ネットワークシステムにおける脆弱性管理
—その仕組みと手法—

郷間　佳市郎

インターネットの発展にともない，セキュリティ問題が社会の注目を集めている．セキュリティ問題の原因の多くは，システムに存在する脆弱性の問題である．脆弱性をいかにして管理し解決するかが，セキュリティ問題を解決するための鍵であるといえる．本稿では，近年大きな発展があった効率的な脆弱性の検知手法を紹介するとともに，検知された脆弱性の危険度をいかにして管理するかといった手法について，数値化の手法などを中心に最新の取組みを紹介する．

キーワード：インターネット，セキュリティ，脆弱性

1．はじめに

コンピュータ同士をつなぐネットワークは，もはや日常生活の中にまで入り込んできている．何気なく使っている日常生活のインフラが，ネットワークにつながっていたりする状況になってきており，生活のインフラとして重要度が以前に増して高まっている．

しかし，生活の一部となりつつあるネットワークではあるが，その安全性についての問題がある．例えば，コンピュータウィルスやワームなどである．

かつて，ウィルスやワームが，フロッピーディスクのような外部記憶媒体を通して伝播していた頃には，その感染力はたいしたものではなかった．しかし，インターネットの普及により，世界中のコンピュータ間をネットワークというリアルタイムのパイプラインを通してデータが行き来するようになって，様相が一変した．世界中に瞬時でつながるという仕組みが成立したことによって，世界のある地域で発生したウィルスやワームが，あっという間に世界中に拡散するといったことが，実際に発生するようになってきている．

2．脆弱性管理の必要性

このような，いわゆる「セキュリティ問題」の原因は，システムに存在する脆弱性の問題が，その多くの部分を占めている．脆弱性とは，システムの欠陥のことを言う．本来は許されない行為が，プログラムの設計やシステム設定上のミスにより，可能となってしまう場合がある．このような，設計者の意図しない操作がコンピュータシステムに対して行われてしまう可能性のある欠陥を総称して「脆弱性」と呼ばれている．ワームやウィルスは，このようなシステム上の欠陥を悪用して，コンピュータに対して不正な行為，例えば，勝手にコンピュータ内の文章を電子メールで送ったり，コンピュータ内の文章を破壊して利用不可能にしたりといった行動をするわけである．

つまり，ウィルスやワームによる被害を抑えたいと考えた場合，その対策として有効なのは脆弱性を解決することである．もし，脆弱性が解決されていれば，ウィルスやワームといったものによって攻撃をされても，コンピュータが不正な動きをすることはない．

脆弱性と攻撃，そしてそれによって発生する被害の間には次のような関係が成立つ．

① 脆弱性＋攻撃＝被害の発生

攻撃が発生しても，その原因となる脆弱性が存在しなければ，被害は発生しない．

② 脆弱性の情報公開日≦攻撃の発生日

攻撃が発生する前に，その原因となる脆弱性の情報は，すでに公開されている．

つまり，脆弱性というものを管理できれば，攻撃に対して耐性のあるシステムを構築することは可能であることを，上記の関係は示している．実際にそのような取組みを全社レベルで行っていたため，世界的なワームの流行から被害を免れた企業は多い．そのような企業ではどのような取組みを行っているのであろうか．

その一つの例が，「脆弱性管理」と呼ばれるものである．コンピュータウィルスやワームといったものは，健全なコンピュータには基本的には感染しない．常に，

きょうま　けいいちろう
京セラコミュニケーションシステム㈱
〒108-8605 港区高輪 2-18-10

社内のコンピュータシステムを，脆弱性という観点で管理することにより，被害発生の危険の原因を事前に解決していたわけである．

3. システムの安全度は，放置すれば必ず劣化する

コンピュータシステムが複雑化した現在，脆弱性というものがなくなることは，現在のところ考えられない．常に新しい脆弱性が発見され，それに対する解決策が発信され続けている．したがって，システムの安全度は，放置されれば必ず劣化する．これに対する解決策としては，脆弱性の検査をできるだけ短いタームで行い，安全度（セキュリティレベル）をキープし続けることが必要である（図1）．

しかし，実際のシステムの現場では，このような脆弱性の検査は，システムの構築時にだけ行われ，その後は放置される場合が多い．また，検査を行っている場合でも，その間隔は1年，あるいは半年に1回といった場合が多いのが現状である．原因は，検査を人手で行った場合，そのための費用が多くかかってしまっていたことや，診断時にコンピュータに過度の負荷を与えてしまうような検査が行われ，システムが停止してしまうことがあったからである．

しかし，最近では，このような検査を専用のソフトウェアやアプライアンス（専用の機器）によって自動的に行われることが一般化してきている．また，後述するが，診断対象に過度の負荷を与えない検査方式が実用化され，システムを停止させることなく検査を行えるようになってきている．

これによって，これまではperiodic（断続的）に行われてきた脆弱性の診断が，continuous（継続的）に行えるものとなったことによって，診断結果は，「点の情報」から「線の情報」に変化してきた．つまり，これまで，「脆弱性診断」と単に呼ばれていたものから，「脆弱性管理」といった，継続的な情報収集と管理を意味する考え方が生まれてきたわけである．

4. どのようにして診断対象に対する負荷を抑えているのか

継続的な診断を実現し，脆弱性の管理を可能とするためには，診断対象に対する負荷を抑えた脆弱性検査の方法が必要となる．これによって，常時診断を行うことが可能となり，診断結果を継続的な管理情報として扱えるようになる．では，実際にはどのような技術が，常時診断を可能としているのであろうか．その代表的な方式が，プロファイリングによる方法である．

5. プロファイリングによる診断対象の絞込み

脆弱性診断によって診断対象に過度の負荷がかかってしまったり，最悪の場合には停止してしまうことの第一の原因は，診断対象に送信される大量の診断パケットの多さにある．脆弱性診断を行う専用のソフトウェアやアプライアンスは，診断ルールというものをもっている．ちょうど，潜水艦がソナーによって敵を見つけるのと同じように，このルール一つごとに，いくつかの通信パケットを診断対象に送信し，それに対して返ってきた通信パケットの情報から，脆弱性情報を診断をするわけである．この診断ルールの数であるが，製品によっても異なるが，多いものでは1,000種類近

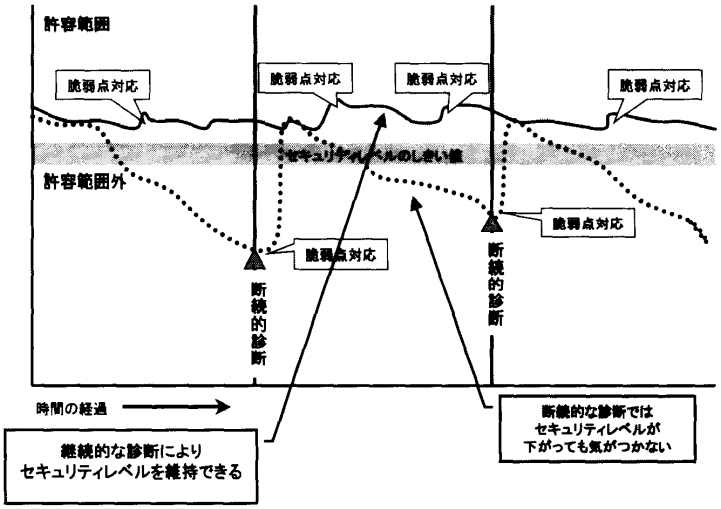

図1 安全度（セキュリティレベル）をキープし続けることが必要

くのルールを持っている．これらすべての診断を1台の診断対象に対して行った場合，大変な量の通信パケットを診断対象に発信することとなり，大きな負荷を与えてしまう．

実は，個々の診断ルールは，一つ以上の特定のOSやアプリケーションと紐づいている．例えば，あるルールはWindows NT 4.0 SP 2以前のバージョンのためのものであったり，またはCisco IOSだけのものだったりする（すなわち，NimdaやBlasterの脆弱性に関する診断はUNIXマシンに対しては意味がないからである）．このようなことから，診断対象が何であるかをあらかじめ見極められれば，そこに存在するはずのない脆弱性の診断ルールを，ルールセットの中から削っていけば良い．言い方を変えれば，そこに存在することが予想されるものだけを診断すれば良いというわけである．

このような，診断対象をはっきりさせ，対象を絞り込むことを「プロファイリング」という．この技術の進歩が，過度な負荷をかけない脆弱性診断を可能にしている．診断対象が何であるかを絞り込み，必要のない診断ルールを自動的に削り，本当に必要な診断のみを行うということによって，診断時の通信パケットは大幅に軽減されるわけである．

6. 様々な「スキャン」を組み合せた，最新の脆弱性診断

では，どのようにして，プロファイリングを行っているのであろうか．実は，いくつかのスキャンを組み合わせ，それぞれのスキャンの結果情報を組み合わせることによって，判別を行っている．脆弱性の診断もスキャンの一つであるが，これに先立って，様々なスキャンを行い，診断を行うルールを絞り込んでいく．

製品によっても異なるが，次のようなスキャンを組み合わせることによって，プロファイリングが行われている．

① アドレススキャン

診断対象の存在を確認するのが，このスキャンである．存在しないIPアドレスに対して，この後のプロファイリングを行っても非効率的であり時間の無駄である．最も簡単な方法としてpingに対する応答があるかどうかによって判断を行う例がある（図2-1）．

② ポートスキャンとプロトコルスキャン

診断対象の存在がはっきりしたならば，次に，その対象のネットワークへの入口がどうなっているかを探

・ **対象となるコンピュータを探す行為**

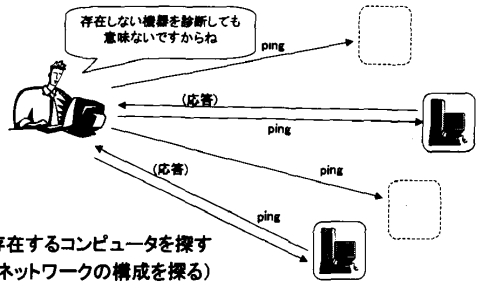

図2-1 アドレススキャン

・ **対象となるサーバの「入口」を探る**

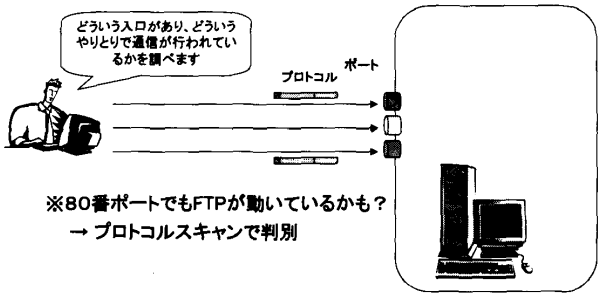

図2-2 ポートスキャンとプロトコルスキャン

・ **対象となるサーバの中身を探る**

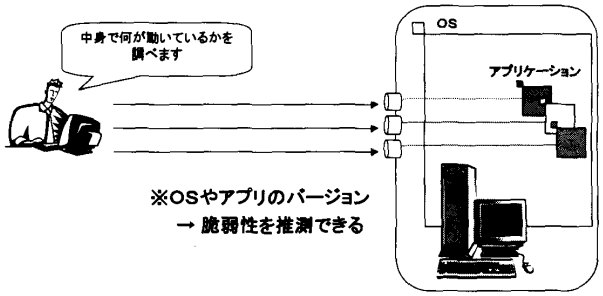

図2-3 OSスキャンとアプリケーションスキャン

・ **脆弱性の確認を行う**

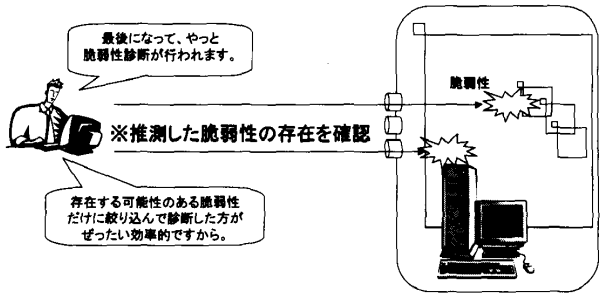

図2-4 脆弱性診断

るためのスキャンが行われる．これがポートスキャンとプロトコルスキャンである．ポートがわかれば，どのようなアプリケーションが稼動しているのかを予想できる．そしてさらに，そのポートが応答するプロト

コルを判別することによって，さらに確定的な情報を収集する（図2-2）．

③ OSスキャンとアプリケーションスキャン

診断対象入口の状況を把握できたならば，その中で稼動しているOSやアプリケーションの判別を行う．これがOSスキャンとアプリケーションスキャンである．これによって，診断を行うべき対象がかなりはっきりする．存在することのない診断ルールを動的に削除することが可能になるのである（図2-3）．

以上のような，いくつかのスキャンを行った結果から，実際の脆弱性のスキャンが行われる（図2-4）．この時までには，行うべき脆弱性の検査ルールはかなり絞り込まれている状態となるわけである．

7. 脆弱性の危険度を数値化する試み

これまで述べてきたように，脆弱性を管理するために継続的な診断はできるようになってきたが，その結果を統計的に処理するためには，数値化といったアプローチが必要である．従来，脆弱性の危険度は，高（High），中（Middle），低（Low）といった，かなり大雑把な区分けで，その危険度が管理されてきたと言える．これに対して，最近，このような危険度を「数値化」し，定量化して管理しようという動きがある．

図3は，米国のnCircle Network Security社が採用している，脆弱性の「数値化」の数式である（www.ncircle.com）．nCircle Network Security社は，数値化のために，「脆弱性の情報が公開されてからの日数」，「脆弱性自体の脅威」，そして「攻撃を成功させるためのスキルセット」の，三つの要素を用いている．次は，その内容である．

① 脆弱性の情報が公開されてからの日数

発見されたばかりの脆弱性の危険度は高くないが，その後は急激に危険度は上昇する．しかし，その危険度は，どこまでも急激に高くなるというものではなく，ある一定の段階でその高まりは鈍化し，落ち着く．このような変化は，$y=\sqrt{x}$ という曲線で表され，この数値を算出する仕組みになっている．

② 脆弱性自体の脅威

リモート（つまり遠隔操作）からアクセスでき，かつ管理者権限が奪取されてしまうような脆弱性は，非常に危険度が高いと．これに対して，ローカルでアクセスしなければ（すなわち，サーバそのものに入り込んでからでないと）攻撃が成功しないような脆弱性の危険度は，低い．このように，リモートから管理者権限が奪取されてしまう脆弱性ほど危険度が極端に高くなるような，$y=x!$ という式でこの数値を算出する仕組みになっている．

③ 攻撃を成功させるためのスキルセット

マウスをクリックするだけで，攻撃が成功してしまうようなツールが出回っているものは，スキルがない人（例えば中学生）でも攻撃ができてしまい，危険度が非常に高い．これに対して，実行に専門的なスキル（例えば，C言語やマシン語でのプログラム）が必要なものは，危険度は低い．このように，専門的なスキルが必要なものであれば，急激に危険度が下がる $y=1/x^2$ という曲線で表わし，この数値を算出する．

これらの数値を，図3に示される数式により処理す

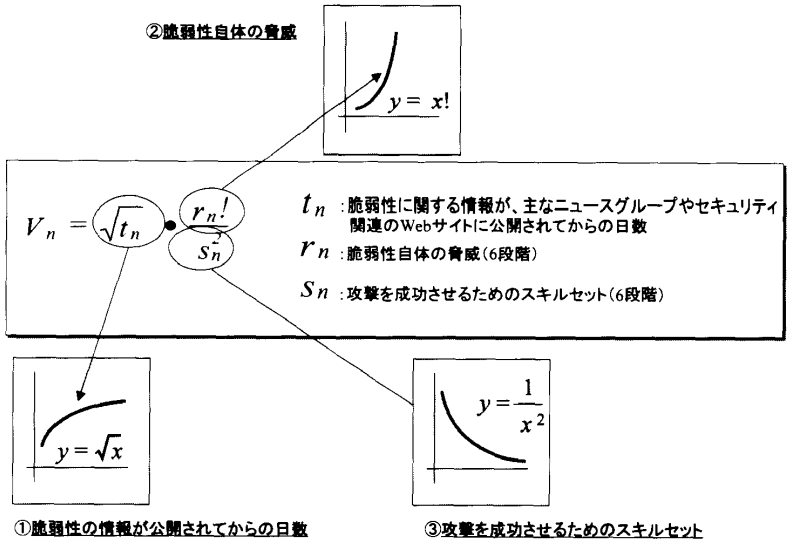

図3 脆弱性の数値化

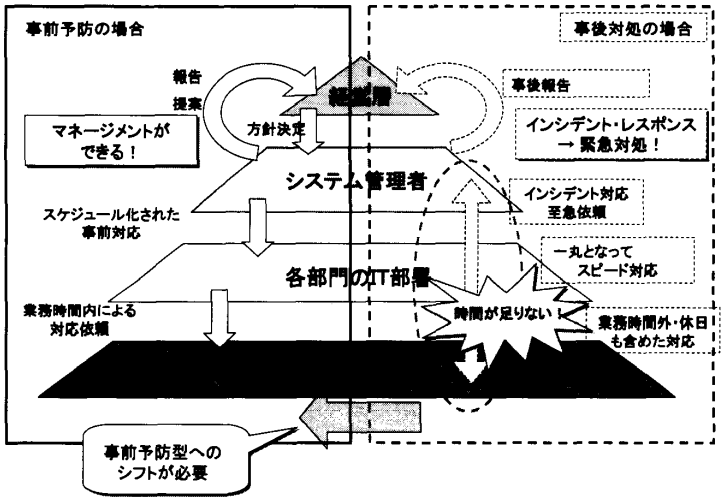

図4　事前予防型へのシフト

ることにより，脆弱性の危険度（V）を算出している．

数値化の試みはまだ始まったばかりであり，定量化の手法にも普遍的なものはない．しかし，このような数値化の仕組みが整えば，それをもとにコンピュータウィルスやワームといったものの危険度を数値化し，定量的な指標で危険性を管理できるようになる．数値化されれば「しきい値管理」や「統計分析」が可能となり，それを基にして自動的に問題を発見・解決する「自動管理」が実現する．「自動管理」は，セキュリティの運用を画期的に変革する可能性を秘めている．

8．継続的な脆弱性の情報収集は，復旧コストを抑える効果もある

継続的な脆弱性管理は，「事前予防」型のセキュリティ運用を実現する．すなわち，ウィルスやワームによる被害が発生してから対策に着手するのではなく，それらの攻撃が起きる前（つまり脆弱性が検知された時点）に，対策に着手することが可能になる．

事前予防型によるセキュリティ対策の場合には，まだ被害が発生しているわけではないので，現場のシステム管理者が問題点を発見した後に，それを，経営層に報告・相談するプロセスを経ることが可能である．その結果，方針決定に従った，「マネージメント」された対処が可能となる（図4）．

これに対して，「事後対処」的なアプローチの場合には，緊急対処的な例外処理となってしまい，経営層への報告が事後承認のかたちとなってしまう．これは，本来，経営層が望む「マネージメント」化された業務とは異なる．もちろん，すべてのセキュリティ対策が事前予防のかたちでできるわけではない．しかし，いかにして事前予防的に対応する割合を増やすかが，セキュリティ対策というものを，本来の通常業務のなかに組み込むことにつながる．そして，結果としてシステムの堅牢性を増すこととなり，また，経費の削減にもつながってくる．

9．今後の課題

脆弱性管理に関する取組みは，まだ始まったばかりで課題も多い．特に危険度の数値化の部分については試行錯誤の段階であると言える．

言うまでもなく，脆弱性の危険度の数値化は，最終的には「リスク」の算定のために用いることを考えた取組みである．リスクについては，例えばGMITS (Guidelines for the Management of IT Security) では，次の式が紹介されている．

リスク＝資産価値×脅威×脆弱性

本稿で述べた脆弱性危険度の数値化の例では，この「脅威」と「脆弱性」の分離があいまいであると言える．本来は「脅威」として述べられるべきものが「脆弱性」の数値化のパラメータとして用いられているからである．

しかし，これには理由がある．一般の利用者が「脆弱性」の危険度といったものを判断する場合，そこには「脅威」の要素が入っている．学術的な分野ではこれを明確に分離しようとしているが，製品を提供しているベンダが，あえて両者をあいまいにして取り扱っているのは，一般の利用者にとっては，その方が分かりやすいからである．

このような遊離をどのように解決するかが，今後の課題となると考えられる．