ユーザビリティとセキュリティを両立したセキュリティスキャナシステムの開発

吉本 道隆† 加藤 貴司†† Bhed Bahadur Bista††
高田 豊雄††

今日，セキュリティ専門家らによって，セキュリティの分野においてユーザビリティとセキュリティ（機能性や信頼性など）はトレードオフの関係にあると信じられている [1]．そのため，ユーザビリティ工学に基づいたユーザビリティとセキュリティを両立したセキュリティツールの開発は進んでいなかった．結果として，セキュリティの高いツールにおいてユーザビリティは考慮されておらず，ユーザビリティの高いツールはセキュリティが十分でないという現状である．

そこで本研究ではユーザビリティとセキュリティを両立したセキュリティツールの開発を行う．既に筆者らはユーザビリティとセキュリティの両立性について議論している [2]．しかし文献 [2]はプロトタイプの作成で留まっており，実運用に耐えるレベルの開発と実装には至っていない．そこで実際に開発と実装を行い，ユーザビリティを考慮したセキュリティツールの開発における，実際の困難性や実現性などについて明らかにする．

Development of Security Scanner System with Usability and Security

Michitaka Yoshimoto† Takashi Katoh††
Bhed Bahadur Bista†† Toyoo Takata††

Presently available security products are not usable for all users because there is a gap between the image of users the product developers have and the users who actually use the products. Additionally, it has been believed that usability and security (functionality, reliability and so on) of the products do not coexist. In this paper, we discuss development of a security tool with usability and security and we investigate the difficulty of it.

†清泉女学院大学人間学部
††岩手県立大学ソフトウェア情報学部

1. 背景

今日，不正アクセスは内容も巧妙かつ悪質となり，また同時に件数も増加傾向にあり，不正アクセスのターゲットは個人ユーザから大企業まで，相手を問わない傾向になっている．そのため，全てのユーザに適切なセキュリティに関する対策が求められている [1]．もしユーザが基本的なセキュリティ対策を行っていれば，ほとんどの不正アクセスを未然に防ぐことが出来たとも報告されているが [1]，ユーザはセキュリティに対する知識を持ち合わせているとは必ずしも限らない．

一方で，ユーザビリティとセキュリティ（機能性や信頼性など）はトレードオフの関係にあると言われている [3]．そのため，ユーザビリティ工学に基づいたユーザビリティとセキュリティを両立させたセキュリティツールの開発は進んでいなかった．その為，多くのセキュリティ技術やツールが開発・提供されており，それらの適切な利用によってセキュリティ確保が充分に可能な状態にあるにも拘らず，現実にはそれらの積極的な導入に至らず，結果として前述の文献 [1]にある通り既知の脆弱性を悪用した攻撃が後を絶たない．即ち，既存ツールのユーザビリティの低さがセキュリティ対策の遅れを招く結果をもたらしていると考えられる．よって信頼性や機能が充分確保された既存ツールのユーザビリティの改善されたものが提供されれば一層のセキュリティの確保・向上が期待される．

我々は文献 [2]において，ユーザビリティとセキュリティの両立可能性について述べ，それらの両立可能性を示した．しかし文献 [2]において作成したシステムはユーザビリティテストを目的としていたため，プロトタイプ作成までに留まっている．本論文では実際に開発と実装を行い，ユーザビリティを考慮したセキュリティツールの開発における，実際の困難性や実現性などについて明らかにする．本論文では，セキュリティツールに関する作業例題としてセキュリティスキャナを採り上げる．

本論文では第 2 章においてこれまでの成果について述べ，第 3 章において実装を行ったセキュリティスキャナシステムについて述べる．そして第 4 章において実際の困難性や実現性について議論する．

2. これまでの成果

本章では文献 [2]などにおける成果について述べる．

2.1 セキュリティスキャナ

セキュリティスキャナとは，ネットワーク上の端末に対して，その端末の脆弱性の有無を調べるツールである．通常操作しているだけではわからない脆弱性を発見し，ユーザにその発見された脆弱性とその対処法を通知する．ユーザはセキュリティスキャナの通知内容に従って脆弱性の対処を行う．セキュリティスキャナは既知の脆弱性のほとんどをスキャンすることが出来る機能性を持ち合わせ，脆弱性の有無の情報を

視覚的にユーザに与える．しかし前述の通り，セキュリティスキャナも一般ユーザにとって使いづらいものであることが知られている．また近年，クライアント型セキュリティスキャナはアンチウィルスソフトなどの一機能として提供されている場合が多いが，ネットワーク型セキュリティスキャナはほとんど認知されていない．ネットワーク型セキュリティスキャナはネットワークを介したクライアントに対して脆弱性診断を行うため，設定に起因した脆弱性の診断等に有用である．そのため文献 [2]ではネットワーク型セキュリティスキャナに着目した．

セキュリティスキャナは強力な脆弱性診断ツールではあるが，既存のものはユーザビリティ問題を多く抱えている．2.2，**エラー! 参照元が見つかりません。**節にそのユーザビリティ問題の解消を目的とした開発方法について述べる．

2.2 開発方針

文献 [2]で開発したセキュリティスキャナはユーザ中心設計 [5]に基づいて設計し，また，ISO9241-11 で示されるユーザビリティ要件を満たすものとした．より具体的には，まず始めにユーザ調査を行い，実際のユーザの利用状況を詳細に把握する．その結果に基づき，まずはプロトタイプを構築し，形成的評価を行う．プロトタイプに潜在する問題点の発見と改善を繰り返し行い，問題点を全て解決していく．そして最終的に総括的評価を行うという過程をとるという過程をとる．このような方法論を踏まえることによって，ユーザニーズを十分に満たしたセキュリティスキャナを開発することが出来る．

2.3 既存製品のユーザビリティ調査とそれに基づく開発

筆者らは既存製品の中から Nessus の Windows クライアントソフトウェアである NessusWX を選択し，予備調査を行った上で具体的な問題点把握を，思考発話法を用いて行っている．そのインタフェースを図 1 に示す．

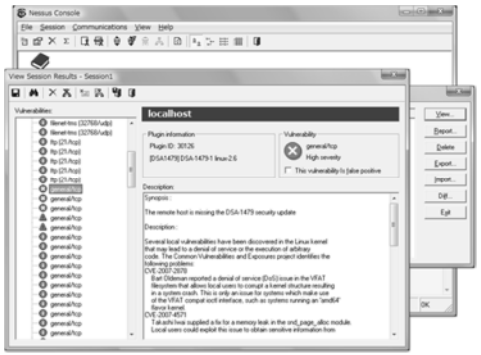

図 1 NessusWX のインタフェース
Figure 1 Interface of NessusWX

致命的な問題点として，起動後に次に何を行えばよいかわからない点があり，またサーバ接続時に入力するクライアント側の IP アドレスがわからない点など，診断すら受けることが出来ないという致命的な問題点を抱えている．また，診断に成功したとしても診断結果において表示される専門用語がユーザにとって難解であったり，解決法が把握出来なかったりした結果が多く見られた．さらに，解決法が把握出来たとしても，被験者がパッチを 1 つ適用して完了したと思い込んだり（本調査においては実験端末には脆弱性が 12 個存在するため，Windows Update を複数回行うことが必要となっている），Windows Update を 1 回行っただけで完了したと思い込んだりしたケースが多数見られた．そもそも今回の調査における NessusWX による起動から診断，脆弱性修正までにかかった平均時間は約 56 分と，実環境であれば使用中止をするユーザが現れかねないことが容易に想像出来，NessusWX はユーザブルでないことがわかる．

NessusWX を日本語インタフェースにしたペーパタイププロトタイピング [4]においても同様の結果が出ており，既存のセキュリティスキャナは到底ユーザの使用に耐えうるものでないことが明らかとなっている．

以上の結果を踏まえ，システムの改善を行うこととした．サーバプログラムである Nessus はそのまま使用し，機能性は保持する．インタフェースを Web ベースとすることでサーバとの接続，クライアント IP の入力の手間を省き（デフォルト値としてアクセス元の IP が入力されている），Flash を用いることによって進捗状況を明確に示し，親しみやすく，OS やブラウザやバージョンに依存しないインタフェース開発を容易に行えるようにした．また，初期状態では最小限のインタフェース提供を行い，ユーザのニーズによって詳細なインタフェースが提供される仕組みを採った．診断結果の表示においても脆弱性情報は基本的に表示せず，解決法を明示し，さらにそれに対する補足情報を追記する仕組みを採った．その他詳細は文献 [2]を参照されたい．

最終的に NessusWX と改善システムにおいて総括的評価を行い，改善システムの改善具合が数値で示され，改善に成功，すなわちセキュリティとユーザビリティの両立に成功した．

3. 実装を行ったセキュリティスキャナシステム

本論文で実装したセキュリティスキャナシステムについて詳細を述べる．

3.1 核となるシステム

提案するシステムは Nessus をセキュリティスキャナの核として採用し，新たにインタフェースの開発を行った．スキャンスクリプトの生成や動作チェックや管理は莫大なコストがかかり，そもそもいくつかのオープンコミュニティがある現在において別のセキュリティスキャナを 1 から作り直すことは車輪の再発明であると判断した

ためである．インタフェースが Nessus と直接通信を行い，Nessus をそのまま動作させることによって，提案するセキュリティスキャナは信頼性をそのまま引き継ぐことができる．

3.2 Adobe Air（インタフェース）

本論文で提案するセキュリティスキャナのインタフェースには Adobe Air を用いることとした．Adobe Air は HTML，JavaScript，Adobe Flash，Adobe Flex など様々な開発環境から簡易に作成することが出来，Flash 以上に柔軟な表現力を持つ．また，Adobe Air のランタイムは Adobe Air アプリケーションをインストールする際に自動的にインストールされ，これは Windows，Mac，Linux においても簡便であることを筆者らは確認している．

インタフェースと Nessus の機能は多対一で結ばれている．ユーザビリティ調査より，インタフェースは初期の状態では必要最小限のインタフェース提供に留め，ユーザが希望する場合に必要なインタフェース提供を行うこととした．また，Adobe Air が Nessus と通信を行うためのパーサも開発している．これにより，核とするセキュリティスキャナとして Nessus が利用できなくなった場合においてもパーサを再開発するのみで解決する仕組みとした．

図 2 に Windows 環境において初回時に“.air ファイル”をダブルクリックした際に現れるインタフェースを示し，図 3 に各プラットフォームで表示される Hello World を示す．

Adobe Air は Windows，Mac，Linux に加え，Android や iOS 端末においても動作する．従ってインターネットに接続されるほとんど全ての端末において動作可能である．このことから提案するセキュリティスキャナシステムは OS に依存しないといっても過言ではなく，Adobe Air が動作するすべての OS とそれらのバージョンの組合せにおいて同一の動作が行われるため，開発コストの削減にもなる．

Adobe Flash から Adobe Air にインタフェースを変更した理由として，Adobe Air は接続するサーバに制約がないこと，ローカル環境に対するファイルの読み書きが行えることから変更を行った．前者により，必ずしもサーバクライアントモデルのセキュリティスキャナシステムでなければならない理由は無いこと，もしくは複数のサーバに接続が可能なこと，また後者により，脆弱性情報をローカルにのみ保存が出来ることから，サーバにクライアント情報を残す必要がないといった理由が挙げられる．

このシステムのインタフェースは初期状態では必要最小限のインタフェース（ターゲット IP アドレスを入力するテキストフォーム，次の画面へ遷移するボタン，拡張機能へ遷移するボタン）のみを提供する．拡張機能へ遷移するボタンを選択することによって，ユーザは詳細なオプション設定を行うことができる仕組みをとっている．このインタフェースの詳細なオプション設定は Nessus の機能と 1 対 1 で対応しているため，Nessus の機能性を何ひとつ失っていない．これは対象物のユーザビリティを向上させても，機能性を保持している，すなわち両立させていることを示している．

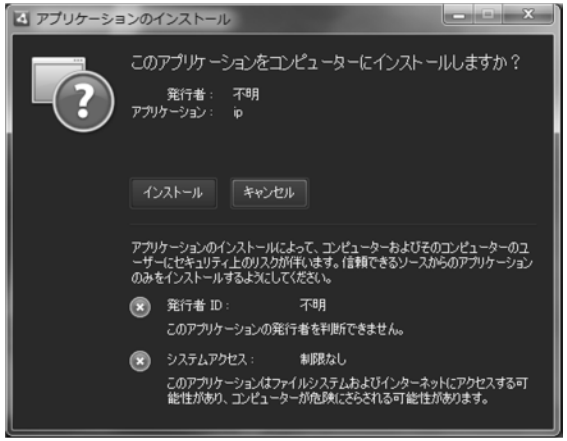

図 2 Windows 環境における Adobe Air インストール画面
Figure 2 Snapshot of Installation of Adobe Air

図 3 Adobe Air 実行結果
Figure 3 “Hello World” in Adobe Air

3.3 ユーザサポートデータベース

現在配布されているセキュリティスキャナは結果表示として脆弱性の説明文と同等の解決法を記述している．しかし文献 [2]の形成的評価の結果より，そのような情報はユーザを惑わすだけであることがわかった．なぜなら CERT/CC，CVE，SecurityFocus，JVN などが提供している脆弱性情報はほとんどのユーザにとって膨大

で平板であるか，かなり難解であるかいずれかであるためである．

そのため本実装ではユーザをサポートするためのサポートデータベースを開発することとした．これはセキュリティスキャナが出力する結果表示を，ユーザが充分に理解出来るように補足を追記したり，わかりづらい単語に対する簡易な辞書機能を備えたりするなどしたデータベースである．事前にサポートデータベースは前述の脆弱性情報提供ベンダーやそれが参照するサイトなどから自動的に，XML 形式で配布されていれば収集・解析を行い，HTML や TXT 形式であれば収集した後に可能な限り情報の解析（たとえば特定ベンダーが一定フォーマットによって提供されているのであればその特徴を用いて解析を行う．多くのベンダーはそれぞれ一定のフォーマットで情報提供を行っている）を行い格納する．たとえば Microsoft 社のソフトウェアや OS に存在する脆弱性は Microsoft Update によって修復可能であり，サポートデータベースはユーザに対して Microsoft Update の使用を勧め，それの使い方も簡素に説明する．もし Microsoft Update による脆弱性対処が不可能である場合はパッチを当てることを勧め，サイトの使い方を説明する．

3.4 脆弱性診断の流れ

システムの概要図を 図 4 に示す．このセキュリティスキャナシステムの流れは以下の通りである．

まず，ユーザはクライアントソフトとして Adobe Air により実装されたインタフェースを起動する（図 5）．インタフェースはサーバ端末に設置している Perl を CGI として呼び出す. Perl はクライアントから送信された内容を解析し, Nessus に送信する．Nessus はそれに基づいて診断のためのプロセスを生成し，クライアント端末に対して診断を行う．この時，CGI はクライアント端末と Nessus のブリッジ役を行っており，これにより Nessus からローカル環境に対して出力している進捗情報のような情報をインタフェースである Adobe Air に送信を行うことが出来る．すなわちユーザは進捗情報などの情報をリアルタイムに取得することが出来る（図 6）．

診断を終了したとき，もし脆弱性が無い状態の場合はその旨をインタフェースは表示を行う. 脆弱性が発見された場合(図 7), インタフェースは CGI から渡された Nessus が出力する情報を基にユーザサポートデータベースから情報を取得する．さらにその情報を基に前述の通り必要最小限，もしくはユーザが希望する場合は詳細な情報を出力する．ユーザはそれらの情報を基に脆弱性の修正作業を行う（図 8）．以上が一連の流れである．

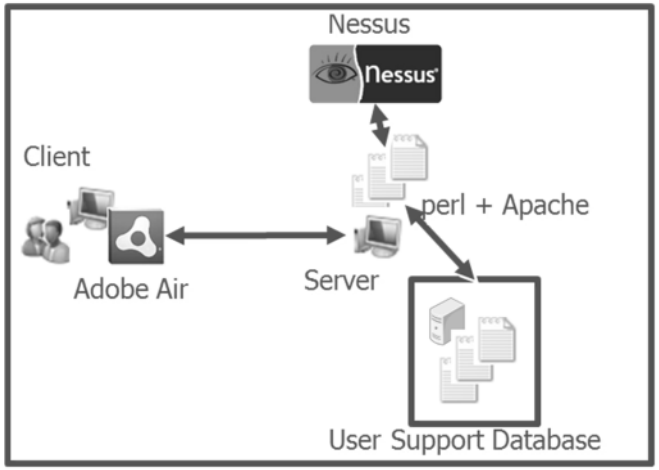

図 4 システムの概要図
Figure 4 Outline of the scanner system

図 5 開発したインタフェース 1
Figure 5 Interface 1

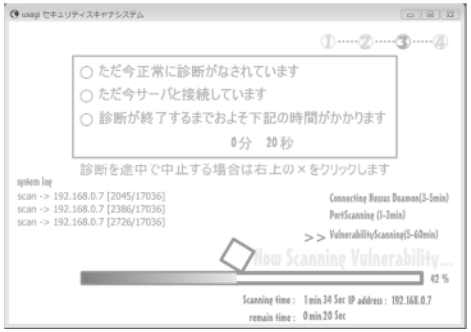

図 6 開発したインタフェース 2
Figure 6 Interface 2

図 7　開発したインタフェース 3
Figure 7 Interface 3

図 8　開発したインタフェース 4
Figure 8 Interface 4

4. 実際の困難性や実現性

4.1 実際の困難性

実際の開発と実装を行って発見された問題点としては主に，ユーザサポートデータベースに起因するものが挙げられる．難解な単語の見せ方や，補足情報の過不足など議論する点が多いことがわかった．

脆弱性のように，単語を置き換えることによってわかりやすくなる場合は良いが，クロスサイトスクリプティングや DDoS 攻撃のような単語は，わかりやすく説明文を伝える際にどうしても長文になりがちであり，いかに端的にみせるかは重要な課題点として残る．しかしこの問題点は今後広く配布を行い，意見を多く集約することによって解決出来る問題であると予想される．

また，今日配布されている OS のほとんどは自動アップデート機能を持ち合わせている．その為，ほとんどの脆弱性対策はアップデート機能の利用を促し，その利用方法を説明することで足りる．しかし設定に起因する脆弱性が発見された際の補足情報は設定の見直しを促す程度が限界である．何故なら必要があるが故に設定を変更しているのであり，本システムはユーザに対し脆弱性があるが本当に今の設定で良いのかと問いただすしか出来ないためである．従って現在，設定に起因する脆弱性に対しては一律して問うことに留まっており，今後の課題である．

一方で，ベンダー情報をユーザサポートデータベースに変換するシステム構築時に，同一ベンダーであるのにも関わらず，年代によって異なる形式で提供しているベンダーがあり，多くの例外処理を余儀なくされている．今後の情報更新の際に変換が正常に行われているかを注意深く観察する必要がある．

4.2 実際の実現性

本システムの導入難度は必ずしも高くないが，手間がかかる仕様となってしまったため，システム全体の簡略化も検討に入れるべきであることも発見された．現在は Perl を CGI として動作させるために，Apache が導入されているが，パーサ群に簡易な HTTPD を内包した方がスマートである．また，ユーザサポートデータベースは CSV 形式と XML 形式が混在しているが，セキュリティの観点から全てを XML のような Adobe Air が直接扱うことが出来る様な形式で構築するべきであること発見された．

また，本システムは VMware のような仮想環境で配布を行う予定であるが，VMware Player もしくは VMware Server の導入難易度については充分な検証を行えなかった．導入自体はさほど困難を感じることはないだろうと予想するが，クライアント OS のシステムを大きく変更する仕様のソフトウェアであるため，この点は改めて検証を行う必要がある．

以上のことから，ユーザビリティとの両立に一定の成果が見られてはいるが，更にユーザビリティの向上可能性があることが発見された．

5. まとめ

本論文では文献 [2]で提案されたプロトタイプを実システムとすることによる仕様の変更、また困難性や実現性について議論を行った。ユーザビリティの観点から、プロトタイプから実システムにするに当たり、実システムをもっと洗練することによりユーザビリティをより高める余地があることがあることが発見された。今後の課題として、実際に広く配布を行い、意見の集約を行い、セキュリティスキャナシステムのユーザビリティとセキュリティを向上させることが挙げられる。

謝辞　本研究は一部、科学研究費補助金 基盤研究(C) 23500094 の助成を受けている．

参　考　文　献

1) 独立行政法人情報処理推進機構： 2010 年のコンピュータ不正アクセス届出状況. (オンライン). http://www.ipa.go.jp/security/txt/2011/documents/2010all-cra.pdf. (参照: 2011-5-15)

2) 吉本道隆, 加藤貴司, Bhed Bahadur Bista, 高田豊雄： ユーザビリティ工学に基づくユーザビリティとセキュリティを両立したセキュリティスキャナのインタフェースの開発と評価，情報処理学会論文誌，51 巻，2 号，pp.529‐541，2010.

3) Cranor, F.L. and Garfinkel, S.: *Security and Usability*, p.xi, Oreilly & AssociatesInc.

(2005).
4) Snyder, C.: *Paper Prototyping*, Morgan Kaufmann Pub., San Fransisco, CA (2003).
5) Norman, A.D. and Draper,W.S.: *User Centered System Design*, Lawrence ErlbaumAssoc Inc, Hillsdale, NJ (1986).