4-4　ビルディングブロック型模倣環境構築システム Alfons とその応用事例 ～セキュリティ検証環境構築基盤と人材育成への貢献～

安田真悟

北陸 StarBED 技術センターでは主にネットワークシステムの検証のために、ネットワーク実証検証環境の自動構築ツールの研究開発を行ってきた。サイバーセキュリティ研究室では、この研究成果を応用し、マルウェア解析やサイバー演習の環境を支援する模擬環境構築システム Alfons を研究開発している。本稿では、Alfons の設計と実装、そして実際の利用事例について述べる。

1　まえがき

インターネットが社会基盤として浸透し、情報セキュリティを取り巻く環境は急速に変化している。特に、感染することでコンピュータ利用者の意図しない様々な動作を引き起こすマルウェアは、標的型攻撃と呼ばれる特定の相手の攻撃に使われる種類も出現するなど、多様化・高度化しており [1]、対策技術の研究開発が急務である [2]。

マルウェアの解析では、初期の感染活動において、マルウェアが Command and Control Server（C&C）サーバ等から追加のマルウェアを導入する場合など、静的解析だけでは全体の挙動が把握できないことも多い。また、ソーシャルエンジニアリングなどにより、あらかじめ入手した攻撃対象の環境情報がマルウェアの作成に利用されている場合もある。この様なマルウェアの解析には、攻撃対象となる個人利用の PC やオフィスの周辺機器などの攻撃対象環境を模倣した模擬環境を用意し、動的解析を行う必要がある。

他方、日本国内でセキュリティに従事する人材（セキュリティ人材）について、潜在的に 8 万人が不足しており、加えて現従事者のうち 16 万人がスキル不足であると指摘されるなど、セキュリティ人材の育成が課題となっている [2][3]。

セキュリティ人材育成には座学だけではなく、実践的な演習が欠かせない。これまで北陸 StarBED 技術センターでは主にネットワークシステムの検証のために、ネットワーク実証検証環境の自動構築ツールの研究開発を行ってきた [4][5] が、我々は、これらの研究成果を活用し、ITkeys[6]、SecCap[7]、Hardening Project[8] 等、様々なセキュリティ人材育成プログラム、イベントに協賛し演習・競技環境の構築・運営を担当している。

イベントごとに模擬環境の構成は異なり、かつ個々のイベントの開催頻度は高くないことから、模擬環境を恒常的に設置・運用するにはコストがかかるため、我々は大規模ネットワークエミュレーション基盤である StarBED 上に、その都度、環境構築と解体を行っている。しかし、StarBED は主にネットワークシステムの実証検証に重点を置いた環境構築支援の研究開発を行っていたため、ネットワーク設定以外は均一な実験インスタンス群を用いた実証検証環境を前提とした構築支援となっている。そのため、アプリケーションの設定、コンテンツや利用履歴が異なるなど、多様なノードの作り込みの支援は不十分であり、環境構築担当者が独自にスクリプトなどを用いて手作業で設定するなど、セキュリティ演習・競技会の環境構築には人的コストに課題があった。

ノード設定や OS インストールを自動化することで、環境構築を省力化することを目的としたツールとして Vagrant[9]、Ansible[10] や Chef[11] などが存在する。これらを利用することで、テンプレートとなるノードイメージの複製からノードを自動生成することも可能であるが、手続き型の処理手順を記述する必要があるなど煩雑な面もある。演習用の模擬環境は実際の企業などの組織ネットワークを縮退・模擬した環境であることから、どの演習も基本となる OS の種類、模擬環境内で提供されるネットワークサービスには重複が多い。その結果、実際の模擬環境の個々のインスタンスの差異は設定ファイルや検体、模擬ドキュメントなどのファイルの存在の差異だけである場合が多い。そのため、よりシンプルにファイルの差し換えにフォーカスしたノード生成手法を用いた方が効率的である。そこで我々は、テンプレートとなる OS ディスクイメージに個々のノードの差分となる実行バイナリや設定ファイル、ドキュメントファイルといったコンテンツ

を挿入することで、ビルディングブロック式にノードを生成し、模擬環境を構築するシステム「Alfons」を提案する。本システムによって多様なノードにより構成される模擬環境の簡便な構築と運用が期待できる。

2　環境構築と課題

模擬環境構築は、その模擬環境内で利用するノードの作成作業を基準に図１に示す３つのフェーズに定義することができる。ここでは、テストベッドにおける一般的な環境構築手順と課題を述べる。

2.1　テンプレート作成

テンプレートとなる OS を物理サーバに直接、または仮想ノードとしてハイパーバイザ上にインストールし、更に必要となるアプリケーションをインストールするなどの、共通の設定を行う。模擬環境で利用する OS が複数必要である場合は複数のテンプレートを用意する。同一 OS でもインスタンスごとにインストールされるアプリケーションが異なる場合は、それぞれ異なるテンプレートを用意するか、全てをインストールしたテンプレートを作成する。しかし、多数のアプリケーションをインストールし、テンプレートの汎用性を上げると、テンプレートイメージの管理は簡便になるが、テンプレートイメージが肥大化し、複製に時間を要するハイパーバイザのディスクを消費するなどの弊害が発生する場合もある。また、クライアント用のインスタンスに多数のサーバ用とのアプリケーションがインストールされているなど、環境の現実性が損なわれる場合もある。

このテンプレート作成フェーズの作業を自動化するために、virt-install 等の OS インストール自動化ツールがある。また、Chef や Ansible でも OS インストール後の設定、アプリケーションの導入と設定などを自動化することが可能である。しかし、テンプレート作成時の作業は目的ごとのテンプレートにつき一度の作業となる。したがって、Chef や Ansible で自動化することは、自動化により削減できる時間に対して手続き処理の記述の負荷が高い。結果的に、この工程の多くは手作業になる。昨今普及しているクラウドサービスでは、最低限のインストールがなされている各種 OS を利用者に提供することで、簡便な利用を実現している。

2.2　インスタンス作成

このフェーズではテンプレートとなるディスクイメージを基に、実際に利用する物理インスタンスや仮想インスタンスとして複製・配置する。同時に模擬環境のネットワークを構築する。VMWare vSphare[12] 等のクラウドコントローラや SpringOS では、これらの複製作業を支援する機能を有しており、ネットワーク設定も行える。ただし、複製に伴い MAC アドレス、IP アドレスやホスト名などインスタンス固有の値が変更となり、OS やアプリケーションの設定と不整合が発生する場合もあるため、複製したノードはそのままでは利用できない場合が多い。

SpringOS ではインスタンス内のネットワーク設定の変更はサポートしていないが、物理ノードとしての展開を前提として、各物理ノードに管理用のネットワークインタフェースを規定し、物理ネットワークインタフェースの MAC アドレスに基づき DHCP でアドレスを固定的に配布することで、複製後の OS を管理できる様にしている。しかし、クラウドコントローラ、SpringOS 共にアプリケーションの設定変更はサポートしていない。

2.3　インスタンス設定

作成されたインスタンスにテンプレートとの差分と

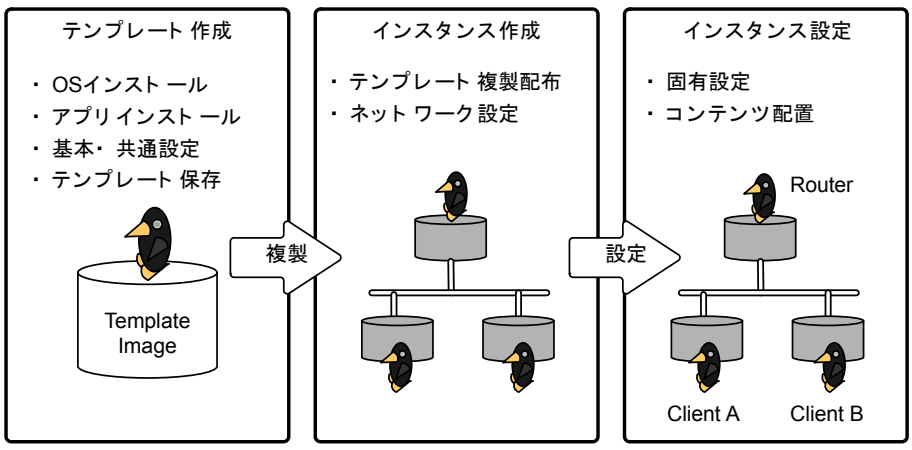

図１　模擬環境構築フェーズ

なる、各インスタンス固有の設定や実行ファイル、ドキュメント等のコンテンツの配置を行う。インスタンスの作成時にテンプレートからの複製によって発生した OS やアプリケーションの設定の齟齬なども、このフェーズで修正する。Xenebula、Anybed ではアプリケーションが利用する設定ファイルを NFS のマウント領域に配置し、各インスタンスが起動後その領域をマウントすることで個々のインスタンスの設定ファイルの配置を集約している。しかし、当該領域への IO アクセスがボトルネックになるなどの弊害がある。Chef や Ansible はデーモンやレシピを用意することで、コンテンツの各インスタンス内への配置を自動化できる。SpringOS でも K 言語で設定作業を記述することで自動化可能である。しかし、これらの手法はインスタンスに管理専用のユーザを設定したり、デーモンを常駐させたりする必要が有るため、模擬環境の用途によっては不要な痕跡が残るなどの弊害がある。

2.4　物理ノードと仮想ノードの両サポート

マルウェアの種類によっては仮想ノードでは期待した動作をしない物もある [13][14]。これは、マルウェア作成者がマルウェアを解析環境で動作しないようにすることで、挙動の解析を妨げようとするためである。これを回避するためには環境構築において物理インスタンス、仮想インスタンス両方をサポートし、必要に応じて作成できる必要がある。一般的なクラウドコントローラでは仮想ノードのみの環境構築しかできない。また SpringOS では物理ノードのみをサポートしているため、利用者がハイパーバイザのインストールを行えば仮想インスタンスも扱えるが、ネットワークブリッジの設定やインスタンスのブリッジへのアタッチ等ハイパーバイザ上の設定を環境構築運営者が管理・制御する必要がある。

3　Alfons のインスタンス作成手法

Alfons では **2** で述べた課題を解決するために、テンプレートの管理、インスタンス作成、設定を単一のシステムで実現している。本章ではそのデザインを述べる。

3.1　ファイルの種類と導入フェーズ

インスタンスに作り込むコンテンツには、そのコンテンツの依存関係の強弱によって大きく分けて 2 種類ある。1 つは主に OS が保存領域を管理し、アプリケーション、ライブラリ等が相互にファイル書き換えを行うなど、依存関係があるファイルである。この種のファイルの作成には OS やアプリケーション標準のスクリプト処理が必要である。そしてそれらのコンテンツは、サーバ、クライアント、ルータ、ソフトウェアスイッチ等そのテンプレートの利用目的を決める際に生成されることが多いため、Alfons の自動化の対象としない。

もう 1 種類のコンテンツはアプリケーション特有の設定やアプリケーションデータ等である。これらのファイルは他の OS やアプリケーションとの依存関係が薄く、変更はファイルの設置・置き換えで済む場合が多い。これらのコンテンツの設定・配置はインスタンス設定フェーズで行うが、これまでの模擬環境構築では、インスタンスを各物理ノードやハイパーバイザ上に展開・起動後に実施していた。しかし、展開・起動後の作業はリモートからの作業となるため、管理用のデーモンや、リモートログインを用いたスクリプト処理が必要である。サイバー演習など隔離環境を指向する模擬環境では管理用のデーモンの常駐や、リモートログインによる処理は、不要な痕跡が残るなどの弊害があるため、作り込みの最後にこれらを削除するなどの作業が必要となる。

ITkeys、SecCap で利用している Alfons の前身となった環境構築ツールでは、この弊害を無くすために、テンプレートから複製されたインスタンスのディスクイメージを、環境構築システムがマウントし直接編集することで、インスタンスを起動せずにマルウェアを挿入している。この手法では、インスタンスを起動しないため、マルウェア配置時にインスタンスのディスクイメージに不要な痕跡が残り難い。Alfons ではこの手法を応用し、図 2 に示す様に設定ファイルやアプリケーションデータも同様の手法で挿入しインスタンスを作成する。

3.2　ディスクイメージの共有・再利用

環境構築において、テンプレートとなる OS がインストールされたクリーンなディスクイメージの作成は時間を要する作業であるため、なるべく共有化・再利用をすることが望ましい。実際、クラウドコントローラを利用した場合には、インストール直後のインスタンスイメージをテンプレートとして登録しておくことが多い。また一般のクラウドサービスでも最低限のインストールがされた OS イメージを利用者に提供している場合が多い。StarBED でも iSCSI 接続でのノード起動において、標準的な OS のテンプレートが提供されており、それを複製してノードの起動が行える。

Alfons では標準的な OS イメージだけで無く、利用者が作成したイメージの利用者間で共有する機能も提供する。これらのテンプレートは OS をインストールしただけの物や、特定のアプリケーションをインス

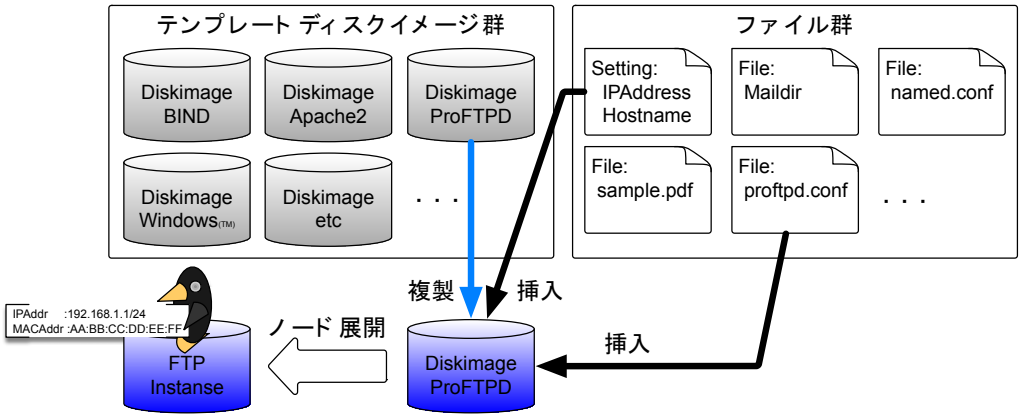

図 2　ビルディングブロック形のインスタンス作成

トールした状態の物など様々なスナップが考えられる。いたずらにテンプレートの種類を増やすことは、保存用のストレージ領域を消費すると共に再利用される頻度が少なくなるが、テンプレートとして提供する OS の多様性を低コストで実現するためには、利用者同士の共有が効果を発揮すると考える。

また、Alfons ではテンプレートイメージからインスタンスを作成する時に、物理サーバに直接インストールされた物理インスタンスと、ハイパーバイザ上に仮想ノードとしてインストールする仮想インスタンスの両方を、単一テンプレートから作成する機能を有している。

3.3　リソースの制御

模擬環境をサーバクラスタ上に構築するためには、物理ノードの電源制御、OS イメージの導入やネットワークの設定等、物理資源の各種設定や制御が必要となる。SpringOS では電源の制御、VLAN によるネットワークの設定を行うデーモン群をプログラムから呼び出すための API モジュール群を提供している。Alfons は StarBED の利用を想定しているため、これらの物理リソースの制御にはこの API モジュールを利用している。一般的なクラウドコントローラでも同様の API を提供しているため、他の環境への移植も可能である。

仮想化ノードを用いたインスタンスを作成する場合、物理リソース制御だけで無く Hypervisor の設定・制御も必要となる。SpringOS では実験設備として提供している物理リソースの制御用 API しか提供していないため、Hypervisor の設定制御は利用者に任されていた。Alfons は独自に仮想リソースの制御を行い、SpringOS が提供する物理リソースの制御と連携することで、物理リソース、仮想リソースの制御を統合的行う。現在の実装では、ハイパーバイザとして Linux KVM と VMware vSphere Hypervisor に対応している。

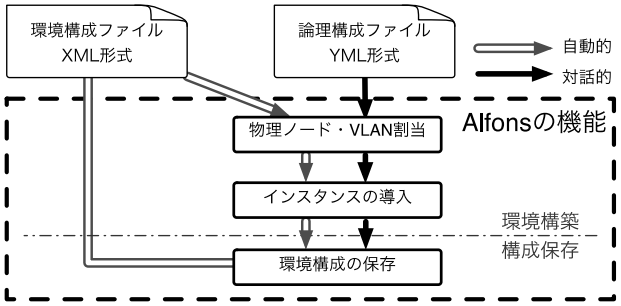

図 3　環境構築フロー

4　Alfons の環境構築フロー

Alfons では **3** で述べたインスタンス作成手順を基に、図 3 に示す 2 種類の環境構築フローで模擬環境の構築を行う。本章ではその手順と環境構築で用いる構成記述ファイルの概要を述べる。

4.1　環境構築フロー

Alfons では、リソースの論理構成ファイルを基準に、物理リソースの割当てから、実験ノードの配置までを CLI により対話的に行う逐次構築と、環境全体を記述した環境構成ファイルによる環境の一括構築の両方をサポートしている。対話的な環境構築フローでは論理構成ファイルを作成し、これに実際に割り当てる物理ノード ID と VLAN ID を指定し、逐次インスタンスの導入を行う。Alfons は指定されたテンプレートに、インスタンス ID に紐付くコンテンツを挿入し、物理サーバに物理インスタンスまたは仮想インスタンスの指定された種別で作成・導入する。この時、ハイパーバイザは必要に応じて Alfons が自動的に物理サーバに導入するため、利用者はハイパーバイザの管

理は不要である。加えて Alfons では対話的に作成・変更した環境の構成を、環境構成ファイルとして出力する機能を有している。Alfons はこの環境構成ファイルを元に、全自動的な環境構築も可能である。この環境構成ファイルは、XML に準拠したファイル形式であり、物理リソースと論理リソースを対応づける情報も含まれており、論理リソースの ID に紐付けられている物理リソース ID 等を変更することで、同一または一部異なる環境を容易に定義可能である。

マルウェア解析や・演習では、解体後に同じ環境または以前の環境を改変した物を再利用することも多い。Alfons では環境構成ファイルに基づく一括構築機能により環境の再構築・複製等の実験環境構築を自動化し、環境構成の容易な再利用を可能としている。

4.2　構成記述フォーマット

模倣環境を構築支援するためには、攻撃対象となったユーザ・組織のネットワーク情報をシステムが解釈可能な記述フォーマットが必要である。Alfons は **4.1** で述べたとおり、目的別に 2 種類のフォーマットに基づいた構成記述ファイルを用いて環境構築を行う。1 つめは模擬環境の物理資源上での論理構成を YAML 形式で記述する論理構成ファイルである。2 つめは論理構成ファイルで記述された論理資源上に実際に構築された模擬環境の構成を XML 形式で記述する環境構成ファイルである。論理構成と割り当てられた物理資源を紐付けると共に、順次環境構築を行った結果を記述する設定ファイルで、この設定ファイルにより構成の保存・再利用を可能にする。

5　活用事例

Alfons 及び Alfons の前進となったツールを用いて北陸 StarBED 技術センターでは、大規模エミュレーション環境である StarBED 上に様々な演習・競技環境を構築し、人材育成の支援を行っている。本稿では代表的な例として 3 つの例を上げる。

5.1　ITKeys, enPiT-Security【SecCap】

ITKeys は文部科学省「平成 19 年度 先進的 IT スペシャリスト育成推進プログラム」として、関西圏を中心とした情報系 4 大学院（奈良先端科学技術大学院大学、大阪大学大学院、京都大学大学院、北陸先端科学技術大学院大学）及び 4 企業・団体（NICT、情報セキュリティ研究所、JPCERT コーディネーションセンター、NTT コミュニケーションズ）の、情報ネットワークの管理・運用の現場でリーダーシップを発揮し活躍できる技術者・実務者を育成することを目的とした産学連携型の教育拠点形成プロジェクトで、平成 19 年度～ 22 年度まで行われた。主に情報系 4 大学院から希望する学生を集め、4 大学院でキュリティスペシャリストを育成するコースとして講義・演習が行われ、各参画組織が分担して講義・演習を行っていた。その中に、北陸 StarBED 技術センターでの 3 日間にわたる合宿形式のマルウェア解析演習が組み込まれており、毎年 20 数名程度の受講生が北陸 StarBED 技術センターに集まってきた。受講生たちは StarBED 上に構築された演習環境で、実際にインターネット上で採取されたマルウェア検体を用いて、マルウェア種別、感染源、感染経路、対策などを学習した。

enPiT-Security【SecCap】は ITKeys の流れをくむ形で、文部科学省「分野・地域を越えた実践的情報教育協働 NW―セキュリティ分野―」として 5 つの連携大学（情報セキュリティ大学院大学、奈良先端科学技術大学院大学、北陸先端科学技術大学院大学、東北大学、慶應義塾大学）で平成 23 年度より行われている人材育成プログラムで、ITKeys 同様に北陸 StarBED 技術センターで合宿形式の ITKeys と同様の演習が行

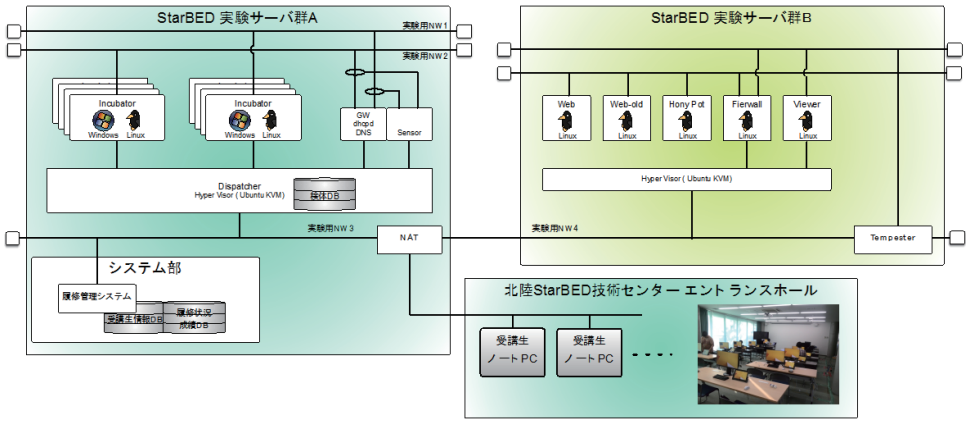

図 4　ITKesy,SecCap 演習環境例

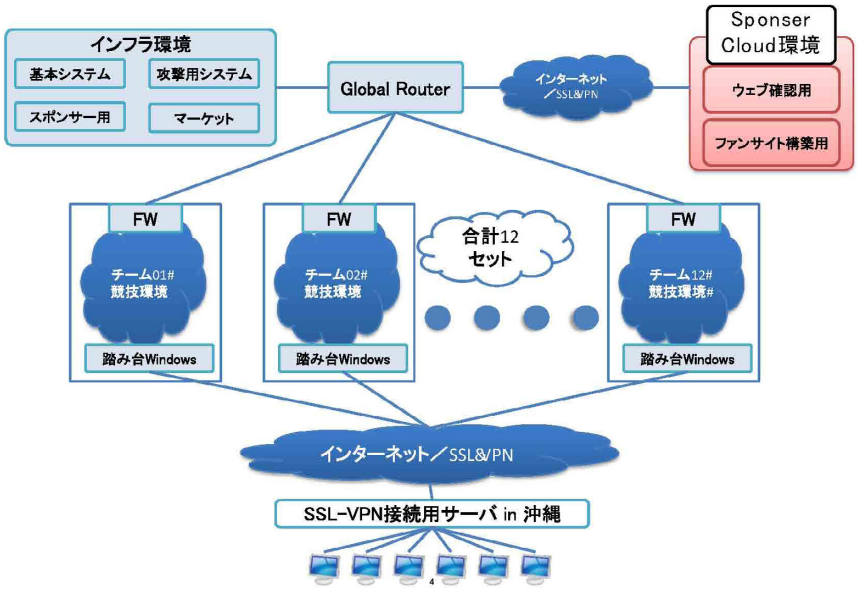

図 5　Hardening 10Market Placet 競技環境概形

われている。

ITKeys、SecCap では Alfons の前身となった環境構築ツールにより、マルウェアの種類や演習の種類に伴い演習環境を変化させながら、環境構築図 4 の様な演習環境を構築している。受講生は 2 名 1 チームに分かれて、それぞれの受講生に貸与されるノート PC 等を用いて、StarBED に作成された演習環境の Windows や Linux 端末を操作する。各チームの環境は複数台のネットワークインスタンスで構成されており、操作端末に感染しているマルウェアの種別の識別や、感染源となった端末の特定、対処法を考え、レポートを発表する形式で演習が行われる。

本演習は実際にインターネット上で収集されたマルウェア検体を用いており、得られた検体のハッシュ値や挙動、プログラム名を、検索エンジンなどを用いて学生が情報収集するなど、他では行われない実践的な内容になっている。この様な演習は大規模な隔離環境が必要となるため、通常のクラウドサービスや各大学が有する機材では実施できない。大規模に柔軟な隔離環境を構築できる北陸 StarBED 技術センター特有の演習である。

5.2　Hardening Project

Hardening Project は WASForum Hardening Project 実行委員会が 2012 年から企画・実行しているイベントで、NICT が特別協賛して北陸 StarBED 上に競技環境を構築している。Hardening Project の最大の特徴は、一般的なセキュリティ競技イベントは攻撃側のスキルを競うケースが多いのに対して、最高の「守る」技術を持つトップエンジニアを発掘・顕彰するために、一貫して守る技術を競うイベントとなっていることである。参加者はチームを組んで仮想の EC サイトを運営し、運営者側からのリアルタイムの攻撃を 8 時間耐久で受け続けながら、サイトの売り上げを競うという競技形式で行われる。2015 年度実施の Hardening 10 MarketPlace からは、Market Place の概念が導入され、ウィルス対策ソフトやセキュリティ診断サービスなどのプロフェッショナルサービスを競技中に各チームが売り上げの中から購入できるルールになり、よりプラクティカルな競技に進化している。当初 Hardening Project は東京で開催されていたが、現在は国際大会への進化を視野に会場を変更し、年 2 回沖縄で開催している。

図 5 は Hardening 10 MarketPlace の競技環境のトポロジ概形、図 6 は 1 チーム分の環境トポロジである。各チームの環境は 22 台のインスタンスで構成されており、沖縄会場から VPN を介して StarBED 上に作られた競技環境のサポート端末（Windows）にリモートデスクトップで接続し操作を行う。また、現在は直接攻撃が及ばない一部の環境をスポンサー企業のクラウド環境上にも作り、StarBED に接続することで、マルチクラウドの競技環境となっており、総計 335 インスタンスの大規模な競技環境である。

図 7 は各チームの手持ち資金の推移である。各チームは 1,000 万円の仮想資金をもって競技をスタートする。商品の在庫購入やセキュリティ商品の購入を行った場合、手持ち資金が減少するため、グラフが下がる。攻撃を受けサイトがダウンするなど顧客（購入クローラプログラム）の商品購入ができなくなると、グラフが横ばいになる。また、Hardening 10 MarketPlace

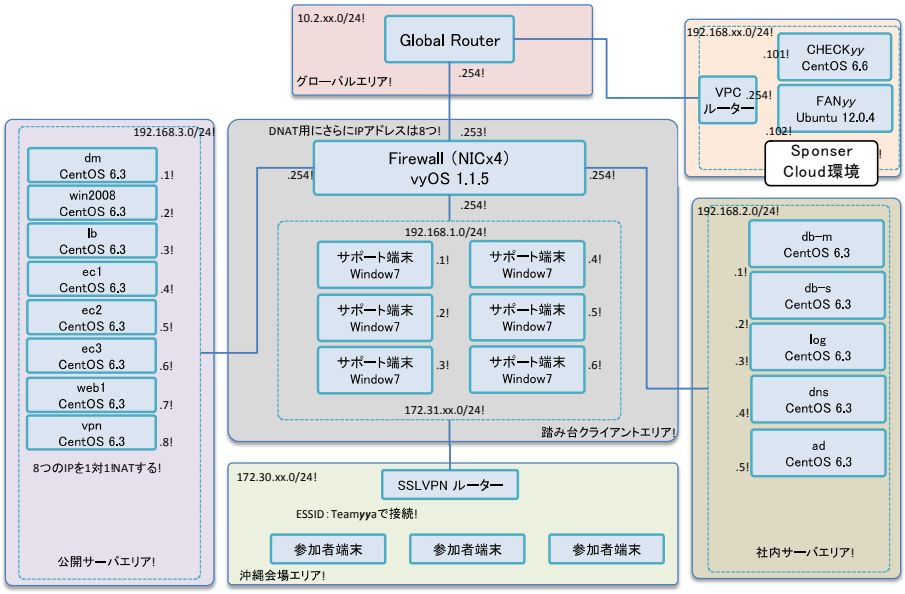

図 6　Hardening10MarketPlace EC サイト環境 (1 チーム)

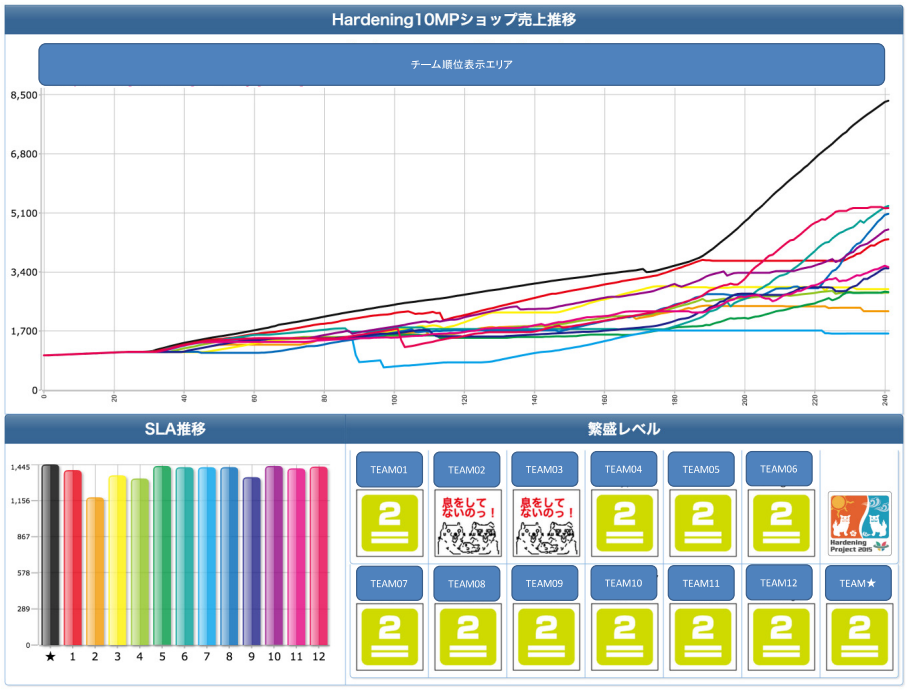

図 7　Hardening10MarketPlace 手持ち資金推移グラフ

からは Service Level Agreement（SLA）の指標が導入された。この指標は繁盛レベルとして機能し、EC サイトがインシデントに正しく対応するなどの健全なサイト運営を行うと SLA レベルが上昇し、顧客の購買速度が上がり売り上げの上昇が早くなる。運営側は攻撃者の役割だけで無く、サイトの不具合についての問い合わせメールや、サイトの改ざんの指摘等の顧客としてのメールのやりとりや、参加者チームの社内の承認者としての社長としてのメール対応なども行い、これらのメールのやりとりも SLA に反映される。このため Hardening Project 競技は単なる技術競技では無く、戦略、チームマネジメント、意思決定などのヒューマンファクターも勝因となる。また、参加者の職種は学生からエンジニア、事務系の職員まで多岐にわたっており、組織としてのセキュリティ対応の総合的なスキルを競う競技となっている。

当初 Hardening Project の競技環境構築は環境規模が比較的小さかったこともあり、SpringOS を用いて構築していたが、初回開催を除いて全て仮想インスタンスで作成されており、SpringOS でハイパイパーバ

表 1　環境構成規模 (物理ノード数とインスタンス数)

開催名	開催日程	開催地	インスタンス種別	1チームあたりのインスタンス数	競技環境合計インスタンス数	利用物理ノード数
Zero	2012/04	東京	物理	5	43	43
One	2012/10	東京	仮想	10	114	26
One Remix	2013/07	東京	仮想	17	154	30
10 APAC	2014/06	沖縄	仮想	22	154	30
10 Evolutions	2014/11	沖縄	仮想	6	88	16
10 MarketPlace	2015/06	沖縄	仮想	22	335	57
10 ValueChain	2015/11	沖縄	仮想	21	275	49

イザの複製・導入をした後は手作業で環境構築を行っていた。これらの知見を基に研究開発途中であった Alfons に各種機能拡張を行った上で、2015 年 6 月開催の Hardening 10 MarketPlace の環境構築から Alfons を導入した。表 1 は Hardening Project 競技会の利用資源の一覧である。競技会の回を重ねるごとに、徐々に環境が大きくなっている。そして Alfons を導入した Hardening 10 MarketPlace と前後して競技環境が飛躍的に大きくなっている。しかし、Alfons の導入の結果、構築負荷が軽減されたため、競技環境の作成人員にはほぼ変化が無く、特にインスタンスの展開作業、ネットワーク設定は数名で行った。

6　まとめ

サイバーセキュリティ研究室では、ビルディングブロック形模倣環境構築システム Alfons の研究開発を行っている。本システムはマルウェアの動的解析のための模倣環境の構築のみならず、セキュリティ人材育成のための演習環境の構築にも利用可能である。そして、前身となったツールも含め、本稿で述べた 3 つの演習以外にも、大小様々な演習環境の構築を行っている。

セキュリティは国家、企業、個人にとって今後更に重要となる。また、ICT 機器の普及とともに人々により身近な課題となっている。そのため、サイバー演習も現在の職業人の育成にとどまらず、防犯訓練、避難訓練など一般的な旧来の物理的なセキュリティ訓練の様に、一般の人々も受講可能な体験型演習も必要とされると考えられる。

サイバーセキュリティ研究室では今後も様々な演習環境の構築ニーズに応え、低コストで多様な演習を実現できる環境構築技術の研究開発及び人材育成への支援と貢献を行っていく。

【参考文献】

1　Mandiant APT1: Exposing One of China's Cy- ber Espionage Units. http://intelreport. mandiant.com/Mandiant_APT1_Report.pdf, 2013.
2　内閣官房情報セキュリティセンター. サイバーセキュリティ戦略. http: //www.nisc.go.jp/active/kihon/pdf/ cyber-security-senryaku-set.pdf, 2013.
3　独立行政法人情報処理推進機構 . 情報セキュリティ 人材の育成に関する基礎調査 . http://www.ipa. go.jp/files/000014184.pdf, 2012.
4　宮地利幸 , 中田潤也 , 知念賢一 , ラズバン・ベウラ ン , 三輪信介 , 岡田崇 , 三角真 , 宇多仁 , 芳炭将 , 丹 康雄 , 中川晋一 , 篠田陽一 . StarBED 大規模ネット ワーク実験環境 . vol.49, no.1, pp.1–14, 2008.
5　Toshiyuki Miyachi, Takeshi Nakagawa, Ken ichi Chinen, Shinsuke Miwa, and Yoichi Shinoda. StarBED and SpringOS architectures and their per- formance. In TRIDENTCOM, vol.90, pp.43–58, 2011.
6　ITKeys. 先導的 IT スペシャリスト育成推進プロ グラム . http://it-keys.naist.jp, 2015.
7　SecCap. 分野・地域を越えた実践的情報教育協働 nw - セキュリティ分野 -. https://www.seccap. jp, 2015.
8　Hardening Project.「守る技術」の価値を最大化 することを目指す、全く新しいセキュリティ・イ ベント . http://wasforum.jp, 2015.
9　Vagrant. https://www.vagrantup.com, 2015.
10　Ansible. http://www.ansible.com/home, 2015.
11　chef. https://www.chef.io, 2015.
12　VMWare vSphere. http://www.vmware.com/products/vi/, 2015.
13　Martina Lindorfer, Clemens Kolbitsch, and Paolo Milani Comparetti. Detecting environment-sensitive malware. In Proceedings of the 14th International Confer- ence on Recent Advances in Intrusion Detection, RAID'11, pp.338–357, Berlin, Heidelberg, 2011. Springer-Verlag.
14　Detecting Malware and Sandbox Evasion Techniques. https://www.sans. org/ reading- room/ whitepapers/ forensics/ detecting- malware-sandbox-evasion-techniques-36667, 2015.

安田真悟　（やすだ　しんご）
サイバーセキュリティ研究所
サイバーセキュリティ研究室
研究員
博士 (情報科学)
サイバー演習環境構築、ネットワークテストベッド