サイバー演習環境統合管理システム CABIN 2016 活動内容

太田悟史 (sota@nict.go.jp)　　高野祐輝 (ytakano@wide.ad.jp)
安田真悟 (s-yasuda@nict.go.jp)　　湯村翼 (yumu@nict.go.jp)

2016 年 12 月 25 日

1　はじめに

このメモでは, サイバー演習統合管理システム CAIBN について, 2016 年の春 (3 月) と秋 (9 月) の WIDE 合宿における活動内容を報告する.

春の WIDE 合宿では個人用演習環境のデモンストレーションを行った. ここでは演習者の操作端末上での挙動の監視に焦点をあてている. 秋の合宿では競技環境としてチームの優劣を判断する評価機能等を盛り込み, イベントとして ”オペレーション大会 ” を開催した.

2　CABIN

CABIN はサイバー演習環境の統合管理システムである. 詳細については DICOMO2016 を参照されたい [1]. サイバー演習環境には環境の構築支援や, モニタリング (監視と観測) 機能が必要であり, 同時にリアリティに対しても配慮が必要である. 春と秋の合宿では,PoC モデルを用いてこれら機能について検証を行った.

3　春合宿デモ

春合宿ではデモンストレーションとして, 個人用の演習環境を用いた競技を行った. 競技内容は, 与えられた演習環境の中について一定の時間内にどれだけ調べられるかを競うものであり, 演習環境内の踏み台役となる端末に接続した状態から競技を開始する. 演習者はネットワークに関する設定情報や, メール等のサービスの提供状況やアカウント情報等を調査する. 演習の開始と終了のタイミングは定めず, 演習の希望がある都度演習を実施することとしたため, 演習者の結果の優劣はデモンストレーション終了後に演習者各自が記述した調査レポートを元に行った.

3.1　演習環境

図 1 は春合宿デモでのネットワーク構成図である. 演習環境そのものは StarBED [2] に構築されており, 合宿会場のデモ用端末から VPN 接続を行い, 演習環境内部の Windows クライアントへ VNC を用いてリモート接続する. 図のノードやネットワークは全て仮想マシン, 仮想ルータで構成されており, 1 つの演習環境は 1 台の物理マシン内部に構築されている. 他クライアントやサーバーへは, 環境内に用意されたメール内の情報や, 利用可能となっている特権昇格ツールを用いる事などによりアクセスが可能となっている.

演習環境の構築には Alfons [3] を用いている. 1 度作成したサーバーやクライアントをコンポーネントとして利用する事により, 複数の演習環境の構築を容易に行える. 演習環境の構築支援の他にも, 構築時間が早いため, 演習環境の再構築 (利用済環境のバックアップと新規作成) として利用した.

今回はクライアントとなる 2 つの Windows 端末にキーロガーを稼働させておき, 演習者の挙動を記録する事とした. キーロガーはキーの押下情報の他に, 操作対象となる Window タイトルや, マウスのクリックイベントについても記録している.

3.2　デモンストレーションまとめ

春と秋を通じて, リアリティへの配慮とモニタリング機能について, 個人向け/イベント向けのサイバー演習環境について検証を行った. 春のデモについては次のとおりとなっている.

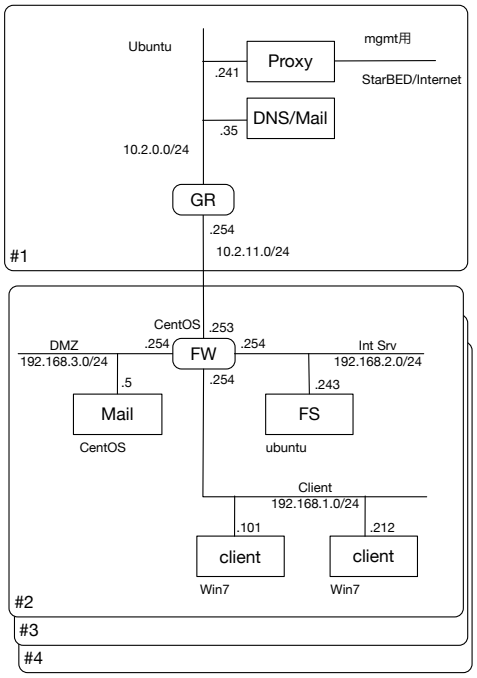

図 1: 春合宿演習環境構成図

3.2.1 モニタリング

春のデモでは, 演習者の挙動の把握を目的として, キー入力情報を取得する. 踏み台となるクライアントに GUI での挙動を確認するため, 操作対象となるクライアント 2 台にキーロガーを用意した. 履歴ファイルはクライアント内部に保存されるため, 挙動の確認は演習の後となっている. そのため, 演習者によるクライアントでの操作によっては, キーロガーが演習中に機能を停止したり, 履歴ファイルが消去される可能性がある.

3.2.2 リアリティ

演習者にとってより臨場感を与えるために, 次の 2 つのリアリティについて考慮した.

- 動的なリアリティ
  演習者以外の第 3 者が演習環境に存在するかのような臨場感のため, 演習者の操作端末にメールの着信通知が表示されるようメールの自動送付機能を設けた.
- 静的なリアリティ
  演習者の操作対象となるクライアントに対して, 本来のユーザーによる生活感を持たせるため, ユーザーによる作成ファイルなど, 初期状態からなるべく”汚す”ようにした.

利用感のある端末やネットワークをある程度実現できたが, 演習者によっては, 動作プロセスの状況などからモニタリング用のツールを発見する場合があった. 演習を支援する機能が演習者の目に付く事は本来の演習操作に支障をきたす場合があるため, 演習者に認識されにくい対応が必要である.

4 秋合宿オペレーション大会

秋合宿では与えられた演習環境について, 一定時間の間にどれだけ脆弱性を改善できるかを競うチーム対抗でのオペレーション大会を催した. オペレーション大会はイベントとして開催したため, 観衆向けに演習者の操作の様子やチームとしての優劣などの状況を逐次確認できる必要があった. そのため今回, 評価を判断するためのモニタリングを行っている.

4.1 演習環境

図 2 はオペレーション大会のネットワーク構成図である. 演習環境は春のデモンストレーションと同様に,

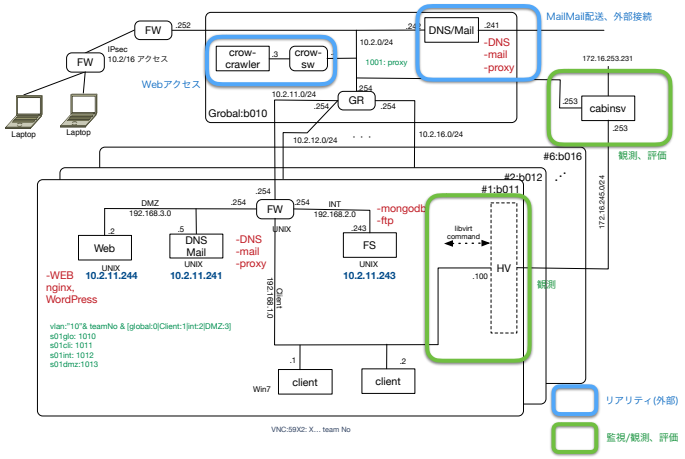

図 2: 秋合宿演習環境構成図

遠隔の StarBED に構築されており, VPN 接続で会場からアクセスする. 春のデモンストレーション時と同様に,1 物理マシンに 1 演習環境が設けられている. 今回は操作大会という事もあり, 使い慣れたインターフェースを持つ演習者各自のノート PC を使って演習環境へアクセスする. 演習環境は中小企業のネットワークトポロジを想定し, よく用いられているサービスとしてメール,HTTP,DNS, ファイルサービスが稼働している. この演習環境に, 脆弱性がある状態をいくつか用意した. 脆弱性については, インターネットで検索すると容易に得られるものやメジャーなものを数多く揃え, 高度なセキュリティの専門知識ではく, 運用スキルに重点を置いたものとした.

聴衆に向けては, 前述の評価結果と共に, 各演習者の状況を確認するため, 操作端末の画面をミラーリングしてプロジェクターに投影するようにしている.

4.2 オペレーション大会のまとめ

秋の大会でのモニタリングおよびリアリティについては次のようになっている.

4.2.1 モニタリング

各演習者の挙動については, 各自の PC からアクセスした先のサーバーの shell の history ファイルで確認することとし, 主に演習中に評価を行うためのモニタリング機能について検討した. 今回は競技中に随時優劣を確認するため, 課題のクリア状況を得る仕組みが必要となる. 例えば課題の 1 つに「社内向けのみのサービスとして稼働させる必要がある」がある場合, その確認のためには次の状況を把握する必要がある.

- 外部向けの公開状況
  FW の外部からのアクセスへの対応
- 内部向けの公開状況
  FW の内部ネットワークでのアクセスへの対応
- サーバー起動時の対応起動時オプションでの, 次回サーバー起動時の起動漏れ対策

これらを確認する方法として, FW 外部からのサービス確認用パケット送信, FW 内部からのサービス確認要パケット送信, Hypervisor からの設定ファイルの内容確認, などが挙げられる. 仮想マシン上のファイル操作には,libguestfs を用いている. 以上のように,1 つの課題に複数のモニタリング結果を用いながら全課題を通じて優劣を決定している. 今回のように,1 台の評価用サーバーから演習環境内の状況を走査する場合, 課題と演習チームの数に比例して処理時間が増加する.

4.2.2 リアリティ

演習者にとってのリアリティとして今回, 稼働中のサービス (メールや WEB ページ等) へ外部からのアクセスがある状況を作り出している. メールについては春の合宿同様に各演習環境の外部からメールサーバーを用いてメールを送信する事とした. WEB ページへのアクセスとして,CROW を用いている. CROW [4]

によってソースアドレスやヘッダのブラウザ情報を変えながらアクセスする事により, あたかも外部から多くの利用者に閲覧されているかの状況を, ネットワークを流れるパケットやサーバー内部のアクセスログ的に作り出している. このパケットの生成は, 臨場感の他にモニタリングで用いている死活確認用パケットを隠す目的も含まれている. 演習者の操作への影響を最小に抑えるため, モニタリングによる不自然な状態の隠蔽が必要となる.

今回, 演習中に死活パケットの存在を発見した演習者が現れた. HTTP サーバーのアクセスログを確認した所, 死活監視パケットの HTTP ヘッダの設定に漏れがあったため, 目立つ結果となっていた. またそれ以外にも, アクセスページなどに変化がなく, 作為的なアクセスを感じさせる状態になっていた.

4.3 まとめ

春と秋合宿を通じて, リアリティに配慮しつつモニタリングを行うサイバー演習環境について検討した.

4.3.1 モニタリングについて

演習者の挙動については, 春, 秋ともにキーロガー (ログ蓄積型) や history ファイルなどの”履歴”で確認していた. 秋合宿では演習者の挙動は画面をスクリーンにミラーリングし, リアルタイムで観衆が見られるようにしていたが, これはモニタリングを観衆に委ねたものであり, 演習環境中の機能としては不足している. イベントとしての演習環境では優劣を判断するため, 課題となるサービスの死活確認や設定ファイルの記述内容の確認を行っていた. 回答用サーバーを持たず, 演習環境の状況で判断する演習形式は臨場感を損なわない反面, 課題の増加や確認内容が複雑化するに従い, 必要となるモニタリング項目が増える.

モニタリング 機能は, 演習操作に影響を及ぼさないように配慮が必要であるが, 春, 秋の演習においては, キーロガーのプロセスや死活確認用のパケットの存在が演習者に認識されてしまった. これにより, 演習者にキーロガーのプログラムを検索させてしまったり, 観測用パケット中の GET オプションからヒントを与えてしまったりと, 演習者の行動に影響を及ぼす結果となってしまった. 演習環境としてのモニタリング機能の隠蔽について, 今後考慮が必要である.

4.3.2 リアリティへの配慮について

演習環境としてのリアリティとして, 演習者が操作するクライアント (のユーザー) へのメールの送付やサーバーへのアクセストラフィック等, 外部とのコネクティビティの存在を演習者に伝えるようにした. 秋の演習環境での, クライアントが接続するネットワークセグメントでは, 用意したクライアント 2 台からのトラフィックしかなかった. 前提としている中小企業の規模としては不足しているため, 規模に即したトラフィックの生成が必要である. 演習者が操作する環境でのリアリティとして, 長期間での利用状況の表現がある. クライアントでは, ユーザーが作成した雑多なファイルや, 蓄積されたメール,WEB のアクセス履歴などがあり, サーバーでは自身の稼働時間や, 長期に渡るサービスログなどが挙げられる. 特に演習内容として過去に遡っての調査作業など時間的な状態遷移の痕跡が必要となる場合, 時系列に沿ったログやファイル等のリソースを提供できる必要がある.

今後も引き続き, サイバー演習環境におけるリアリティと監視/計測について検討を重ねながら, CABIN について研究活動を継続する.

参考文献

[1] 太田 悟史, 安田 真悟, 湯村 翼, 高野 祐輝 *“***次世代サイバー演習環境に向けて***”*, マルチメディア, 分散, 協調とモバイル (DICOMO2016) シンポジウム.

[2] 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証環境***”*, 情報処理, 第 49 巻, 第 1 号, pp. 57-70, 2008 年, 1 月.

[3] 安田 真悟, 三浦 良介, 太田 悟史, 高野 祐輝, 宮地利幸, *“***ビルディングブロック型模擬環境構築システム***”*, インターネットコンファレンス (IC2015).

[4] 湯村 翼, 高野 祐輝, 安田 真悟, 宮地 利幸 *“CROW: OpenFlow* **を用いた動的** *Web* **アクセス模倣システム***”*, マルチメディア，分散，協調とモバイル (DICOMO2016) シンポジウム.

[5] libguestfs.org, *“tools for accessing and modifying virtual machine disk images.”*, `http://libguestfs.org`