

■インテル トラステッド・エグゼキューション・テクノロジー (インテルTXT) システム全体のセキュリティ強化を支援する仮想化応用技術

大原久樹
インテル

インテルTXT誕生の背景

インテルTXTを理解するには、まず、インテルが提供するハードウェアによる仮想化支援技術と、業界標準化団体のトラステッド・コンピューティング・グループ (TCG)において定義される関連技術を知っておく必要がある。

2005年に発表された、インテル バーチャライゼーション・テクノロジー (VT-x)によってハードウェアによる仮想化支援がサポートされてから、PCクライアントとサーバの仮想化環境は大きく変わった。例えば、Linuxでは、Xenによる完全仮想化によりWindowsをゲストOSとして実行できるようになり、KVMやlguestといったインテルVTを前提とする新しい仮想マシン・モニタ (VMM)がLinuxカーネルに取り込まれるようになった。また商用VMMでは、従来の仮想化手法に加えてVT-xを用いることで、サポート対象となるゲストOSの多様性を深める一助となった。

セキュリティの観点から見ると、VT-xを用いてゲストOS間のアドレス空間を分離することで、よりセキュアなゲストOSを実現できるようになったと言える。アドレス空間は、通常、仮想アドレスから物理アドレスへの変換に用いられるページ・テーブルによって管理されるが、仮想化環境の場合は、VMM内で管理されたページ・テーブルによってゲストOSのアドレス空間が分離される。このように、VMMはゲストOSのアドレス空間の“防波堤”として、セキュリティ上、重要な役割も担っている。

VMMにおける主要なセキュリティの課題としては、①DMA (Direct Memory Access)を用いたドライバの脆弱性、②VMMそのものの脆弱性の可能性、の2つが挙げられる。

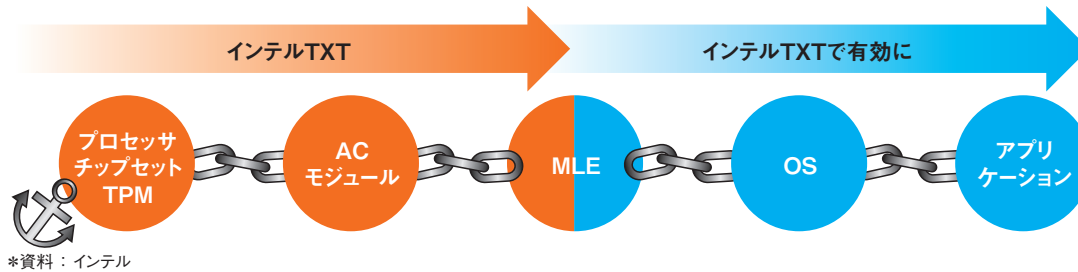
ネットワークやハードディスクのように性能が求められる物理デバイスは、通常、CPUを介さず物理メモリに直接アクセスするDMAの仕組みを用いている。

DMAでは仮想アドレスではなく物理アドレスを使うため、DMAを用いたドライバやVMMに脆弱性があると物理アドレスに自由にアクセスできてしまう。だが、2007年8月にダイレクトI/O対応インテル バーチャライゼーション・テクノロジー (VT-d)が発表されたことで、ゲストOSにとっての物理アドレスからシステムの真の物理アドレスへの変換はチップセット内で自動的に行うことが可能になった。このように、VT-xとVT-dを用いることで、ゲストOSはI/Oを含めてメモリ空間の分離を実現し、ゲストOS内に仮に脆弱性があったとしても、他のゲストOSが影響を受けることはなくなった。

しかしながら、肝心のVMMに脆弱性があった場合、VMM上のすべてのゲスト・ドメインが影響を受けるおそれがある。通常のウイルスやマルウェアでも同様だが、脆弱性への対策として重要なことは、改竄の検出である。ソフトウェアの改竄を検出する方法としては、ハッシュ関数を用いて、利用中のソフトウェアと正常なソフトウェアのそれぞれのハッシュ値を比較することが一般的に知られている。ハッシュ関数は電子署名などにも用いられていて、ハッシュ関数がセキュリティ的に強固であればあるほど異なるデータに対して同じハッシュ値を出力することがまれになる。VMMの場合、改竄を検出するためにVMMのハッシュ値を取得しようとしても、VMMの支配下にあるゲストOSからの検出は当然不可能であるし、ソフトウェアによるVMMの検出はそのソフトウェア自身が改竄される危険性が高い。したがって、VMMが立ち上がる前のハードウェアによるサポートがVMMの改竄検出には不可欠であり、この基盤技術を提供するのがインテルTXTである。

以下、インテルTXT対応のハードウェアが、何をどのように検査(メジャーメント)することで正しい検査を保証(トラストチェーン)するのかについて述べる。

図1: インテルTXTのトラストチェーン



TCGが定義する 2つの重要な概念

メジャーメント

メジャーメントとは、ソフトウェア・コンポーネント(BIOS、ブートローダ、カーネルなど)のハッシュ値を計算し、セキュアなハードウェアに格納することである。ここで言うセキュアなハードウェアとはセキュリティ・チップのTPM(Trusted Platform Module)を指す。TPMの仕様はTCGによって策定されており、最新版はTPM1.2である。TPMにはPCR(Platform Configuration Register)と呼ばれるレジスタがあり、TCGのPC Clientの仕様では24個のPCRが定義されている。PCRは仕様によってそれぞれ各ソフトウェア・コンポーネントに結び付けられている。

メジャーメントでは、ソフトウェア・コンポーネントのハッシュ値はTPMに渡され、該当するPCR値の既存のデータとビット演算が行われたあとで再度ハッシュ値が計算される。TPMでの計算はすべてハードウェア内で行われるため、任意の値にPCR値を設定したり、ユーザーがPCR値を改変したりすることは不可能である。

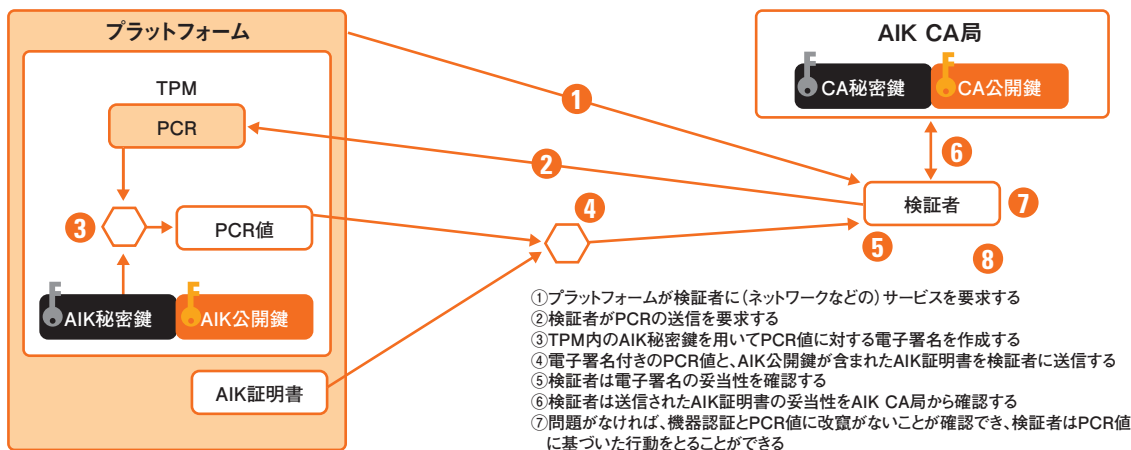
トラストチェーン

システムを起動しVMMが立ち上がるまでの順番は、一般的に「BIOSによるシステム情報の取得→ブートローダの起動→VMMの起動」となる。こうしたブート・プロセスの中で用いられるソフトウェア・コンポーネントの中で1つでも改竄の可能

性があると、VMMそのものの改竄を否定できなくなってしまう。逆に、ブート・プロセス中に実行されるすべてのソフトウェア・コンポーネントのハッシュ値を取得(メジャーメント)できれば、そのシステムを信用できるかどうかを判断できる。これが、TCGで定義されたトラストチェーンと呼ばれる仕組みが生まれた動機である。図1のように、メジャーメント済みのソフトウェア・コンポーネントが、次に起動されるソフトウェア・コンポーネントをメジャーメントする。これを連鎖的に実行することで、カーネルやドライバなどが立ち上がるまでに、すべてのコンポーネントがメジャーメントされたことが保証される。

トラストチェーンにおいて重要なのは、最初に行われるコンポーネント自身のメジャーメントの取り扱いである。最初のコンポーネントをメジャーメントすることはできないからである。TCGではこの最初のコンポーネントをRTM(Root of Trust for Measurement)と定義している。そしてこのRTMをどこに置くかでトラストチェーンは2種類に分類できる。1つはSRTM(Static RTM)で、システムの起動直後に実行されるBIOS内の書き換え不能な領域からトラストチェーンが構築される。もう1つはDRTM(Dynamic RTM)と呼ばれ、CPUの命令など特定のイベントを契機としてトラストチェーンが構築される。SRTMではシステムの起動時のみにしかトラストチェーンを構築できないのに対し、DRTMではシステムを再起動しなくてもトラストチェーンを理論的には再構築できる。インテルTXTはDRTMを実現している。

図2：リモート検証



*資料:インテル

インテルTXTが実現する 堅牢なセキュリティ環境

インテルTXTの構成要素

インテルTXTは、CPU、チップセット、TPM、そしてAC(Authenticated Code)モジュールから構成されている。ACモジュールとは、プラットフォームの構成オプションを検証するためのソフトウェア・モジュールであり、CPU内のACEA(Authenticate Code Execution Area)と呼ばれる領域内でのみ実行される。ACモジュール自身は電子署名されており、その公開鍵のハッシュ値はあらかじめチップセット内に記録されている。

インテルTXTのトラストチェーン

インテルTXTでは、前述したDRTMによるトラストチェーンが構築される(図1を参照)。インテルTXT向けのCPUの新しい命令セットであるSMX(Safer Mode Execution)のGETSEC[SENTER]という命令がRTMを構成する。この命令はメジャーメント環境の構築プロセスを起動する。次のソフトウェア・コンポーネントであるACモジュールのメジャーメント結果は、TCGの仕様に基づいてPCR17番に格納される。次に、ACモジュールが起動し、プラットフォームの構成オプションの検証が行われる。検証結果に問題がなければ、次のソフトウェア・コンポーネントであるMLE(Measured Launched Environment)のメジャーメント結果が

PCR18番に格納される。MLEとは、具体的にはメジャーメントされるVMMのことである。

ここまでが、インテルTXTによって構築されるトラストチェーンである。必要があればOSブート時に起動されるサービス・プロセスや、ドライバ、ユーザー・アプリケーションなどをメジャーメントし、トラストチェーンを延長すればよい。なお、インテルTXTそのものがメジャーメントの対象とするのは、ACモジュールとVMMの2つである。

メジャーメントを用いたプラットフォームの検証方法

メジャーメントされた結果のPCRの値を用いて、トラストチェーン内で改竄が行われたかどうかを検出するための方法としては、ローカルで実施する方法と、別のシステムからリモートで実施する方法がある。

■ローカル検証

正しいと期待されるハッシュ値をあらかじめTPM内のNVRAMに格納し、トラストチェーン内でメジャーメントされたPCR値との比較を行う。比較結果が同一であれば次のコンポーネントの実行を許可し、異なっていればポリシーに応じてシステムダウンなどのアクションを実行する。

■リモート検証

他のプラットフォームからネットワーク越しに検証を行う際には、TPM内に保存されている情報が改竄されないことと、検証先の機器そのものの認証を行う必要がある(図2)。



セキュリティ面から見た利点

インテルTXTのリモート認証の場合、AIK秘密鍵がTPM内に保存されている点や、ハッシュ値や暗号化の計算が物理メモリ上ではなくTPMで実行される点から、セキュリティ強度は高いと言えよう。また、MACアドレスやUUIDのようにソフトウェアによって偽造が可能な手法と異なり、TPMを用いることで、より堅牢な機器認証を実現できる。さらに、PCR17番以降を確認することで、インテルTXTが導入された機器であることを認識することも可能である。そのほか、インテルTXTでVMMを検証し、検証済みのVMMが管理するVT-xおよびVT-dを用いて、ゲストOSのアドレス空間が分離された機器だけネットワークへの接続を許すなど、ネットワークを含めたさまざまな応用例が考えられる。

* * *

従来、セキュリティの運用として行われてきたウイルス／マルウェア対策、ソフトウェア・ベースの機器認証とは異なり、インテルTXTでは、TPMという汎用デバイスを最大限に活用しながら、機器やOS、アプリケーションを認証するための基盤を提供する。また、インテルTXTはインテルVTの拡張として導入される。VT-xとVT-dによってハードウェア・レベルでゲストOSの分離をしつつ、VMMの改竄を検出可能な技術を採用することが、次世代セキュリティの基盤になるからである。

本稿をきっかけに多くの方がインテルTXTに興味を持ち、インテルTXT上のソリューションの発展につながれば幸いである。

〔IDG：月刊Computerworld 2008年1月号に掲載〕