仮想マシンモニタを用いた異常注入システム

星 洋平† 大山恵弘†

1. はじめに

近年，さまざまなセキュリティシステムが研究されている．セキュリティシステムの研究においては，監視対象に実際に攻撃を仕掛け，攻撃が検出されるのを確認することが必要である．しかし，攻撃の余地を残している脆弱なアプリケーションを入手し，かつ，それに対する攻撃コードを入手または作成する作業には大きな手間がかかる．そのため，アプリケーションやOSに異常が発生した時におけるセキュリティシステムの挙動を調べるために，アプリケーションやOSに故意に異常を発生させるシステム (異常注入システム) を用いる方法が提案されている．

既存の異常注入システム[2]では，ランダムな条件で，メモリ上のデータのビット反転などの，単純な異常を注入している．よって，現実の攻撃に近い挙動の異常を注入することが難しい．

本研究では，仮想マシンモニタ Xen を用い，ユーザによるシナリオに沿って，アプリケーションやOSに異常注入を行うシステム HyperAttacker を提案する．

2. 提案システム

本研究が対象としているのは，仮想マシンモニタ Xen の DomU 上で動作中のアプリケーション，OSである．仮想マシンモニタは，メモリやレジスタなどコンピュータの資源を仮想化し，管理している．そのため，さまざまな箇所に異常を注入することが可能である．例えば，コントロールレジスタなど，同一システム上の機構では難しい異常注入が可能である．また，メモリのカーネル領域も操作可能なため，kernel-level rootkit などによる，カーネルレベルの攻撃や異常を注入することも可能である．

本システムにおいて，異常はユーザに与えられたシナリオに沿って注入される．シナリオには，異常が注入される条件 (open システムコールが 100 回発行されるなど) と，その条件が成立したときに実行される操作を記述する．

異常が注入される流れについて述べる．対象とするアプリケーションや OS を，常に動作が Xen ハイパーバイザによって監視されている状態にする．DomU 上のアプリケーションおよび OS の動作は，Dom0 上の制御プログラムに逐次送信される．監視対象がシナリオに沿った動作をしたとき，制御プログラムが DomU の資源を操作することで，異常が注入される．

3. 実装

アプリケーションや OS の監視は，システムコールのフックによって実現している．システムコールのフックには，ゲスト OS が特権命令を行うと制御が Xen ハイパーバイザに移ることを利用している．本システムでは，ゲスト OS のシステムコール開始時のコードを特権命令 (HLT 命令) に書き換えている．

DomU の資源操作は，メモリを共有することによって実現している．Dom0 と DomU でメモリを共有し，共有したメモリの読み書きを行うことで，DomU のメモリ操作が可能である．同様に，Dom0，DomU とハイパーバイザでメモリを共有し，それを通じて DomU の仮想 CPU のレジスタを読み書きしている．

4. 関連研究

文献[3]では，HyperAttacker のプロトタイプについて述べられている．プロトタイプでもシナリオに沿ってアプリケーションに異常を注入することができる．しかし，プロトタイプは，ユーザレベルのプロセストレース機構を用いて実装されているため，カーネルレベルの攻撃や異常注入を行うことはできない．

文献[1]では，仮想マシン内のプロセスを外部から制御するシステムを提案している．本システムで，システムコールフックによるアプリケーションの監視を行う際に，尾上らのシステムで用いられている手法を踏襲した．尾上らのシステムは，本システムと異なり，システムコールの実行制御を行うだけであり，監視対象のアプリケーションや OS に異常を注入することはできない．

† 電気通信大学大学院電気通信学研究科情報工学専攻

5. 現状と今後の予定

仮想マシンモニタ Xen を用いてアプリケーションや OS に異常注入を行うシステム HyperAttacker を提案した．現在は提案システムの実装中であり，DomU のメモリ，レジスタ操作およびアプリケーションや OS の，システムコールのフックによる監視が可能である．

今後は，実行コードにフックを入れる方法を考案，実装する．その後，注入する異常の考案，実装を行い，本システムを用いた場合のオーバーヘッドの測定を行おうと考えている．

参 考 文 献

1) K. Onoue, Y. Oyama and A. Yonezawa: Control of System Calls from Outside of Virtual Machines, *Proceedings of the 2008 ACM Symposium on Applied Computing*, Fortaleza, Ceara, Brazil, pp. 2116–2121 (2008).
2) M. Le, A. Gallagher and Y. Tamir: Challenges and Opportunities with Fault Injection in Virtualized Systems, *First International Workshop on Virtualization Performance: Analysis, Characterization, and Tools*, Austin, Texas (2008).
3) 大山 恵弘: セキュリティ機構の開発と評価のための異常注入システム，日本ソフトウェア科学会第 25 回大会論文集 (2008).