Computer Security Symposium 2017
23 - 25 October 2017

効率的なペネトレーションテストのための テストプラットフォームの提案

木藤 圭亮† 西川 弘樹† 山本 匠† 河内 清人†

概要：製品に搭載されるソフトウェアに対して脆弱性がないか検査する、ペネトレーションテストの重要性が近年急増している。しかしながら、複数システムが協調動作するシステムに対してペネトレーションテストを行うためには、サブシステムの状態を切り替えながらペネトレーションテストを手動で行う必要があり、非効率的であった。本稿では複数システムが連携して、状態が切り替わるシステムに対して、効率的に状態復元が行えるペネトレーションテスト手法を提案する。

キーワード：ペネトレーションテスト、VM、システム状態復元

Proposal on Test Platform for Efficient Penetration Testing

KEISUKE KITO† HIROKI NISHIKAWA†
TAKUMI YAMAMOTO† KIYOTO KAWAUCHI†

Abstract: Importance of penetration testing for software on products is increasing. However, in order to perform a penetration test on a system in which a plurality of systems cooperatively operate, it is necessary to manually perform a penetration test while switching the state of the subsystem which is inefficient. In this study, we propose efficient penetration test platform for systems that have intercommunication between subsystems. Efficient state recovery is enabled by our scheme.

Keywords: Penetration test, Virtual Machine, System State Recovery

1. はじめに

出荷前の製品やリリース前のシステムに対して、脆弱性が残存するかを試験するペネトレーションテスト(以下、ペンテスト)が近年注目を浴びている。ペンテストでは、実際の製品やシステムに対して、テスターが既知の脆弱性や、脆弱な設定等を悪用して、システムをどこまで攻撃できるかをテストする。

しかしながら、近年の IoT や System of Systems に代表されるようにテスト対象となるシステムはより複雑化しており、複数のサブシステムが緊密に連携して構築されるシステムにおいては、あるサブシステムの状態が別のサブシステムのプログラムの挙動に影響を与えている場合が多い。そのためペンテストを行う際には、各々のサブシステムの状態を切り替えながら、対象システムに対してペンテストを実施することで、ペンテストの網羅性を向上させる必要があるが、効率的に状態を切り替えながらペンテストを行うプラットフォームはこれまで提案されてこなかった。

本稿では、仮想計算機（以下、VM）上に再現したサブシステム間で連携のあるテスト対象を、スナップショット数を削減しつつ、効率的にシステムの任意の状態の組み合わせを効率的に状態復元可能なペンテスト向けプラットフォームを提案する。提案手法を用いることで、システム間連携のある複雑なテスト対象においても、効率的にペンテストを実施可能となる。

2. 背景と課題

2.1 ペネトレーションテストとシステムの状態

ペネトレーションテストは侵入テストとも呼ばれ、コンピュータシステムに対して、既知の脆弱性や設定の不備を悪用して、実際に侵入や攻撃が可能かどうかをテストすることである。一般的な脆弱性診断とは異なり、残存する脆弱性を特定するだけではなく、脆弱性を悪用してどこまで攻撃できるかなどを、実際にテスターが攻撃プロセスを経てテストを行う。

ペンテストでは、実際のシステムに対してあらゆる攻撃手法を用いて侵入が可能かどうかを検査する必要がある。脆弱性診断等で用いられる脆弱性スキャナは、システムに対して既知の脆弱性があるかどうかを判断するプログラムであり、テスト対象のシステムの状態は考慮しない。そのため、単にツールによる検査を行うだけでは、脆弱性をうまく発見できない場合が考えられる。そのためテスト対象のシステムを実際の構成で構築し、テスト対象の持ちうる状態をできる限り多く再現した上で、テストを行う必要がある。しかしながら、IoT や System of Systems に代表されるように、システム間が連携して大きなシステムを構成し

†1 三菱電機株式会社 情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation

て、システムが複雑化しているため、システムの持ちうる全ての状態を、手動ですべて再現してペンテストを行うのは非効率的であり、効率的に実施するためのプラットフォームが提案されていなかった。

たとえば、n 個のサブシステムから構成されるシステムにおいて、各々が k 状態を持ちうるとすると、システム全体としては、最大で n×k 状態存在することとなり、サブシステム数が膨大になった場合は、システム全体として持ちうる状態数も膨大になってしまうため、ペンテストではこれらの状態を手動で切り替えるのは現実的ではない。

2.2 VM を用いたシステムの状態復元

2.2.1 VM を用いた単純な手法

テスト対象システムの状態をペンテスト時に、状態復元を行うために、テスト対象を VM 上で再現する方法がある。単純な手法では、システムの持ちうる状態をすべて網羅するように動作させ、各々の状態において VM のスナップショットを取得し、ペンテスト時にスナップショットから復元することで状態を再現することが可能である。しかし、n 個のサブシステムと、各々のサブシステムで k 個の状態を持つ例の場合は、各サブシステムを VM 上で再現するため、n 台の VM を同じタイミングでスナップショットを取得し、かつ n×k 回だけ同様にスナップショットを取得する必要がある。このため単純な手法では、復元にかかる時間が最少であるが、スナップショット数が膨大となってしまうため、現実的ではない。図 1 に本手法のスナップショット取得の概念図を示す。たとえば、3 状態を持つ装置①、2 状態を持つ装置②と、装置①、装置②の状態に依存して状態が決定されるテスト対象で構成されるシステムの場合は、全部で 6 状態存在することとなり、本手法ではそれぞれにおいてスナップショットを取得する。

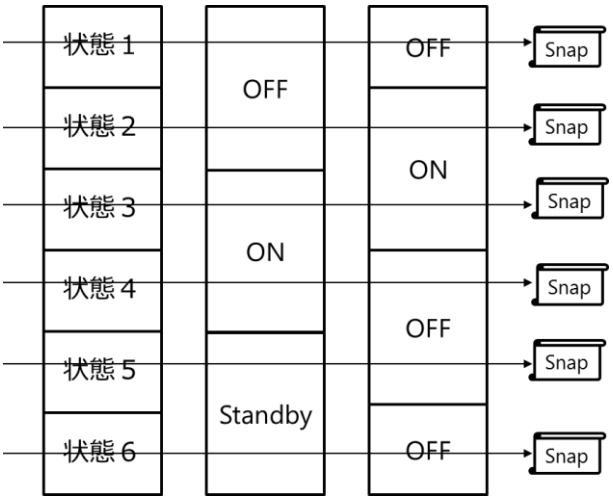

図 1　単純な手法でのスナップショット取得

2.2.2 大道らの手法[1]

大道らは、VM で動作するシステムの状態復元を、スナップショット数を削減して行うために、状態切り替えに伴う入出力データや通信をキャプチャしておき、ある時点のスナップショット以降の状態を、あらかじめ記録しておいた入出力データや、キャプチャした通信データを対象システムに対して送ることで、状態の切り替えを実現して、スナップショット数の削減を図る。大道らの手法の装置構成を図 2 に、概念図を図 3 に示す。

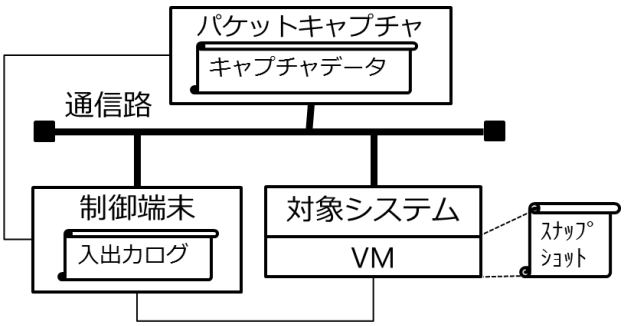

図 2　大道らの手法の装置構成

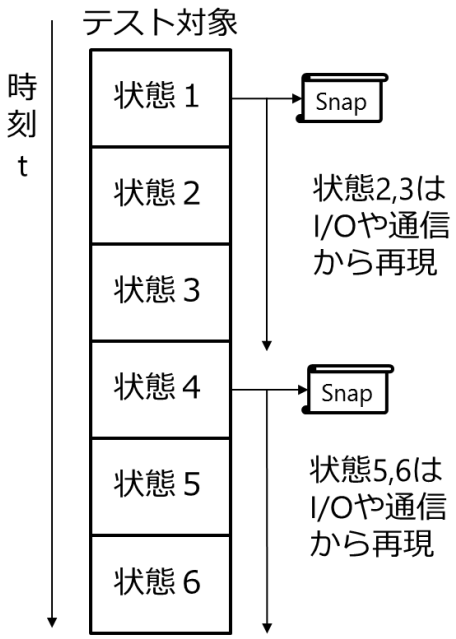

図 3　大道らの手法の概念図

しかしながら、大道らの手法のでは複数システムが連携する状態については考慮されておらず、スナップショットや I/O ログを取得するフェーズにおいては、複数システムの状態の組み合わせを網羅させるように動作させる必要があり、状態数が膨大になると、スナップショットを取得するフェーズにおいても膨大な時間が必要となる。

3. 提案方式

本節では提案方式について説明する。はじめに、ペンテスト対象のシステムに以下のような前提をおく。

① テスト対象システムが、連携システムと通信等を行い連携するシステム
② テスト対象以外のシステムは、システム間連携がなく独立して動作する
③ ペンテスト対象システムから、連携システムの状態切り替えが行われる
④ ペンテスト対象は連携システムの状態を認識している

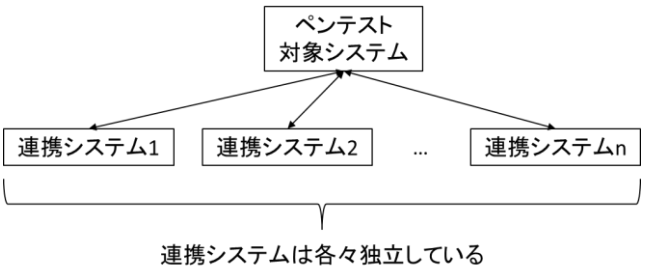

図 4　提案手法における対象システムの前提

3.1 装置構成

提案方式を構成する装置の構成図を図 5 に示す。装置はペンテストを行うペンテスト端末、テスト対象となるテスト対象端末、テスト対象と連携して動作する装置①、装置②、そして状態変化を行う通信をとらえるパケットキャプチャ装置がある。この中で、テスト対象、装置①、②は VM 上で構築され、任意のタイミングでスナップショットが取得可能である。また、パケットキャプチャ装置内には、通信の要求と応答とを紐づける通信入出力表を持ち、パケットキャプチャ中に生成する。

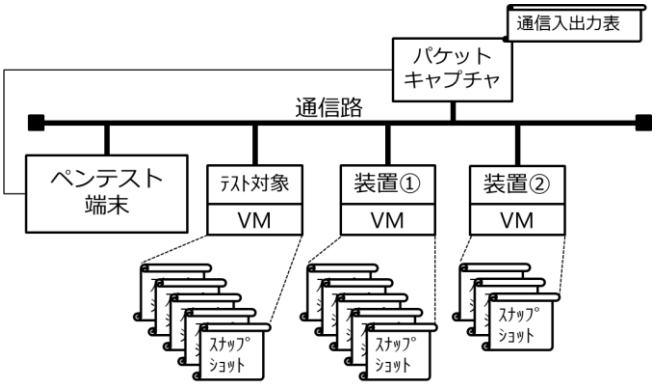

図 5　装置構成図

3.2 スナップショット取得フェーズ

ペンテスト中に状態を効率的に復元するために、VM のスナップショットを用いる。スナップショットをすべての状態の組合せだけ取れば、任意の状態を容易に再現可能であるが、スナップショットを単純に取得すると、スナップショット数が多くなり、現実的ではない。また、大道らの手法では、システムの状態組み合わせを網羅するように動作させてスナップショットを記録する必要があるため、

そのため、今回は各サブシステムの全状態の組み合わせではなく、全状態がスナップショットとして記録する方式を提案する。

例としてテスト対象が HMI、装置①が 3 状態、装置②が 2 状態を持つシステムについて、スナップショットを取得する例を用いて解説する。HMI は装置①、②の状態に応じて、HMI 自身の状態も変化すると仮定する。はじめに、スナップショットを取得するために、システムを動作させるためのシナリオを作成する。このとき、ユーザが念入りにペンテストを行いたい状態がある場合は、その状態組合せを含むようにシナリオを構成する。次にシナリオに沿って動作させ、各々のサブシステムが持つ状態についてスナップショットを取得する。

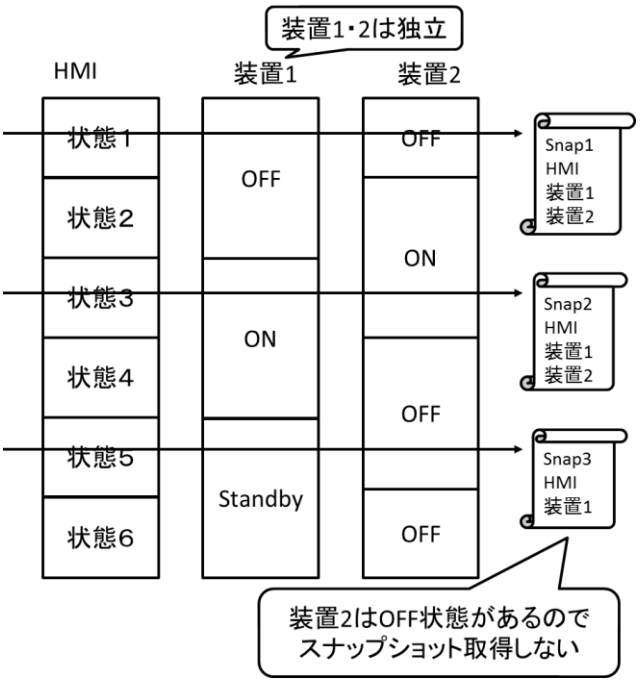

図 6　スナップショット取得時のイメージ

本手法を採用する場合のスナップショット数は、各々のシステムの持つ状態数の和と、スナップショット取得回数だけ必要となる。また、全ての状態組み合わせを網羅するように動作させる必要はないため、スナップショット取得フェーズにおける処理時間の短縮も期待できる。

3.3 状態再現フェーズ

状態を再現するためには、ペンテスターが再現したい状態を入力する。各サブシステムの状態の組合せをユーザが入力し、その状態を再現する。

はじめに、再現したい状態の組合せで取得したスナップショットが存在するかを判断する。存在すればそのスナップショットをロードし終了する。

状態組み合わせの通りのスナップショットが存在しない場合は、連携システムはその状態のスナップショットをロードする。テスト対象システムは、ユーザが指定した状態組み合わせとテスト対象が認識している状態組み合わせが一番多くなるときのスナップショットをロードする。このときテスト対象が認識している状態組み合わせとは異なる状態にある連携システムの VM を一時停止状態にして、テスト対象システムの状態不整合を修正する。

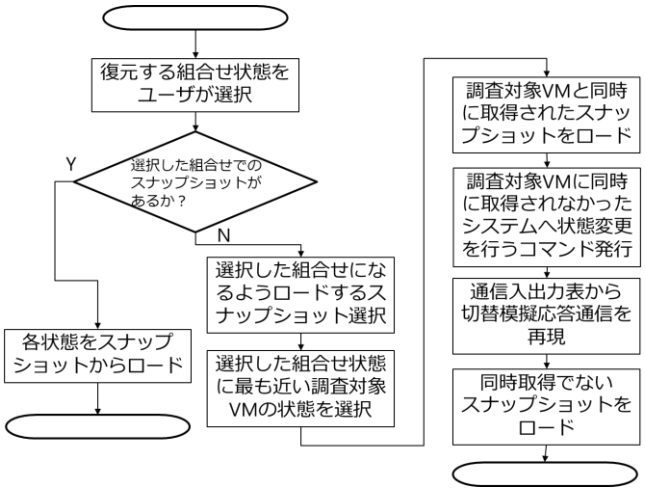

図 7　状態復元フェーズのフローチャート

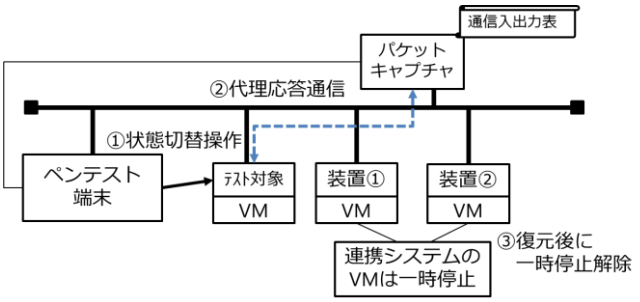

図 8　復元時の通信の流れ

テスト対象の状態不整合を解消するために､テスト対象の状態切り替え操作後に、状態が切り替わる際に発生する通信を、パケットキャプチャ装置から予め取得した通信入出力表から適切な通信応答を行う。代理応答通信によってテスト対象が認識している、連携システムの状態を整合させることで、状態復元を達成することができる。

状態復元後のシステム間の通信が不整合を起こす可能性がある。例えば TCP のシーケンス番号等の不整合が発生する可能性があるため、復元後の通信については、①セッションを一旦切断するか、②パケットキャプチャ装置がプロキシのように振る舞って不整合を解消するように通信を中継する方法が考えられる。

4. 考察

4.1 連携システムを模擬で行う場合との差異

連携システムを実システムで再現せずに、模擬プログラム等で再現して、テスト対象をファジングすることが考えられる。しかし、実際のシステム構成とは異なるので、ペンテストとしては不十分となってしまう。例えば、ペンテスト端末からの攻撃通信の内容から、テスト対象から連携システムへのリクエストが発行され、その応答がテスト対象への攻撃として機能してしまう場合などが考えられる。そのため、可能な限り実システムに近い形でペンテストを行う必要がある。

4.2 状態再現に必要なスナップショット数

各連携システムの状態組み合わせの数だけ、状態のスナップショットを取得する場合、スナップショット数は、各システムをi、システムの総数Mを、各システムが持つ状態数をs_iとすると、総状態数は$S = \prod_{i=0}^{M-1} s_i$で示される。総状態数には、システム構成によっては存在しない状態があるため、あくまでもシステムが取りうる最大状態数ということになる。単純な手法ではSだけスナップショットを取得する必要があった。

提案手法では、各システムが独立しているという前提を置くことで、スナップショット取得数は$S' = \max(s_i) + \sum_{i=0}^{M-1} s_i$となり、大幅に削減することができる。スナップショットに加えて、通信データを取得する必要があるが、一般的に VM の状態を保存するスナップショットに比べると通信データ量は少ないため、状態復元のために必要な総データ量を削減することができる。

5. まとめ

複数のサブシステムが連携して動作するシステムに対する、効率的なペネトレーションテストプラットフォームの提案を行った。仮想計算機上に再現されたテスト対象を、スナップショット機能と、機能切り替え時の通信から、任意の状態を効率的に再現することを示した。

今度の課題として、本提案方式のプロトタイプ実装と、提案方式では対象としなかったサブシステム間の連携があるシステムに対して、効率的に状態復元可能となるように改良を加える。

参考文献

[1] 日本国公開特許, 特開 2009-080705 号

[2] サーバ仮想化技術を利用したアプリケーション障害再現システムの提案, 樋口 毅ほか, 情報処理学会研究報告システムソフトウェアとオペレーティング・システム（OS） 2009(6(2009-OS-110)), p19-26, 2009

[3] David Kennedy 他, 実践 Metasploit-ペネトレーションテストによる脆弱性評価, 岡真由美(訳), オライリージャパン, Tokyo, 2012