。

② First-Match エントリは、そのアドレスに適用すべき侵入検知ルールの集合と 1 対 1 に対応する。

上記②を満たすには、検知ルールのアドレス条件部に加え、その交差分（図 1 の A∩B）に対して独立したフィルタが必要である。さもなければ、優先度の低いルールが検知されない。



filter#	Proto	Src IP	SP	Dst IP	DP	Rule
1	TCP	x.y.z.0/24	110	a.b.c.0/24	80	A、B
2	TCP	any	any	a.b.c.0/24	80	A
3	TCP	x.y.z.0/24	110	any	any	B

図 1：検知ルールと CAM フィルタの関係

2.2. CAM フィルタ列の算出

2.1 節に示した CAM フィルタ列は、CAM フィルタの包含関係を示した半順序グラフを生成し、半順序ソートすることで得られる。

まず、検知ルールから直接に得られる CAM フィルタの集合をもとに、ノードのみで構成される初期グラフを生成する。次に、各フィルタに対して順にリンクおよび交差フィルタを生成する。

フィルタ A に注目してリンク生成を行うときは、まず、処理中のグラフ上で A に対する極大元・極小元・交差する元をすべて抽出する。次に、抽出した極大元の各々に対して A をその子として登録する（図 2 左側）。極小元に対しても同様である。A と交差する元に対しては交差フィルタを生成して処理待ちキューに追加する（図 2 右側）。

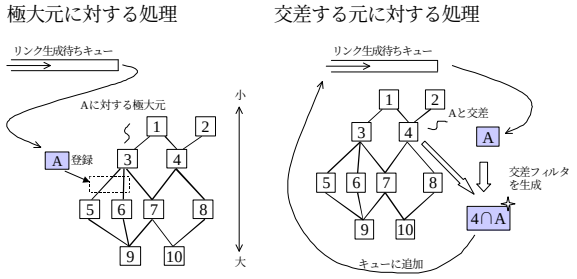


図 2：半順序グラフ生成処理

3. CAM フィルタの展開量

3.1. ルールの増加に対する展開量

CAM フィルタを (Proto, Src IP, Src Port, Dst IP, Dst Port)で表現した場合の展開量を、フリーIDS である Snort[2]**の Version 1.9.1 のルール集合をサンプルとして評価した結果を表 1 に示す。

ルール数の増加につれて Src IP, Dst IP, Proto の種類は頭打ちとなるが、Src Port や Dst Port の種類は増加を続ける。フィルタ展開量は概ね Src Port 数×Dst Port 数に比例して急速に増加する。CAM には既に数百 K エントリの容量をもつ品種が存在するが、各フィルタに所属するルール集合を関連付けるために CAM フィルタ数×所属ルール数分のメモリが必要である。本開発では CAM フィルタ数にして数十 K エントリ程度を実用上の目安とした。

表 1：ルール数の増加とフィルタ展開量

ルール数	アドレスの種類					フィルタ数	
	Proto	SrcIP	DstIP	SP	DP	初期	展開後
100	4	3	2	3	33	41	93
500	4	3	3	12	57	78	537
1,000	4	6	10	50	105	190	9,695
1,485	4	6	10	50	110	195	10,135

Development of In-Line Intrusion Detection Method for Hardware Accelerated Signature Matching, Part 2
Michiaki HARADA*, Tetsuo SADAKANE*, Norimitsu NAGASHIMA*
*Information Technology R&D Center, MITSUBISHI Electric Corporation

** Snort(TM)は米国 SourceFire INC.社の登録商標。

3.2. ルールの種類と展開量

各ルールの内容を吟味した結果、下記のようなルールが特に展開量を増加させることがわかった。

①Source Port を指定したルール

主要なルールはサーバへの攻撃パケットの特徴に着目して Destination Port を指定する。一方、有害サーバの挙動や犠牲ホストが返す応答パケットに着目したルールでは Source Port が指定される。

Source Port 数、Destination Port 数がともに増えていくことにより、展開量が爆発的に増加する。

②Backdoor 等の検知ルール

多くの攻撃は既存サービスを対象とし、Well-Known ポートに対して検知されるが、侵入者が犠牲ホストに配置した Backdoor、Zombie は特異なポート番号を使用する点に着目して検知される。一方、このようなルールに対しては CAM フィルタ毎のルール数は 2～3 個である。約 1500 ルールに対する計測結果、195 個の初期フィルタのうち、所属ルール数が 2 個以下のフィルタが 138 個を占めている。

4. 展開量の抑制法の検討

前節の分析結果を元に、展開量の抑制法を検討し、効果を見積もった。

4.1. ポート指定の有無によるルール集合の分割

Source Port／Destination Port の指定の有無に応じてルール集合を下記のように 4 分割し、個別に CAM フィルタ列を構成して検知する。

①Src Port/Dst Port 双方指定のルール群
②Dst Port 指定のルール群
③Src Port 指定のルール群
④Port 指定のないルール群

CAM フィルタの形式は①～④で個別としても、すべて同じ形式「(Proto、Src IP、Src Port、Dst IP、Dst Port)→Rules」としても性能に大差はない。

3.1 と同じルール集合に対する適用結果を表 2 に示す。初期フィルタ数は分割・非分割時で変わらないが、展開後の合計では 10,135 個→620 個と大幅に削減できた。分割時は Source Port／Destination Port の交差が発生しなくなり、3.2-①の問題が解消されることがわかる。

この方法では各パケットに対して最悪で 4 回の CAM 検索が発生するが、出力されるルール数の合計は分割前と同じなので、性能低下は無視可能な程度である。また、4 分割されたルール群の処理は独立しているので、並列に処理すれば CAM 検索回数による性能低下も回避できる。

表 2: ルール集合を分割した時の展開量

Src Port	Dst Port	ルール数	フィルタ数	
			初期	展開後
any	any	194	19	48
指定	any	152	39	42
any	指定	1,068	123	514
指定	指定	71	14	16
分割時合計		1,485	195	620
[参考] 非分割時		1,485	195	10,135

4.2. 文字列照合が不要なルールの除外

文字列照合を含まないルールは別の CAM 表を用いて高速に検知できるから、本手法による検知対象から除外できる。

実験では、表 2 のルール集合に対して初期 CAM フィルタ数が 195 個→159 個に削減された。削減量はさして大きくないが、性能低下は CAM 検索回数の増加のみにとどまり、容易に適用できる。

4.3. ルール数の少ないフィルタの省略

3.2-②の対策として、初期フィルタから所属ルール数が少ないものを除去する。所属ルールは、除去対象のフィルタを包含する極小のフィルタ群に移す。これらのルールは CAM 照合後に各ルールのアドレス条件を再照合することで正しく扱える。

この方法では CAM フィルタでの絞込みが粗くなって検知速度が低下するので、4.1、4.2 を適用しても容量が枯渇した場合に限り、移動先フィルタへの悪影響が小さいものから処置することが望ましい。また、TCP の場合、セッション学習時に一度だけアドレス照合とルールの絞込みを行い、その結果をセッション情報と共に記憶することにより、後続パケットの処理を高速化することができる。

5. おわりに

本報告では、IDS においてルールの絞込み手段として CAM を使用した場合のフィルタ列の算出方法、展開量の抑制法を示した。一方、監視対象サイトにあわせて検知ポリシーをチューニングすることによっても、容量効率の向上が可能である。今後は、センサ側、管理装置側の双方から検討を継続して進めたい。

参考文献

[1] 「シグネチャの H/W 検索を考慮したインライン型 IDS の開発(1)」 貞包 哲男他、情報処理学会第 66 回全国大会、3J-6
[2] Snort: The Open Source Network Intrusion Detection System, http://www.snort.org