安心して暮らせる社会構築のためのセキュリティ戦略とドライバーウエアの提案

慶應義塾大学環境情報学部教授　＊武藤佳恭（Yoshiyasu Takefuji）
サイエンスパーク株式会社　小路幸市郎（Koichro Shoji）
三浦秀朗（Hideaki Miura）
川出智幸（Tomoyuki Kawade）
野崎隆（Takashi Nozaki）

論文要旨

ディジタル社会は、人々に便利さと楽しみを与えてきているが、その進歩に伴って解決すべき多くの社会問題も引き起こしてきている。光の部分は勝手に成長していくが、安心して暮らせる社会を構築するためには、影の部分をいかに小さくするかが重要である。セキュリティの現状・対策を把握し、早急に社会全体（政産官学・市民）にしらしめ、セキュリティ・リテラシーを向上させる必要がある。また、ディジタル社会基盤の信頼向上のために、新技術のドライバーウエアを提案する。

〒２５２０８１６
藤沢市遠藤５３２２
慶應義塾大学環境情報学部教授　武藤佳恭
電話：０４６６－４７－５１１１
ファックス：０４６６－４７－５０４１
電子メール：takefuji@sfc.keio.ac.jp

目次

背景

ディジタル社会は、専門家の予想を上回るスピードで成長してきている。ディジタル社会基盤の一つがインターネットであるが、インターネット技術は、そもそも一般市民が使える事を前提に構築されたものではなく、インターネットに起因する問題が生じるのは当たり前の話である。著者の武藤は、“インターネットの遊び方”と題して、インターネットの光の部分を１９９１年の末から１．５年間に渡って共立出版の bit という雑誌に連載し、日本で初めてインターネットとインターネットの利活用を本格的に紹介した。１９８３年夏からフロリダ州タンパの武藤の自宅から、自作設計の３００ｂｐｓ音響モデムを使って南フロリダ大学コンピュータサイエンス学科につなぎながら、インターネットを利活用していた。当時は、武藤の自宅から電子メールを日本の友人に送るためには、アメリカ大陸から太平洋を越えて日本へは富士ゼロックスの人工衛星を経由していた。

そのような和やかなインターネットの時代から、インターネットが米国で社会問題になり始めたのは１９８８年ぐらいからである。きっかけは、コーネル大学の学生が、ウイルスプログラムを構築・配布して、インターネットの一部を致命的に麻痺させたのである。クリントン政権の元で、ディジタル社会基盤の構築に関して多くの政策を打ち出されてきたが、セキュリティ政策に関しては結果として甘いものであった。米国では、社会基盤としてのセキュリティ政策の大転換を迎えたのは２００１年9月１１日以降である。それまでは、セキュリティの中心はファイアーウォールなどの守りの技術であり、完璧と思われるセキュリティを目指していた。セキュリティ政策の大転換の後は、守りのためのセキュリティ政策から次のような戦略的セキュリティ政策に変わった：

１）セキュリティはどうがんばっても完璧ではなくハードウエア、ソフトウエアや人間のオペレーションには必ず欠陥があることを前提にすべての仕組みやシステムを再構築・再管理・再運営すべきである、

２）問題があった場合にはいかにして速やかにリカバーさせるかを考えて構築・管理・運営すべきである、

３）定期的に対処するのではなく、セキュリティホールがあるかどうかイベントドリブンで確かめながら速やかに対処すべきである（ベストプラクティス)、

４）管理者や利用者には必ず悪い人間がいる（または出てくる）事を前提にシステム構築・管理・運営すべきであるなど、性悪説に基づいた政策を次々と打ち出してきた、

５）新たなセキュリティ技術を調査・開発・研究する。

米国のセキュリティの大転換で注目すべき事は、“The National Strategy to Secure Cyberspace”の中で、安心して使えるサイバー空間を目指すには、政府だけの努力や力だけでは駄目であり、政府、地方自治体、民間、一般市民が一丸となって取り組む以外にはセキュリティ問題点を克服できない事を初めて認めながら宣言したことである。“The National Strategy to Secure Cyberspace”は the *National Strategy for Homeland*

Security の重要な一つの構成要素であり、“***National Strategy for the Physical Protection of Critical Infrastructures and Key Assets***”の中で重要なインフラや資産を守るための国家戦略を詳しく述べている。

これらの米国における大政策転換は、予算配分にも多大な影響をもたらしている。たとえば、新参者である Dept. of Homeland security の IT 予算は、米国政府の省庁の中では２番目に大きい規模になってしまった。

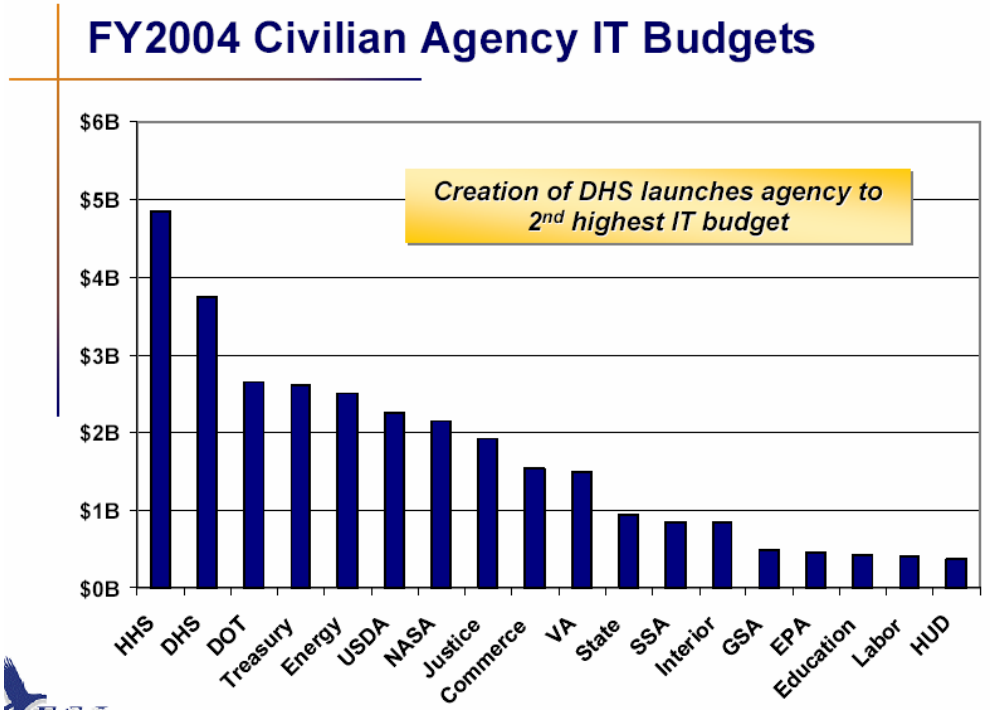

図１　米国各省庁の IT 予算（FSI 提供）

HHS:　Department of Health and Human Services

DHS：Department of Homeland Security

DOT:　Department of Transportation

米国政府がセキュリティだけに費やす２００３年予算は、＄４．７Ｂで５６４０億円に相当する。

図２　セキュリティは米国政府の土台（FSI 提供）

現在の米国の大統領府が描いた、『納税者が満足する政府とその効率化』において、各省庁のパフォーマンスを測定しその改善を図ることは大事なことである。OMB：Office of Management and Business は各省庁のパフォーマンスを測定し、大統領に直接報告する仕組みである。効率と言っても、情報セキュリティは必要不可欠な政府の情報インフラの土台としての認識があり、高いセキュリティなしでは、政府のサービスはありえない。

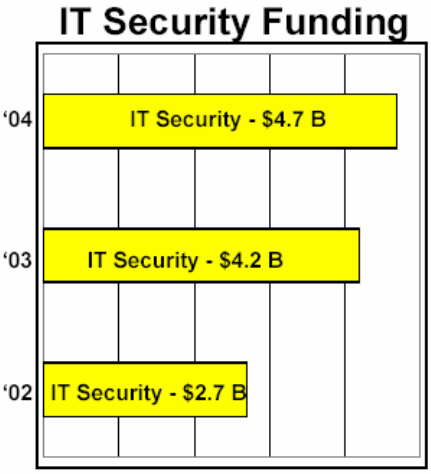

図３　米国政府の IT におけるセキュリティ予算の変化

米国政府の IT におけるセキュリティ予算は、確実に伸びてきている。セキュリティも含めて、省庁のスコアーをつけ急激な改善を図ろうとしている。

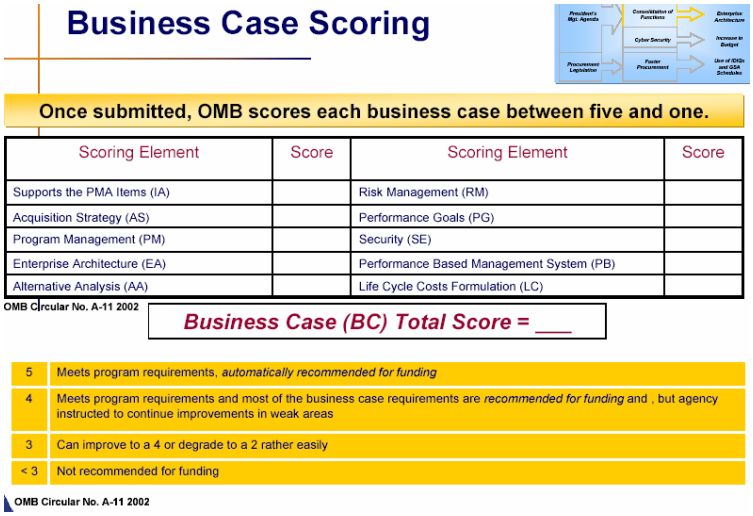

図４　政府のスコアリングの考え方（FSI 提供）

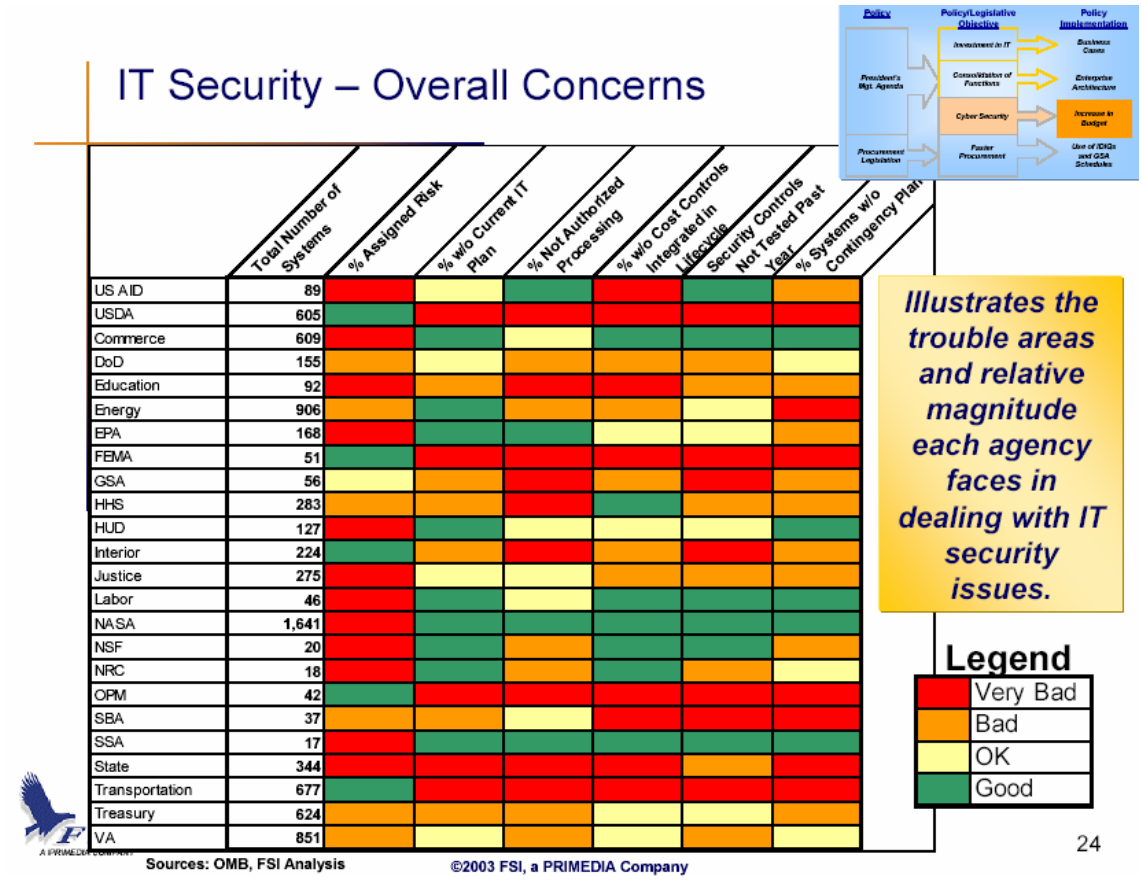

図５　米国政府省庁ごとのセキュリティレーティング（OMB および FSI 提供）

米国政府の最新の省庁ごとのセキュリティレーティングを上に示す。米国では、かなりの努力をしているにもかかわらず、省庁のセキュリティレベルは、いまだにひどい状態であ

る。日本では米国のように積極的にセキュリティを高める努力をしておらず、現在丸裸に近い状態で e-Japan 構想が展開しており、早急な対応が望まれる。

日本のセキュリティをどのように向上させるか？

安心して暮らせる社会を日本で構築するためには、まずセキュリティに関する現状把握が必要であり、問題点を再認識し、その問題点を克服するために国家戦略を立てなくてはいけない。セキュリティに関する現状把握と問題認識がないために、日本での政策はあまりにも ad hoc であり対処療法的であり、また政産官学・一般市民にセキュリティ・リテラシーがないためにインターネットに関連する社会問題が深刻になってきている。

民間と国や地方自治体の役割は異なる。国や地方自治体は、セキュリティを情報基盤と位置づけ一般市民に安全な電子政府を構築しサービス提供すべきである。

MS-SQL ワーム事件を通してみるセキュリティ対策

２００３年1月２５日（土）午後８時３０頃から武藤の自宅に複数の大手新聞社から電話やメールが来た。「韓国全土でネットワークが麻痺しているのだが、原因が何かわかりますか」という同じ問い合わせである。さっそく、SANS 協会（世界最大のセキュリティ教育・研究機関）が運用しているインターネットストームセンター（http://isc.sans.org/）にアクセスしてみると、ポート１４３３が top２位で暴れまくっていた。午後９時１７分の時点で日本の大手新聞社の中では、産経新聞のみが「韓国全土でネットワークが麻痺している」というウェブ報道をしていた。ほぼ同じ時刻に、米国の報道サイト（CNN サイトや USAToday サイト）にアクセスして見ると、「MS-SQL を経由して 2001 年に流行った code red のようなウイルスが暴れまくっている」と詳しく書いてあったので、それを見ながら電話で複数の大手新聞社に伝えた。慶應義塾大学の管理者からのメールには、午後５時にポート UDP/１４３４が攻撃されているのでポート１４３４を閉じましたというメールであった。その後日本の大手新聞社のサイトにも、「'Sapphire'あるいは'MS-SQL Slammer'と呼ばれるワームが MS－SQL を介して、ポート UDP/1434 へ DOS 攻撃し暴れまくった」という報道があった。このワームは１分以内に４千サイト以上に感染している。ワームに感染した１台の MS-SQL サーバーからは、なんと１秒間に５０M ビット以上のトラフィックデータを吐き出す。このように大量のデータをホストコンピュータに送りつけることを DOS 攻撃(denial of service）と呼び、ネットワークにダメージを与え麻痺させる。ちなみに、２００２年１０月２３日には、分散 DOS 攻撃によって、世界に１３台しかないルート DNS サーバーのうち９台が麻痺させられた。その後これらのルート DNS サーバーはセキュリティ強化されたにもかかわらず、MS-SQL ワーム事件では１３台のうち５台が麻痺させられた。ワームとは関係ないが、別のルートサーバの事件では、１９９７年７月に人為的設定ミスにより、世界中のインターネットが４時間停止している。

MS-SQL ワーム事件から学ぶべきもの

まず、日本政府・企業・大学・個人で反省すべきことは、２００２年７月２４日にマイクロソフト社から出されたパッチファイル(セキュリティホールをふさぐためのソフトウエアプログラム)を実行していれば、この事件は起きなかったかもしれない。“かもしれない”という表現を使わなくてはいけないのは、問題が根深いからである。現在わかっている脆弱性あるいはセキュリティホールは６００００種類あり、頻繁に悪用される脆弱性は２００００種類ぐらいである。有名な脆弱性データベースは、米国政府の NIST にある。(http://icat.nist.gov/icat.cfm)

オープンソース

OS やアプリケーション・ソフトウエアのソースプログラムをオープンにして、誰でもがそのソースコードを読める事は、ソフトウエア教育にとってたいへん意義があり大事な事である。しかしながら、ソフトウエアをオープンソースにすることで、セキュリティ問題が直ちに解決することには、残念ながら簡単に結びつかない。何故ならば、セキュリティ問題を解決するためには、セキュリティ問題のある OS の欠陥を発見し、修復しなくてはいけない。もしもその修復自体に問題があれば、セキュリティホールの傷口を広げる事にもなる。セキュリティホールとは、ソフトウエアの欠陥や、バグ、あるいは設定ミスなどであるが、その欠陥が致命傷で深い論理的問題に起因する場合は、その修復は容易な事ではない。OS の名前が変わったり、バージョン番号が大きく変わるのは、深い論理的問題を解決しようとする場合が多い。

日本のベンダー・製造メーカー・SIer が、オープンソースによって簡単にセキュリティ問題を解決できるとは考えにくい。なぜならば、他人が書いた OS の欠陥を発見し修復できる力があれば、もっと良い OS をとっくに開発できたはずである。OS の開発・修復がなかなかうまくいかない原因は、ハードウエア技術のめざましい成長にもよる。問題が修復される前に、新しい問題が次から次に出てきて、OS の技術者の能力を常に超え続けているからである。

オープンソースで注意が必要なことは、多くのローカル版が存在し登場する事である。いろいろなローカル版では、セキュリティ問題もさることながら、問題なく正しく動くデバイスドライバーがなかなかないことである。デバイスドライバーとは、ハードウエアデバイスを動かすためのソフトウエアで、OS に組み込まれる。

オープンソースで一番期待できることは、ソフトウエア教育にとってたいへん役立つ事であり、質の高い OS 技術者を育成できる可能性がある。オープンソースでの大間違いの議論とは、直ちにセキュリティ問題やその他のソフトウエア問題が解決できるという幻想を抱いている所にある。

ソフトウエアがPL法の対象となった場合、問題の責任を回避するためには出来るだけ情報開示することが必要になってくるが、ソフトウエアにおける、最大限の情報開示とはプログラムソースの公開である。今、世の中で望まれていることは、欠陥のないOSを早急に手に入れることである。すなわち、安全なOSを設計構築できる人材育成が早急に望まれている。

オープンソースのセキュリティ

セキュリティに関する間違った事柄が、堂々と日本国の政策・指針の中にある。具体的に指摘すると、例えば、”オープンソースソフトウェアは、信頼性・セキュリティの確保に効果を発揮すると考えられている。”という文書は、経済産業省の公式文書である：
http://www.meti.go.jp/kohosys/press/0003873/0/030331software.htm

そこで、我々は、定量的に比較検討するために、オープンソースソフトウエアとクローズソフトウエア（商用ソフトウエア）とを、脆弱性数と高い危険度の脆弱性数の２点で比較してみた。

ここで用いたツールは、米国政府のＮＩＳＴ（National Institute of Standards and Technology）が運用管理する、世界的に最も信用できる脆弱性メタベース・サーチエンジンである。２００３年５月２７日の時点で、５７３８の脆弱性が登録されている。（http://icat.nist.gov/icat.cfm）

脆弱性のタイプは、次のように分類されている。
１．“Input validation error”，
２．“Access validation error”，
３．“Exceptional condition handling error”，
４．“Environmental error”，
５．“Configuration error”，
６．“Race condition”，
７．“Design error”，
８．“Other”．
また、脆弱性の高い危険度とは１．２．３．のいずれかである。
１．遠隔のアタッカー（ハッカー）がシステムのセキュリティ保護を破る事ができ、ユーザーまたは管理者の権限を奪い取る事ができる場合がある。
２．内部のアタッカー（ハッカー）がシステムの制御を完全に奪い取る事ができる場合がある。
３．ＣＥＲＴ／ＣＣのアドバイザリーミーティングを開くような場合である。

OS では、オープンソースの Linux 及び BSD とマイクロソフト社の WindowsＯＳにおいて、脆弱性数と高い危険度の脆弱性数を比較した。インターネットブラウザーでは Netscape と Opera と MS IE を比較した。データベースでは、Oracle と postgreSQL と mySQL を比較し、ウエブサーバーでは、Apache と IIS を比較した。

表１　オープンソース・クローズソースの脆弱性数

Products	脆弱性数	高危険度の脆弱性数
Linux	523	266
Solaris	237	155
BSD	272	138
Windows	436	152
Microsoft	736	306
GNU	71	31
Java	143	86
Netscape	85	36
Opera	163	91
MS IE	195	98
Oracle	76	41
postgreSQL	15	9
mySQL	24	12
Outlook	49	24
Apache	80	29
IIS	111	39

結果：脆弱性をなくすのが非常に難しい OS では、オープンソースソフトウエアの代表である Linux は信頼性・セキュリティの向上どころか商用に比べてかなり劣っている。一方アプリケーションでは、商用とほぼ同様の信頼性であることがわかる。Apache のようにどの商用製品よりも世界中で使われている場合は、商用製品に負けないくらいかなり質が高いことがわかる。面白い発見は、ある製品の脆弱性の約半分が高い危険度の脆弱性であることである。シェアーが低い製品は、一般に脆弱性が少ないが、潜在的にある脆弱性がただ発見されていないのかもしれない。

脆弱性スキャンは犯罪行為？

大学などの教育現場では、セキュリティに関して、明確にどこからが犯罪行為で、どこまでが許されるのかなど、はっきりしない点が余りにも多く、混乱を引き起こしており重大問題である。セキュリティに関して、是々非々がきちっと定まっておらず、日本はIT先進諸国とはとても言いがたい。

例えば、警視庁では、ping はサイバーテロ攻撃に分類されている。コンピュータ工学で、ping はネットワーク基本技術のポーリング手法であり、その基本手法が犯罪行為とされている。法務省などでは、ウイルスに関する法案を成立させようとしており、セキュリティ研究者が防護技術を向上するために実験に用いるウイルスも犯罪行為とみなされるかもしれない。

法律の現場でも、教育の現場同様、セキュリティに関してははっきりしない事が余りにも多く、混乱と社会問題を引き起こす原因にもなっている。

『脆弱性スキャンは犯罪行為なのかどうか？』に関して、著者の武藤は痛い経験を積んだ。脆弱性スキャンはそもそもファイルの属性や構成要素を見る単なるソフトウエアであり、ファイルのあるなしを判定しているだけである。通常のウエブアクセスと比べても、ファイルフェッチをしていない分、たいへん弱い命令行為を行うソフトウエアであり、問題にすること自体、問題である。

同様に、『ポートスキャンは犯罪行為か？』には、早急に是々非々を示す必要がある。セキュリティの是々非々を決めずに行われている、日本の大学のセキュリティ教育は信じがたい状況であり、問題はますます深刻になりつつある。

政府や地方自治体がすぐにできること

セキュリティに関する不明確な部分を早急に明確化する必要がある。我々が抱える深刻な脆弱性にかかわる問題を列挙すると次の５つである。

１）そもそも一般にわかってない脆弱性あるいはセキュリティホールが存在する。

２）わかっている脆弱性に対して、パッチファイルがないものがある。例えば、ブラウザーのIEは２００３年８月３０日の時点で２１のパッチファイルがない脆弱性がある。

３）パッチがある脆弱性に対して、対処していないコンピュータが数多くある。

４）脆弱性やパッチの意味がわかっていない管理者が多くいる。

５）脆弱性をふさがないことが、インターネット上の多くのシステムに迷惑をかけることがわかっていない管理者が多くいる。

上の５つの問題点を我々は、日本政府・企業・大学・個人のレベルで再認識する必要がある。根本的に、これらの５つの問題を引き起こしているのは、ソフトウエア（OS、アプリケーション・ソフトウエア、デバイスドライバーなど）およびソフトウエア産業が甘やかされてきたことに起因する。ハードウエアはPL法に従わないといけないが、ソフト

ウエアはPL法から除外されてきているため、ソフトウエアのセキュリティホール・バグが多く存在する。一般に、ユーザーや管理者はシステムやソフトウエアは完全で完璧なものと考えがちであり、特に組織（政府・企業・大学）の長にセキュリティの意識はほとんど見られない。今回の問題を含めて、脆弱性検査とその修復(パッチ)をしていれば、これほどひどい被害はなかったであろう。

脆弱性にかかわる５つの問題点は簡単に解決できないため、大事なデータやシステムを保護するため、問題が起こることを前提にシステム構築することが性悪説的設計である。つまり、ウイルスやハッカーに攻撃された場合、ホストコンピュータやネットワーク機器を直ちに復旧させる必要がある。そのためのツールはデータ・ネットワークインテグリティ・ツールと呼ばれ、ウイルスやハッカーにファイルやレジストリーが変更されたりまたは新しく加えられた部分を検知し、侵入される前の元の状態に戻す情報を管理者に提供するソフトウエアである。

最新の米国政府の調査によると、ネットワークダウンの原因の３９％が人為的設定ミスであり、ソフトウエアのバグが３９％、３％がハッカーなどである。これらの事実から、ほとんどの場合、ネットワークダウンでは人為的ミスが主な原因であることを認識すべきである。

セキュリティ技術で一番誤解がある商品は、ファイアーウオールである。ファイアーウオールとは、単純なネットワークフィルターであり、どのパケットを通しどのパケットを通さないかパケット選別するソフトウエアあるいはシステムである。つまり、ファイアーウオールは設定が命であり、ファイアーウオールをいれたら、すべて大丈夫というものではない。ファイアーウオールはネットワークパケットフィルターと呼びなおしたほうがよいかもしれない。

セキュリティ技術で一番大事なことは、ネットワークインフラや情報通信にかかわる多くのソフトウエア（OS やアプリケーション）やネットワークシステムは性善説のもとに構築されてきていることを認識することである。すなわち、現在のほとんどの性善説システムを性悪説システムに再構築しなくてはいけない。例えば、ネットワークでは IPV4 や最近では IPV6 プロトコールがあるが、いずれも性善説に基づいている。性悪説システムでは、ネットワーク上には常に悪意の第三者、すなわち盗聴者がいることを前提に、暗号符号化などの対策が必要であり、世界の常識である。ワイルドなインターネット上に信頼できる通信経路を確立させるシステムが VPN（virtual private network）であるが、そのソフトウエア自身も完璧ではない。無線 LAN 802.11b では、使われている暗号手法 RC4 が簡単に破れることがわかったことにより、優秀なストリーム暗号が現在熱望されている。性善説システムでは、簡単なパスワードで保護しようとするが、性悪説システムでは、パスワードと同時に本人確認のための biometrics（指紋、サイン、虹彩など）などが重要である。性悪説システムでの、何時何処での情報に関しては、GPS などの技術が大変有望である。なぜならば、GPS（global positioning system）は位置情報だけでなく、Universal Time

Coordinated (UTC)など、地球上で時間同期において100ns以下の誤差で時刻同期できるので期待できる。

５つの問題点への具体的対応策

１）そもそも一般にわかってない脆弱性あるいはセキュリティホールが存在する。

セキュアープログラミングと呼ばれる新しい分野がある。完全に、セキュリティホールのないソフトウエアプログラムをいかにして達成するかを研究する分野である。今のところ、完璧にセキュリティホールのないソフトウエアプログラム作法は確立していない。しかしながら、最新の技術でできるだけセキュリティホールの少ないソフトウエアプログラムを作ることは、社会にとって必要不可欠である。また、セキュリティホールを探し出したり発見する技術も大変重要である。社会啓蒙としては、すべての既存のソフトウエア（OS、ネットワークソフトウエア、アプリケーションソフトウエアなど）は不完全な製品であり、それらの製品にはセキュリティホールがあることを認識し、その事実を社会に広く知らしめる必要がある。

２）わかっている脆弱性に対して、パッチファイルがないものがある。

１）で述べたように、完璧にセキュリティホールのないソフトウエアプログラムを構築することは難しいが、セキュリティホールが発見されたとしても、その起因が深い場合には、その修正プログラム（パッチファイル）を作ることは容易でない。パッチファイルを構築することは、セキュリティホールが深い部分に起因する場合、OS を根本的に再構築することに等しい。パッチファイルがない間の、緊急対策と対応を考えておく必要がある。

３）パッチがある脆弱性に対して、対処していないコンピュータが数多く存在する。

脆弱性に対してパッチファイルがあるにもかかわらず、対処していないコンピュータシステムが数多く存在する。パッチファイルを適用することは、インターネットに接続する管理者にとって必要最小限の仕事である。このように、必要最小限の管理作業をしない組織・個人は、インターネットに接続するサービスを提供すべきでない。社会啓蒙としては、脆弱性対処の周知徹底をすべきである。

４）脆弱性やパッチの意味がわかっていない管理者が多く存在する。

必要最小限の管理作業をしない組織・個人があまりにも多いのは、インターネットサービス管理者への教育を怠ってきたからである。ネットワーク・サービス管理者としてのセキュリティ教育が緊急に必要である。

５）脆弱性をふさがないことが、インターネット上の多くのシステムに迷惑をかけることがわかっていない管理者が多くいる。

必要最小限のセキュリティ対策をしないことは、他人・他の組織への迷惑につながることを認識すべきである。管理者の管理ツールとして、最小限備えなければいけないものは、脆弱性検査ツール、不正侵入検知ツール、改ざん検知ツール、データ・ネットワーク復旧ツールなどである。また、これらのツールは、常に最新版にアップデートする必要がある。既存のウイルス対策ソフトウエアはすべて全く無力であった。

提案する新しいセキュリティ技術

セキュリティの専門家の誰もが、一致した次の４つの認識を持っている。（４つの認識に関して、米軍研究所のセキュリティ責任者、Whitehouse の CIO、米軍のセキュリティ責任者、セキュリティ民間企業 CEO などとのプライベートな交流を通して再確認した）

１．脆弱性のない安全な OS は現在存在しないし､安全な OS の完成にはまだ時間がかかる。
２．現在の OS の脆弱性を対処するパッチを、短期間にすべてのシステムに対処するのは不可能である。
３．現在のアンチウイルスソフトウエアは、シグネチャー方式のため、新しいウイルスには対処がむずかしい。
４．現在の IDS はうそのアラームが多く、現場では多くの場合役に立っていない。

これらの４つの認識を踏まえ、著者らはセキュリティ問題を解決する手法を考案した。ここに提案する新しいセキュリティ技術は､OS を変更したり､パッチを当てることなく（セキュリティホールは開いたままで)、防ぐという方法である。提案する手法は、OS にパッチをあてたり変更することなく、防御できる画期的な方法である。現在の OS の一番下の階層がデバイスドライバーである。現在ばらばらのデバイスドライバーを、一つの階層にすると、強靭なセキュリティ層（ドライバーウエア層）ができあがる。ドライバーウエアとその手法を説明するために、脆弱性のタイプを解析してみた。

２００３年８月２９日の時点で、脆弱性のどのタイプがどれくらい悪用されているのか、調査してみた。結果は次のとおりである。NIST の脆弱性メタベースを使って調べると、ルートの権限を奪う脆弱性の総数が１３３９個存在する。

表２　脆弱性タイプと脆弱性数

脆弱性のタイプ	脆弱性数	ルートの権限を奪う脆弱性数
buffer overflow	1269	520
format string	153	70
metacharacter	155	58
race condition	145	52
boundary condition error	254	32
directory traversal (..)	309	17
Cross-site scripting (XSS)	190	14
unknown	74	7

一番頻繁にワームやハッカーに使われる手法は、buffer overflow の手口であり、脆弱性数が１２６９個で、ルートの権限を奪う脆弱性数が５２０個と圧倒的に多い。２００３年1月のＳｌａｍｍｅｒや8月に蔓延したＢlaster なども同様な buffer overflow を利用して、システム障害を引き起こしている。２番目に多いのが format string error の手口でルートの権限を奪う脆弱性数が７０個､次が metacharacter を突いた手法で５８個、４番目が race condition をついた手法で５２個、５番目は boundary condition error で３２個､6番目はdirectory traversal と呼ばれるdot の手口で１７個､7番目はcross-site scripting で１４個、わかっていない手口が７個である。

一番大事なことは、このような手口を許さないことであるが、著者らは最も頻繁な脆弱性の手口である、buffer overflow に関して現在研究している。その概要を簡単に説明する。

buffer overflow を利用した不正コードの実行は、プログラムが実行されるときに使用されるスタックと呼ばれる“プログラマーがあまり意識しない“部分の必然的な振る舞いを利用している。プログラムが実行される時、プログラムを効率よく実行させるために用意されたサブルーチン関数群が呼び出される。これらの関数はプログラムから渡して使用するパラメータ（argument）とその関数内で使用するパラメータ（local variable)、および、呼び出し元へ戻る位置を保持した return address（instruction pointer）を伴って実行される。通常、これらの関数の実行が終了すると return address に示された呼び出し元のメモリ空間内の実行位置に戻され実行が継続される。しかし、この呼び出された関数内のプログラムの作り方に不具合があると、return address を local variable の確保領域を意図するしないに関わらず越えて上書きすることが可能である。通常 return address のコードの書き換えは意図しないコードの実行となることから、buffer overflow を起したプログラムは通常プログラムの不定な実行状態となり、場合によってはプログラムが暴走、あるいは停止することが殆どである。しかしながら、return address を意図的に自分が用

意したコードに戻すことが可能であるならば、意図された不正なコードが不正なこととはわからずに実行されてしまう。Slammer に代表されるバッファオーバーフローによる脆弱性を突いた攻撃はこの方法を使用し、自分が意図するコードを実行させている。

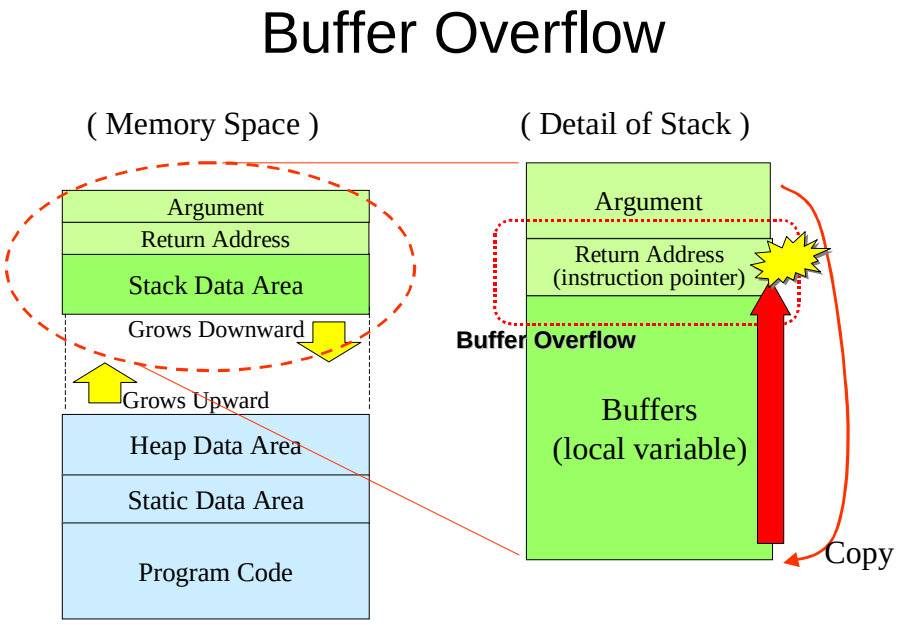

図６　Buffer overflow による return address の破壊

このような buffer overflow による不正コードの実行は、プログラムの実行を管理しているＯＳは全くわらかない状態である。この不正コードの実行を防ぐには、reteurn address の改ざんを如何に防ぐかあるいは検知するかが非常に重要な課題となる。近年、この問題の解決方法に対して、ＯＳ自身に修正を加える方法（Openwall Linux kernel patch project）やコンパイラ自身に buffer overflow を防ぐような仕組み（GCC extensions, libsafe, propolice, stackguard, libmib, MS .net compiler）を提供しているが、いずれも該当するプログラムコードの変更およびリビルドが必要となる。つまり、対策が施されるまでに時間が掛かることと、そして、buffer overflow という汎用的な手法にも関わらず未知の攻撃に対しては何の効果もないことが問題になっている。

我々は、実行時におけるメモリスタック内の return address への改ざん防止と検知を行なう、ドライバーウェア（Driverware）という、新しい技術概念を利用してシステムを開発している。

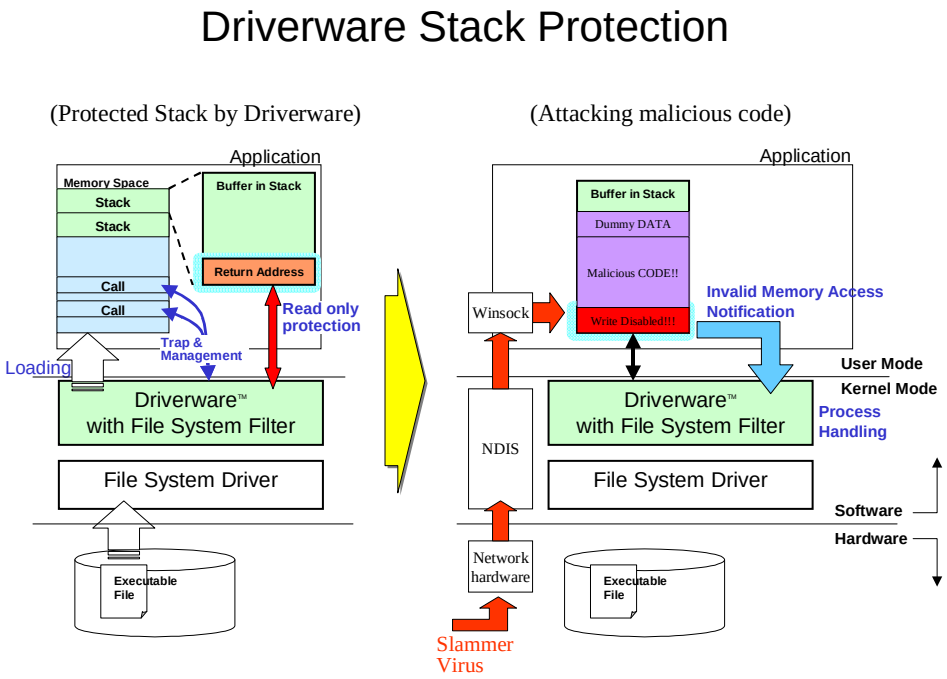

図７　Driverware によるスタック保護・検出機能の概略図

Driverware Stack Protection システムは、既存のハードウェア、ＯＳ、アプリケーション・ソフトウェアを全く変更することなく利用できる画期的な不正コード防止システムである。アプリケーション・ソフトウェアに buffer overflow の脆弱性を持っていたとしても、Driverware Stak Protection システムが、プログラムの起動時に不正コード実行を検出のための仕掛けを行い、不正コードが実行される前に検出し、不正なコードの実行を完全に抑止することが可能となる。

提案する手法は、従来のシグネチャ方式ではなく、プロセス指向の方式である。要約すると、提案する手法は、ＯＳを変更したり、バッチを当てていなくても、ＯＳやシステムを守ることができる画期的手法である。今後、提案手法を実装してから具体的評価をする予定である。

本論文を結論すると、安心して暮らせるサイバー社会を構築するには、セキュリティの現状を把握しその対策を再認識して早急に対応することが重要である。完全なハードウエア、ソフトウエア、運営管理はありえないことを考慮して、システムを再構築すべきである。性悪説に基づく思想でサイバー社会を構築すべきである。

参考文献

１．“The National Strategy to Secure Cyberspace”、米国政府
２．“*National Strategy for the Physical Protection of Critical Infrastructures and*

Key Assets”、米国政府
３．ＮＩＳＴの脆弱性メタベース・サーチエンジン（http://icat.nist.gov/icat.cfm）
４．http://www.openwall.com/linux/
５．　http://www.openbsd.org/33.html
６．http://www.gnu.org/software/gcc/extensions.html
７．http://www.research.avayalabs.com/project/libsafe/
８．http://www.trl.ibm.com/projects/security/ssp/
９．　http://www.immunix.org/stackguard.html
１０．http://www.mibsoftware.com/libmib/astring/
１１．
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vccore/html/vclrfGSBufferSecurity.asp