7J-01

組込み機器における Linux 完全性検証機能 IMA の性能評価

村松 拓実† 金城 聖† 毛利 公一†
†立命館大学情報理工学部

1 はじめに

近年，IoT 機器を筆頭に組込み機器を利用する場面が増加している．インターネットを介して通信をする機器が増加していることから，組込み機器がサイバー攻撃にあう可能性がある．そこで，攻撃者の侵入等によってファイルが改ざんされる脅威への対策法としてファイル内のプログラムやデータが正しい状態で保存されていること，すなわち完全性を検証する仕組みが必要である．汎用的な PC の多くは完全性検証機能を備えているが，組込み機器ではリソースの制約のため搭載されることが少ない．しかし，組込み機器全体の完全性を確保するためにはシステム起動時から OS 起動後まで完全性の検証を連続して行う必要があり，OS 起動後は OS 上のファイルを対象とした完全性の検証を常時行うため，OS 上のファイルを対象とした完全性の検証が組込み機器の動作に大きく影響を及ぼす．よって，組込み機器に完全性検証機能を搭載するため，OS 上のファイルを対象とする完全性検証機能を組込み機器で動作させた際の課題を明らかにする必要がある．

本論文では，OS 起動前の完全性は確保されているという前提で，OS 上のファイルを対象とした完全性検証の動作を調査する．調査では Linux カーネルが提供している完全性検証機能である IMA(Integrity Measurement Architecture) を実際に動作させ，組込み機器上で完全性を確保する際の課題について考察する．

2 Linux の完全性検証機能 IMA

Linux カーネルは IMA と呼ばれる完全性検証機能をサポートしている．IMA はカーネル空間で動作し，メモリ上に開かれるファイルやプログラムの完全性を検証する [1].

2.1 概要

ファイルの完全性検証は，ファイルの内容が不正に改ざんされていないかを確かめることである．

IMA の完全性検証機能はユーザまたはシステムがファイルを開こうとした瞬間から動作する．図 1 に IMA によるファイルへのアクセス制御の動作を示す．ユーザがファイルを開こうとすると，開かれる前に IMA はファイルのハッシュ値と正しいハッシュ値の比較を行い，値が異なっていればユーザのファイルへのアクセスを拒否する．

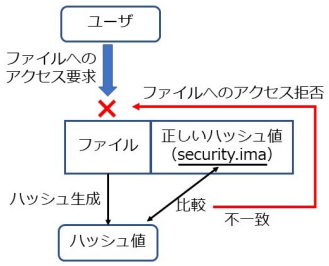

図 1 IMA によるファイルアクセス制御

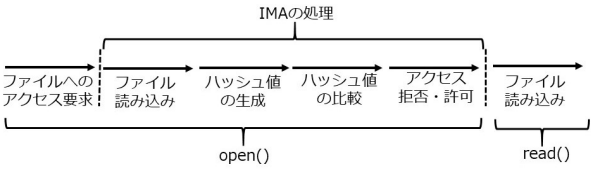

図 2 IMA の完全性検証の動作

2.2 ファイルアクセス動作への影響

IMA が有効になっている場合，ファイルへのアクセス要求があると，IMA による完全性の検証が追加で行われる．そのため，アクセス要求をしてからファイルが読込まれるまでの時間は IMA による処理時間分だけ増加する．また，リソースに制限がある組込み機器では，IMA による処理時間の増加も大きくなり，プログラムの応答時間に大きく影響がでる可能性がある．そこで，ファイルへのアクセス要求をしてからファイルを開き操作できるようになるまでの時間を測定することで，組込み機器における IMA の有効による応答時間への影響とその原因を探る．

3 IMA による完全性検証の動作時間の測定

3.1 IMA の完全性検証の動作

IMA による完全性検証時の動作は図 2 のような内訳になる．図 2 のファイル読み込みがストレージ転送速度，その他の IMA の動作は CPU 性能に影響を受けると考えられる．

Performance Evaluation of Linux Integrity Measurement Architecture on Embedded Device
Takumi Muramatsu†, Akira Kanasiro†, and Koichi Mouri†
†College of Information Science and Engineering, Ritsumeikan Univ.

表 1　測定環境

CPU	Cortex-A72(ARMv8) 4 コア
メモリ	4GB
OS	Raspberry Pi OS（2022-09-22 リリース）
ストレージ	SSD(USB3.0)，microSD(USB3.0)

3.2　測定項目

以下の 3 項目について，IMA の処理時間を測定して比較する．

1. IMA の有効と無効
2. CPU クロック数の変更
3. ストレージを SSD と microSD カードに変更

1 の測定で IMA の有効による処理時間の増加度を確認する．2 と 3 の測定では，それぞれ CPU 性能とストレージ転送速度による処理時間の変化を確認し，処理時間増加のボトルネックを調査する．

3.3　測定方法

ハッシュ値の計算や読み込み速度はファイルサイズに影響されるため，1KB，10KB，100KB，1MB，10MB，100MB の 6 つのファイルを用意し，それぞれを検証対象のファイルとして測定した．測定には図 2 のように open 関数と read 関数を用い，各関数の開始から終了までの時間を測定する．open 関数はファイルへのアクセス要求と IMA の処理時間の測定，read 関数はファイル読み込み時間の測定を担当する．測定は 10 回行い，その平均値を最終的な測定結果として使用する．

3.4　測定環境

調査に使用するデバイスは “Raspberry Pi 4 Model B” であり，スペックは表 1 のとおり．SSD の最大転送速度は USB3.0 接続のため理論値で 625MB/s である．microSD カードは 45MB/s である．

3.5　IMA の有効・無効

IMA の有効と無効それぞれの場合で測定する．CPU のクロック数は 600MHz で固定している．測定結果は表 2 のようになり，測定時間は open 関数と read 関数それぞれの測定時間の合計である．また，IMA の有効による時間の増加割合を知るため，IMA が無効の場合に対する有効の場合での測定時間の比率も記す．

測定結果より，ファイルサイズが小さい場合では IMA の有効と無効で測定時間に大きな変化は無い．しかし，ファイルサイズが大きくなるほど IMA を有効にした場合の測定時間の増加率は大きくなり，100MB のファイルでは IMA の有効により測定時間が 2.05 倍に増加した．

3.6　CPU クロック数とストレージ転送速度

CPU のクロック数を 200，400，600，800MHz の 4 種類に，測定対象のファイルが保存されているストレージを SSD と microSD カードの 2 種類に変更し測定した．ファイルサイズが 100MB のファイルを対象とした測定の結果を表 3 に示す．

この結果より，クロック数の値が低いほど open 関数と read 関数それぞれで測定時間が増加した．そして，同じクロック数であれば，ストレージが SSD と microSD カードの場合の時間に大きな差はない．

表 2　IMA の有効・無効による測定結果

サイズ	測定時間 (ms)		無効に対する有効の比率
	IMA の無効	IMA の有効	
1KB	2.546	3.136	1.23
10KB	2.343	2.681	1.14
100KB	4.955	8.771	1.77
1MB	33.645	64.423	1.91
10MB	307.376	619.407	2.02
100MB	3,073.813	6,303.811	2.05

表 3　100MB のファイルを対象とした CPU クロック数とストレージ転送速度の測定結果 (ms)

クロック数	SSD		microSD	
	open	read	open	read
200MHz	9,842	8,826	11,321	6,975
400MHz	5,147	4,098	5,325	4,132
600MHz	3,507	2,796	3,429	2,798
800MHz	2,771	2,113	2,831	2,105

3.7　考察

3.6 節の測定結果より，クロック数の低下により read 関数の測定時間が増加したので，ファイル読み込みに CPU 性能が関わっていることが分かる．そして，SSD と microSD カードの理論上の転送速度は 10 倍以上の差があるのにもかかわらず，SSD と microSD カードの測定時間の差は小さい．そのため，IMA による完全性検証でのオーバーヘッドはストレージ転送速度による影響は小さく，CPU 性能による影響が大きいと考えられる．

4　おわりに

本論文では，様々な条件で IMA を動作させることで完全性を確保する際の課題について調査した．今後は性能が低い環境で完全性検証機能を有効にした場合でもソフトウェアの動作に支障が出ない手法について検討する．

参考文献

[1] 宗藤 誠治，須崎有康：高信頼を実現する Linux の新しい機能，情報処理 Vol.51 No.10，pp.1284–1293 (2010 年 10 月).