InternetWeek97 チュートリアル

インターネットセキュリティ(1)

１９９７年１２月１６日（火）１０：００－１７：００

岡田　高(ＪＰＣＥＲＴ／ＣＣ)

目的

最近は国内でもクラッキングの被害が多数報告されていますし、Web ブラウザのセキュリティホール、SPAM/E-Mail 爆弾など、課題は増加する一方です。このため、このチュートリアルでは、こうした問題を広く取り上げて、今知っておくべきセキュリティ対策についての基本的知識を解説します。

1. JPCERT／CCについて

2. 不正アクセスの現状

3. セキュリティの考え方

4. サービス利用者のセキュリティ

5. 一般的なサービスのセキュリティ

6. 各種サービスのセキュリティ

7. 非常事態への対応

8. まとめ

9. 参考情報

1. JPCERT／CCについて

JPCERT／CCは、日本国内を対象とした不正アクセス情報に対応する緊急対応組織、IRT（Incident Response Team）と呼んでいます。インターネット全体のセキュリティの向上を目的として活動しておりますが、特に日本国内を対象として日本語でサービスをおこなっているのが特徴です。活動内容は、大きくわけて「不正アクセスの予防」と「事後対応」の二つの立場からのものがあります。基本的には皆様が万が一不正アクセスを受けられた場合に、どのような不正アクセスを受けたかを届けていただいています。その届け出を多くの所からいただくと、どのような不正アクセスが実際おこなわれているかという傾向がつかめますから、それを把握した上で、もしも大規模な同じような種類のアタックがおこなわれていた場合には、警告するというようなことをおこなっています。その結果、昨年の10月より1年あまり活動していますが、その間その種の警告文書のようなものは4件ほど発行しています。

また、緊急的な警告とは別に電子メールの不正な中継に関する技術メモという形で技術文書も発行しています。また、個々に届け出いただいた方に対しても極々一般的なアドバイスですが不正アクセスの再発防止に関する簡単なアドバイスをさしあげています。

また、自分のところが踏み台に使われてしまったとか、自分のところに攻撃してくるところも踏み台に使われているようだといった場合、直接相手に連絡しずらい事情があるかもしれません。そのようなときには、JPCERT／CCが代わって相手のサイトに対し「あなたのサイトがもしかしたら不正アクセスをうけているかもしれないので確認をしてください」と連絡することも行っています。

JPCERT／CCは民間の中立的な立場から非営利で活動しているので、残念ながら不正アクセスをした者を懲らしめてくれというような要望にはお応えできません。あくまで技術的な面で今後の対策を普及させていく組織ですのでご了承ください。

JPCERT／CCへのアクセス方法を以下に示します。

- WWW
  http://www.jpcert.or.jp/
- 情報提供用メーリングリスト （登録方法）
  http://www.jpcert.or.jp/announce.html
- 電子メール: info@jpcert.or.jp
- 電話:　　03（5575）7762
- FAX:　　03（5575）7764

2. 不正アクセスの現状

・JPCERT／CCへの届け出

1996年10月～1997年9月の間に400件弱の攻撃の届け出を受けています。この数のなかには、未遂の攻撃とか、不可解なパケットをフィルタした記録が残っているとかというような届け出も含まれています。また、攻撃が実際あったにも拘わらずJPCERT／CCへの届け出をしなかったものも存在すると思われ、この数が即日本国内の不正アクセスの総数の目安にはなり難いとは思われます。ただ、この数字から日本国内のサイトが実際に攻撃を受けているということは言えると思います。なかにはかなり深刻な被害を受けておられるところもあります。個々の具体的な不正アクセスの実態については、JPCERT／CCのWebページに3ヶ月ごとに活動概要を、年に1度活動報告を載せておりますので、詳しくはそちらを参照ください。

・不正アクセスの動向

JPCERT／CCに届け出られた不正アクセスの動向は以下のとおりです。

- 特定のソフトウェアのセキュリティ・ホールへの攻撃
  ちょうど1年前の sendmail への攻撃、1年弱前の INN(Internet News)に関する不正アクセス、最近かなり多くなってきた phf(Web サーバの httpd のサンプルプログラムとして付いてくる CGI プログラム)を利用した攻撃、最近国内外を問わず続いているのが古いIMAPサーバに対する攻撃
- 電子メールの不正な中継と電子メール爆撃
  SPAMの中継サイトとしてつかわれているという例が継続的に寄せられています
- パスワード推測， パスワード破りに関する報告もあります
- 原因不明の侵入

・侵入後の行動

侵入者は一般的に次のようなことを順におこないます。

①root 権限の不正入手

まず、何らかのの方法で一般ユーザのアカウントでシステムに侵入する。大抵のシステムには一般ユーザが何らかの操作をすることによって不正にスーパーユーザ(root)になれるセキュリティホールがあるので、それを使って root になる

② トロイの木馬の設置

トロイの木馬といわれるプログラムをインストールする。システムの裏口や、各種のシステム情報の収集、自分の行動を管理者から見えないように隠蔽(例えばUNIXでは、ps, netstat 等のシステムモニタリングコマンドの置き換え版を作って自分の行動を隠す)といったものが使われている。

③ パケット盗聴プログラムの設置

イーサネットのようなネットワークでは、電気的にはあらゆる通信がすべてのノードに届いている。システムの設定を変更することにより他のマシンの間で流れている通信を傍受できるようになる。多くのケースでは、その中から認証情報(ユーザ名とパスワード)を集め、あとで他のシステムに侵入するときにそれを使う

④ 踏み台アタック

侵入ホストを使ってさらに他のサイトのセキュリティホールを探してそこに侵入する

・スキャン型の攻撃

特定のセキュリティホール（例えば、phf とかIMAPサーバ）への攻撃を多数のホストに対して機械的に試行するもので、最近よくおこなわれている不正アクセスの形態です。この攻撃の特徴は、特定のサイトへの攻撃が目的ではなく、単にたくさんのサイトを攻撃すればどこかのサイトを攻撃できるだろうという方向でやっていることです。その結果、サイトの事情（あるサイトが有名か無名かなど）に関係なくインターネットに接続していると、一定量のリスクを負わされるということになってきています。つまりインターネットに接続するからには、どんなサイトであっても、なんらかの守りが必要になってきていると思います。この種の攻撃でよくおこなわれているものは、UNIXの/etc/passwd を盗み出す攻撃です。/etc/passwd にあるパスワードは暗号化されていますが、これは破れないわけではありません。総当たりで、あるいは辞書に載っている単語と照合していくことによって破るツールは世の中に出回っています。このようなファイルが盗まれると、ツールを使って侵入される危険が増すこととなります。

・サービス妨害攻撃

英語では、Denial Of Service Attack といわれている攻撃で、外からデータを送り付けるとシステムが落ちてしまう、あるいはシステムが本来発揮するはずの性能で運用できなくなる、という種類の攻撃です。この種類の攻撃は、現在流行っているのではないかと考えています。ですが、検出することが難しいからなのか、なぜかJPCERTにはそういう報告があまり寄せられてこないのが現状です。基本的にこの型の攻撃はかなり手軽に試すことができるため、今後増えてくるのではないかと思われます。なかには本質的な対策が困難なものもありますし、逆に簡単な対策で防げるものもあります。こういった種類の攻撃も存在しますので、セキュリティ関係の情報には注意していただきたいと思います。

3. セキュリティの考え方

・セキュリティへのステップ

実際にサイトを安全な状態に保つためにどのようなことに留意しなければならないかについて説明します。セキュリティに関しても通常のシステム構築と同様に次のステップを踏みます。

- 条件の明確化
- ポリシーの決定
- 設計
- 実装、構築
- 検証
- 運用、評価
- 繰り返し

最初に重要になってくるのは、諸々の条件を最初によく検討しておくことです。特に、どういった人がシステムを使うのか、どういった人がシステムを運用するのか。どういうものがシステムの内部に存在し、今後どういう器材を追加してゆくのか。また、そのために投入できる予算はいかほどか、あるいはいかほどの時間をシステムのメンテナンスに割いてゆけるのか。といった、諸々の条件を最初によく考えておくことが重要であると思います。

次に、前述の条件に基づいて、どういう方針でシステムにセキュリティを導入してゆくのかということを決めておく必要があります。セキュリティというのは、守りを固めてゆく段階で利便性とか操作の簡単さを犠牲にせざるをえない面があります。また、便利性の向上とか業務の効率化を図ろうといった、システムの本来の導入目的を阻害するほどセキュリティに注力しても本末転倒なわけです。したがって、どれくらいのバランスでセキュリティを確保するのかといったことを考えておく必要があります。また、セキュリティに関しても、どういうセキュリティを大事に考えてゆくのか、どういうセキュリティを必ず守らねばならないのかをまず考えておかねばなりません。

例えば、不正アクセス対策を第一に考えるのか、あるいはとにかく提供されているサービスがそのまま動き続けることが大事だと考えるのかによって、運用方法に違いが生じます。つまり、とにかくすぐにセキュリティパッチを当てようという方針を選択するのか、あるいは、先にテストシステムでテストしてからでないと実運用システムにはパッチを当てないことにしようという方針になるのかという違いがでてきます。そういう意味で、最初にセキュリティの優先順位をつけておくことが大切になります。

方針をまず最初に考えてから、セキュリティを実現するシステムをデザインして、その後で実際に構築してゆくわけですが、理想どおりに事は運ばないのが常で、一進一退しながら少しづつ最適なセキュリティ環境に近づいてゆくことになろうかと思われます。

・セキュリティポリシー

セキュリティポリシーの話題については、どのセキュリティの教科書にもある程度の量を取って触れていますが、現場のエンジニアが私はこうするからこうだといって決めていいものではありませんので、このチュートリアルでは簡単に触れておくだけにします。

組織的にセキュリティポリシーに関する公式文書をドキュメント化しておいて、それに従ってみんなが行動する。しかも新しくセキュリティ関係の事情が変化したらタイムリーにその文

書をアップデイトして常に最新の状態になっている。……というのが本来は理想です。しかし、現実の組織ではなかなか理想どおりにはゆかないのではないかと思います。特に、一度公式文書にしてしまうとなかなか更新ができなくて、かえって困るというところもあります。また、明確なポリシーがなじまないような組織というのもあるかと思われます。

それではそういう場合どうすればよいかといいますと、とりあえず自分がどういう方針で活動してゆくのか、あらかじめよく考えておかれるのがよいのではないかと思われます。たとえ非公式なドキュメントであったとしても、ドキュメントにまとまっていれば、例えばシステム管理を後任に引き継ぐ場合とか新人のシステム管理者を育てる場合など、かなり有効なドキュメントにはなるのではないかと思います。また、セッティングをしているうちに何のためにやっていたのかわからないほどつい夢中になってしまった場合に、自分の決めたセキュリティの方針に従って反省、見直すのも必要ではないかと思います。

・構成要素の選定

ここでは具体的にセキュリティに関する実装について説明します。いかなる構成要素、具体的にいえばいかなるＯＳを使うのかあるいはいかなるサーバソフトを使うのかというようなことが気になることだとは思います。

このときに最も留意する必要があるのは次の点だと思います。すなわち、どれだけ安全にシステムを維持できるかは、システム管理者がどのくらいシステムに慣れているか、どれだけ労力、時間をそのシステムに注ぎ込めるかに掛かっているのではないかと思います。たとえば、大抵のシステムでは、ある程度機能性重視の傾向があり、デフォルトでもなんでも動いているということがよくあります。そういった場合、例えば使わない機能を無効にしたり、これはいらないから殺したり、これは必要だから残したりといった選択をしていかないと、安全にはなかなかなりません。そういったことから、どんなシステムについても、システムへの慣れとシステムに専念する労力の量が、安全性にかなり影響します。

システムの構成要素の選定については、実運用にはいる前にどれだけ管理者が慣れているかということが大きくセキュリティに影響します。余裕があり、外からは切り離した内側のプライベートなところでテストをやって、慣れてから実運用に入るというようにできればある程度解決することではあります。ずっと使っているシステムですと、情報を継続的に仕入れる癖も付いていますし、昔からの蓄積でセッティングにも慣れていますから間違いも少なくなることでしょうから、やはり慣れたシステムの方が安全かと思います。

そういうことで、まずは自分がどれぐらい操作（セッティング）に慣れているかという基準を重視していただきたいと思います。

・フリーソフトウェア

構成要素の選定ということでは、もうひとつ、フリーソフトウェアについてとりあげておきます。フリーソフトウェアは危ないと思うか、あるいはフリーソフトウェアだから安全だと思うか、という問題があります。

実際のところは、もともとインターネット自体がＢＳＤ文化あるいはフリーソフトウェア文化で育ったということもあります。また商業製品でも結局フリーソフトウェアに独自の変更を加えたようなものも結構あります。そこで、私としては、「そのものの」安全性にはそう大差ないのではないかというような感触を持っています。もちろん一般論ですから、あまりにも実験的す

ぎて安全性があまいソフトウェアがあるかもしれません。これについて断言するのは難しい面もあります。

では、商業製品を買う価値はないのかというとそうではありません。商業性品では、セキュリティパッチというものがコンスタントに提供されていて、そのままそれをインストールするだけでセキュリティホールを潰せるというメリットがあります。一方で、フリーソフトウェアは穴が見付かったらすぐパッチが開発されますから、フリーソフトウェアを自分でコンパイルする人であればどんどんと自分で穴を塞いでゆけます。ですから、そういう労力をかければ、より安全にできるチャンスはあります。

OS に含まれるベンダー提供のソフトウェアを、フリーのものに置き換えて更新するということも有効な不正アクセス対策として行われています。しかし、前述のように、ベンダーが独自の変更（拡張）を施している場合、本当に置き換えて大丈夫かというようなことが気になります。不正アクセス対策よりもシステムの安定性を重視しているシステムでは、そういうセッティングに踏み切れないケースもあると思います。

結局、何が目安になるかといいますと、どれぐらいそのシステムに管理コストをかけられるかという一点に尽きるのではないかと思います。例えば、現金があるという環境ですと、買った方が人間の数が減って楽だというようなことになるかもしれませんし、資金はないが若い学生が管理をやっていてがんばってくれる環境ですと、じゃおまえがんばれといった方が安全になるのかもしれません。フリーソフトウェアはソースがあるという言い方をよくしますが、ソースプログラムを読んで自分でそこそこの対応ができるという手間暇、時間、実力があればもちろんその方が安全にできるチャンスはあります。逆に自分は make install だけしかできないような状況で、フリーソフトウェアのほうが、ソースがあるから安全だという考えでフリーソフトウェアを導入するのもどうかなとは思います。そのあたりをよく考えて自分のシステムに安全なコンポーネントを選んでいただきたいと思います。

・セキュリティの要素技術

いろいろなコンポーネントを組み合わせてセキュリティを実現していくわけですが、次に論理的セキュリティの要素技術について述べます。多くの教科書を見てみると次の分類で定着しているようです。

●認証
●アクセス制御
●監査記録、ログ
●データ保全(data integrity)
●データ秘匿(data confidentiality)
●否認防止(nonrepudiation)

「認証」というのは、お互いに相手がどういうシステムかあるいはどういう人間かというのを明らかにする手段のことです。インターネットの場合、外部からの不正アクセス対策ということになりますとホストの認証、IPアドレスの認証あるいはホストの名前の認証といったような機械単位の認証の話とある機械のあるユーザあるいはどの機械の上でもいいからこのユーザという形のユーザ単位の認証という二つの話題があります。次に、認証結果に基づいてアクセス制御をしたりとかログをとったりとかデータ保全、データ秘匿、データ保護をおこなったりするわけです。

「アクセス制御」というのは、インターネットの場合ですとホスト単位の制御をやることもありますし、このホストのこのユーザだけという制御をやることもありますが、何らかの制御を加えていくことによってセキュリティを実現しようとするわけです。

「監査記録」や「ログ」は、システムにアクセスした記録やシステムの運用記録を保存しておくことです。もしも不正がおこなわれていた場合には後から見直すとかあるいは不正の兆候を事前にキャッチするというようなことをおこないます。

以下は不正アクセス対策やインターネットセキュリティとは少し離れますが簡単に触れておきます。「データ保全」というのは、データの消失を防ぐということです。たとえば、RAID ディスクのような冗長なディスクを用意したり、あるいはバックアップを取って後でリストアするというようなことでデータの消失を防ぐということです。「データ秘匿」というのは、アクセス制御とも関係しますが、見られたくない人にデータを見せないということです。どちらかといえば暗号を使った話をさすことが多いようです。「否認防止」というのは、「あなたこれ言ったでしょう」というのを言い逃れできないようにするための技術で、これも暗号技術を基礎とするディジタル署名などを使って実現してゆくものです。

このなかでは、不正アクセス対策という意味では、認証、アクセス制御、および監査記録・ログというのが大事になってきます。以下これらの事項について詳しく説明します。

・ユーザー認証

インターネットに直結している機械でも、遠隔ログインサービスが稼動していて、遠くからユーザ名とパスワードを打ち込んでその機械にログインできるようになっているものが多いのではないかと思います。そういうシステムでは最初のユーザ認証のところで破られないようにするというのが非常に重要になってくるわけです。

多くのサイトでこの認証に使っているのは何といっても「パスワード」ではないかと思いますが、これを如何に安全に使うかということが大事なことだと思います。パスワード認証自体

は非常に安くできるという魅力がありますので、今後なくなってしまうということはないでしょうが、インターネットに直結して直接リモートログインができるような環境ですと、非常に注意を払っていただかないと危ないのではないかというような昨今の状勢になってきています。

どうして危ないかということですが、まず、ユーザにこういうパスワードをつけると危ないというようなことをいろいろ教え込んだとしても、やはり大抵のユーザは結局危ないパスワードを付けてしまうということがあります。厳密な統計の数字ではないですが、システム管理者の間でよく交わされる会話で、「この間うちのサイトでパスワードクラックしたら何%破れてしまった」、「いや、それくらいならまだましで普通これくらいはいくよ」というのがあります。いくら注意しても、何人かは危ないパスワードをつけてしまうのです。

よしんば辞書に載っていないようなあるいは何らの特定のパターンに合致しないようなパスワードであったとしても、それが短いパスワードあるいはアルファベットだけのパスワードだったりすると、最近の計算機パワーはそれを総当たりで破ってしまうところまできているそうです。パスワードの最大長が8文字だとすると、8文字一杯でしかもアルファベット、大小文字、数字、記号をみんな使って、さらにその上定期的に変更し、しかも同じものを再利用しないといったように注意を払っていかないと、なかなか安全に運用することは難しいのではないかと思います。

ただ、危険なパスワードが使われている場合でも、それが今すぐ危ないかというとそういうわけではなくて、危ないのは、例えば何らかのセキュリティホールを使ってパスワードファイルが盗まれた場合です（単純に危ないパスワードが付いているからといって、直接接続して推測されるケースは少ないと思います）。もしもシステムが8文字よりも長いパスワードを採用しているのであれば、なるべく8文字よりも長いパスワードを使うに越したことはないと思います。

本質的な解決策として、もう少しほかの認証機構を導入すればいいのではないかというのがあります。最近かなり定着してきた「ワンタイムパスワード」がそれです。

少し認証するときの手間が増えてしまいますが、最近はかなり便利なソフトウェアもありますし、難しい部分をハードウェア化した製品も売っています。この場合、あらゆるユーザ認証をこのワンタイムパスワードにしてしまうとかえって不便だということでみんな使わなくなってしまったり、いい加減なパスフレーズの入力といった誤った使用法だとワンタイムパスワードと言えども危ないかもしれないということにもなり兼ねませんので、どういうところに適用してゆくかというのを判断するのが重要なのではないでしょうか。たとえば、インターネット経由でログインするときはワンタイムパスワードを使い、安全なＬＡＮ上では普通に8文字パスワードを使うというのも一つのやり方です。もちろんワンタイムパスワードは、ネットワーク盗聴対策にも有効で、ネットワークが信用できないと思えばワンタイムパスワードは必須になってきます。

認証技術としては他にもいろいろあります。例えば、最近フリーで流行っているＳＳＨ（secure shell）はrlogin，rshの替りに使うようなソフトウェアですが、ＲＳＡ暗号を使った認証が使えるようです。あるいはバイオメトリックス（生態計測）といった、我々のいままでの生活には馴染まない概念かと思いますが、そういったものも出てきています。そういったものがもっと手軽に利用できるようになってきますと、もっと安全なユーザ認証を構築できるようになると思われます。

先ほどからバランスの話を少しづつしてきていますが、程度の低い認証システムから高度なものへ並べてみると、認証なしというのは論外として、パスワード認証よりはワンタイムパスワードのほうが（正しく使えば）もちろん安全です。さらに凝った認証システムを導入すれば、認証を正規でないユーザが通過することはさらに難しくなると思います。ワンタイムパスワードにしますとコストは増えることは事実ですから、そのあたりのバランスを考え、もう一つは自分のところのネットワーク環境を判断した上で適切な場所に必要に応じて強力な認証手段を導入する運用を考えてみていただきたいと思います。

・アクセス制御

アクセス制御には、特定のユーザあるいは特定のホストからのアクセスを許す、あるいは特定のユーザあるいは特定のホストからのアクセスを拒否するというのが考えられます。

今のインターネットで使える方法には、よくあるのが「ファイアウォール」で一括して落す、ルータやファイアウォールの「パケットフィルタリング」で通信そのものを何らかの方法で遮断する、あるいはそういうのを持っていないところでは、ホスト単位のサーバの付加機能、「tcp_wrappers」みたいに本物のサーバの前段階で呼び出されるようもので対応するあるいは「xinetd」というような inetd の入れ替えのツールで対応するなど、いろいろ方法はあります。

基本的なアクセス制御の考え方として、もちろんアクセス制御をしないでみんな許可というのもありますが、普通こういう場合に紹介される二つの考え方としては、許可するもの以外は全部拒否というのと、拒否するもの以外は許可というのがあると思います。

どちらがより安全かといいますと、管理者が把握している範囲内でこれとこれはちゃんと面倒を見られそうだからいいだろう、他は全部ダメという、許可する以外拒否というポリシーのほうがより安全な方法論になるかと思います。しかし、これも実際の運用によっては、本当にこれでうまくいくのかというとなかなかそうはいかないかもしれません。というのは、システムを使っているうちに新しい要求というのが出てきます。例えばシステムを立ち上げた段階で、これとこれは許すこれは許さないあるいは他は全部許さないという設定をしていたとしましょう、しばらくすると「ねーねー、ここでは何とかかんとかはできないんですか」とかいう声がユーザから挙がってきたりするわけです。そういう時に、セキュリティポリシーで確固としてダメだという体制になっているところだと断りきれるわけですが、その辺が組織的に曖昧なままなんとなく管理を引き受けている状況ですと、なかなかユーザの声にいやとは言えません。そうすると、許可するもの以外は拒否というポリシーでやっていたにも拘わらず、だんだん自分の把握しきれていないものに対して穴を開けていかざるをえないことになっていきます。その結果、いつの間にかシステムがよく解らない状態になっていたということになってしまいます。

拒否するもの以外は許可という方法ですと、例えば、この間あれが危ないと言われていたのであれを止めようかというやり方になってきます。これもこれで、結構安全性を保つという意味からすると、何かあるたびに対策を施さないといけないという意味で辛くなってくるのではないかと思います。

例えば妥協案の一つとしては、無条件で許可するものは許可するが、そうでないものはあとからログを取って後から調査できるようにしておくというのも一つの方法かもしれません。

インターネットに対して防御をとる場合は、基本的には何らかのアクセス制御はかけるようにしないとなかなか安全にはならないと思います。特に、自分の組織のIPアドレスが始点のアドレスにあるパケットが外から来たら、それはフィルタリングをして落さないと危ないとか、そういう極々当たり前のフィルタリングというものがいくつかあります。そういうものは必ずアクセス制御すればかなりの改善になります。ですから、アクセス制御というものは不便になることだからいやだという一言で片づけずに、自分の必要とするセキュリティレベルあるいは自分の必要とする利便性にあわせて、どの程度のアクセス制御を導入するかを考えていただきたいと思います。

・ログ

ログは、不正アクセス対策にとどまらず障害の監視、稼動状況のモニタリング、システムのパフォーマンスモニタリングなどの目的で記録するわけですが、集めれば集めるほど、後で管理、整理して解析すれば役立つわけです。しかし、ログの場合、そこにはユーザのプライバシー情報も格納されることになるので、そういう情報の取り扱い方が難しくなってきます。組織的にセキュリティポリシーが定まっていて、システム管理者がある種類のログを取り解析結果を出しても構わないとか、それをこういう範囲なら開示しても構わないが、こういう範囲しか開示してはいけないなどといったルールが決まっていれば何の問題もないわけですが、なかなか実際そこまできっちりと定まっているところは少ないのではないかと思います。そういう場合、ユーザに内緒でこっそりログを取るのというのも一つの方法かとは思います。しかし、できればあらかじめユーザに、このシステムはセキュリティ対策として、こういう監視をしているが構わないかという確認を取った上で、円満に運用していただくというのがシステム管理者の自己防衛上からも、特にルールのない組織では安全なのではないかと思いますがいかがでしょうか。

ログを多く取れば取るほど、後で見直したときに有効な情報が残っている可能性は増えます。そういった意味では、例えばＵＮＩＸ系のホストでは syslogd というのがあって、syslog.conf を書き直すとログの量を増やしたり減らしたりできますが、そういうのをサイトの事情にあわせて設定をして、デフォルトのログよりも多めに取るよう対応をしてくだされればと思います。基本的に、インターネットを経由した通信・アクセスに関しては、できるだけすべてのログを取るというのがある意味で理想かと思います。そういうログが確実に取れていると、万が一不正アクセスを受けたのではないかといった場合にも、攻撃元サイトの特定や被害状況の把握に役に立ったりすることになりますから、基本的にはたくさんログを取るようにした方がよいのではないかと思います。ただ、ログを多く取ろうとするとよく犯すのがディスクを溢れさせるというミスです。これも少し調節が難しいですが、よく注意の上ログをたくさん取ることをお薦めしておきたいと思います。

・ファイアウォール

ファイアウォールという名称は常々お聞き及びと思います。もう既に導入されているところもあるかと思います。ファイアウォールとは、、セキュリティの管理レベルの違うものの間に一台もしくは何台かの機械を組み合わせておいて、そこでそういう違うセキュリティの管理レベルをもつ場所のあいだの中継を一括して管理して、関所みたいな形でセキュリティ管理をやっていこうというものです。一般的によくあるのは、インターネットと自分の組織の間に設置されるのが多いですが、もちろん自分の組織の内部で、あそこは比較的セキュリティよりももっと他のものを重視している、こっちはセキュリティが重要だといった場合には、その間に置いたりする、というやり方も考えていいのではないかと思います。

ここで大事なのは、間にファイアウォールを一個置くことは、そこにセキュリティ管理を集中させ、対策コストを下げるという役割を持っているということです。そういう本来の考え方とは別に最近一部ファイアウォールの売られ方をみていますと、ファイアウォールがあるから安全とか、とりあえずファイアウォールを置いておけば安全ですという売られ方をすることがままあるの聞いております。それではファイアウォール1台置いておけば本当に安全なのかと言いますと、必ずしもそうではないでしょう。実際にはファイアウォールの種類、実装あるいは仕組みにも依りますが、防御できない不正アクセスも世の中にはある程度ありうるという事

実はやはりなくなりません。ファイアウォール自体になんらかのセキュリティホールがあって、それをそのままアップデイトしないで放置しておいたとか、あるいは設定をミスしていたので本来防げているはずの攻撃を通過させてしまうとかそういうこともあるかもしれません。そういうことから、ファイアウォールを置いたからといって安心せず、他の対策もあわせて実施することをお薦めしておきたいと思います。

また、ファイアウォールを置くと、そこで制限が加わるので不便になるかもしれないという感じを持つ方も多いのではないかと思います。そういった場合、例えば、ファイアウォールではアクセス制御を行わず、そこで集中して外部との通信のログを取っておいて、問題が起こったときに調査できるようにしておくといった使い方も考えられるのではないかと思います。そういう目的で使う場合に、商用ファイアウォール製品は、少し高価かもしれません。が、例えば、大学の研究室みたいな環境で、本格的なファイアウォールを置くほどでもないし、それだと少し不便かなというような場合、イーサネットが2枚ささったPCをもってきてフリーのUNIXでしかるべき設定をし、パケットだけチェックするというようなセッティングも考えられるかと思います。もちろんその場合は、お金の替りに労力がかかります。

どういう方法を取るかは別にして、何等かの形でファイアウォールもしくはファイアウォールみたいなものを置いて、そこで集中してインターネットとの通信を監視していただくということを是非ともお勧めしておきたいと思います。

ファイアウォールでは防御できない不正アクセスもあるし、他の対策もあわせて行わなければならない面もありますが、ファイアウォール自身の活用ポイントとしては、サイトの事情に合わせたセッティングをするということが大事なのではないかと思います。ファイアウォール製品を購入する場合、サイトの管理者の方が納入業者の方に十分に希望を伝えられないと、納入してゆく側というのはこれならきつめなので安全とか、これなら緩めなので何でもできるでしょうとかいうような形で、納入業者の言うがままにややもするとセッティングしてしまいがちではないかという感じを持っています。そういう場合、ファイアウォールを導入した結果あまりにも不便すぎるとか、結局ファイアウォールを置いたが攻撃を許してしまったというようなことにもつながります。やはりファイアウォール製品の機能、特性あるいは自分の構築したファイアウォールの機能や特性を把握した上で、自分のサイトのポリシーを反映した設定をして、その上で毎日のメンテナンスでは、ファイアウォールにより不正アクセスをはじいたログの監視、ファイアウォールのソフトウェアのアップデイトとかを続けていかれるというのが活用のポイントではないかと思われます。

・VPN

VPNとは、Virtual Private Network の略で、インターネットを経由した組織間の通信を暗号化することなどによって事実上のプライベイトネットワークを構築するという技術です。最近ですと、エクストラネットとかいう言い方もされておりますが、いわば他の組織と一本の専用線みたいなものでつないで通信しようというような考え方です。これもやはりそういう製品が出ますと、VPNの間は安全なので、特にセキュリティ対策はいらないと考えるかもしれません。しかし、それも程度によりけりで、接続先の相手が同じセキュリティに対する考え方で同じような運営をしているとは限りません。むやみにこの場合の接続先を信用してしまいますと、向こう側と一蓮托生になってしまいます。本当にセキュリティに気を付けないといけない環境ではやはり「適度な認証」を加えておくとか、あるいは「適度なアクセス制御」をおこな

っておくとかいうようことが大事になってきます。そういった意味で、相手に対しての何らかの評価を行って、必要なサービスだけを通過させるというような形で認証、アクセス制御をおこなっておくことが大事になってきますし、あるいはこいう特別な接続相手に対してどういうポリシーで当たるかを常々考えておかねばならないのではないかと思います。

ここまでセキュリティに対する考え方ということで、いろいろなトピックスを挙げてきました。基本的には、認証とアクセス制御とログ取りの組み合わせで大抵のサービスの対策を行ってくださいということを説明してきました。次にどこにも分類できなかったトピックですが、いわゆるデマ情報に対してどう接していただきたいかということについて説明します。

・セキュリティ情報とデマ

セキュリティ情報に限らず、最近インターネット上でチェインメールとかデマだというようなメールが飛び交ってまして、いつまでたってもこういう手のデマ情報はなくならないようです。よくあるのがこのメールを開けてはいけませんというあの「Good Times」です。開けるとウィルスがというそういうメールが流れてきたりするわけです。こういうデマを広げないためにできることというのは、それをもらったからフォワードするというのではなく、まず自分で信頼のできる情報源からある程度自分で裏を取って、その上でどうしても人に教えたければ教えるということをすれば、少しずつそういうデマ情報は駆逐されていくのではないかと思います。その上で、一次情報源へのＵＲＬなどを書いて正確なところはここにあるからここを見てくださいという形で伝えると、自分の伝えた情報の信憑性も増すでしょうし、受け取った側も役に立つ情報になるのではないかと思っております。これに関連して有名なサイトのＵＲＬを以下に挙げておきます。

- http://ciac.llnl.gov/ciac/CIACHoaxes.html
- http://www.ipa.go.jp/SECURITY/index-j.html

一番目は「サイアック」でアメリカの緊急対応機関のひとつですが、ここに Internet Hoaxes Page が設けてあり、こういうメールは本物が確認されていないとか、こういうメールは本物が確認されていなかったが真似をした別の危ないものが出回っているようだ、といった情報が載っています。もう一つは日本の「ＩＰＡ(情報処理振興協会)」というところで、やはりＷｅｂサイトの文書を出していて、こちらは同じようなことが日本語で書かれています。「こういったメールは他への転送は行わないでください」とこのＷｅｂページにも書かれていますが、こういったものがきちんとした管理者のところへ渡ったところで、その管理者によってデマであるという事実を広めて欲しいと思います。

ただ、メールを開くとウィルスに感染するというのはもともとはデマだったのですが、最近のメールソフトウェアでは添付ファイルを自動的に開くという便利な機能があって、マクロウィルス入りのメールが送られてくると、その結果マクロウィルスに感染するといったようなこともあります。あながちデマだと決め付けけてしまうのも危険を招くかもしれません。

常に言えることは信頼できる情報源で確認をするというのがいいのではないかと思います。そこにＪＰＣＥＲＴ／ＣＣを入れてくださっても構わないですが、一応守備範囲というものがありまして、一応ウィルスはわれわれの守備範囲外ということになっております。

4. サービス利用者のセキュリティ

ここではサービスを利用する側から見たセキュリティについて説明します。システム管理

者向けのセミナーで、なぜ利用者側から見たセキュリティの話か言いますと、一つには、各サイトのユーザがそれぞれ正しい使い方をして、ある程度自分の身は自分で守ってもらわないとなかなかセキュリティというのは維持できず、そのユーザにどういうふうにシステムあるいはサービスを使っていただくのかというのを指導するのは、システム管理者あるいはシステムの運用を担当されている方ではないかということからです。もう一つは、利用者側から見たセキュリティといっても、利用者の使っているPC、ワークステーション、ネットワークはやはりシステム管理者の皆さんが面倒を見ていらっしゃるわけですので、それらの設定に関連した対策というのはいくらかございます。そういったものもここで整理しておきたいと考えるからです。

・コンテンツによるリスク

ここでは、サービス自体が原因で起こる問題ではなく、サービスを経由して送られてくる情報とかそういったものが原因で発生するリスクについて説明します。ここでいうリスクとは、情報がどうやって送られてくるかというのとは特に関係なく、サーバ側から怪しい情報が届いたというのをきっかけで問題が起こるリスクのことです。

良く知られているものに、ワールドワイドウェブの特定のページを見ると特定のブラウザのセキュリティホールが悪用されてローカルファイルが消えたとか、その他の不都合な動作が起こったとか、あるいは電子メールで知り合いからワープロとか表計算ソフトのデータが送られてきてそれを開いてみたらマクロウィルスに感染してしまったとかというのがあります。そのほかにも最近のソフトウェアの高機能化に伴って、送られてくるデータをきっかけにいろいろ自動的な操作をしてくれる便利なソフトウェアが増えてきていますが、その一方でソフトウェアをデザインする側もあまり予想もしなかった形でセキュリティ上の問題を引き起こすことが多くなってきています。最近の話では、ウェブブラウザにセキュリティホールがあって、手もとのローカル側にあるPCのファイルの情報がサーバ側に伝達されるというのもあります。今後こういう類のリスクは引き続き増大してゆくのではないかと思います。

こういったことは、ユーザの習慣とか、使っているソフトウェア、セッティングとかに結構依存しています。例えばこういういろいろな付加機能の自動動作というものを全部殺して使っているとある程度回避できたり、電子メールでファイルが送られてきた場合それを開かないようにしようという習慣を身につけていたら遭遇しないかも知れません。とはいっても機能があれば使ってしまうのが普通のユーザですから、システム管理者の側で何らかの手が取れるものであれば取りたい、といういう感じではないでしょうか。

最初の対策としては、取り合えずソフトウェアをバージョンアップしてセキュリティホールがあるバージョンはどんどんと新しい版で置き換えてゆこうというのがあります。ただこれも難しい面があります。最近のソフトウェアというのはかなり急激なペースでバージョンアップ、機能の追加がおこなわれていて、こういうエンドユーザ層で使っているソフトウェアだと本当にこれが対策になるのかという意味では少し疑問符が付いてしまうのではないでしょうか。

ただ、不正アクセス対策の基本的な考え方からいけば、既知のセキュリティホールは悪用されるが、今後新しく発見されるセキュリティホールがいつまでも内緒で悪用されるということはまずありません。ある程度の時点で必ずそういうセキュリティホール情報は表に出ますから、ソフトウェアメーカの方もそれにあわせて対策することができるわけです。そういった意味で、バージョンアップをすることには、意味はあるのでないかと思います。

もう一つの対策としては、ソフトウェアの設定でそういう追加機能を殺しておくとか、自動的にソフトウェアが起動されてしまわないようにしておくという対策を取ることもできます。

あるいはどうしてもメールでファイルを送りたいが相手を心配させずにデータを送りたいといった場合、直接ソフトウェアのファイル形式で送るのではなくテキストファイルの形に変換して送ってあげるというのも対策としてあると思います。例えば、表計算ソフトですとＣＳＶ形式のテキストファイルにしてしまえば危ない情報がファイルによっては伝達されませんので、そういう習慣をユーザに付けていただくというのも一つの方法だと思います。

次の対策としては、ウィルス対策ソフトというのがあります。ウィルス以外に対しては効果はありませんが、添付ファイルの問題で大騒ぎしている昨今、かなり有効な対策なのではないかと思います。ただ最近のマクロウィルスは変種がたくさん出回っていて、ウィルス対策ソフトを頻繁に更新しないとうまく検出しないという話を聞き及んでおります。

このように、なかなかどの対策もパーフェクトにはできなくてかつどれも導入すればそれなりに利便性は低下しますので、どの程度そういうリスクを負うかという点を良く考えて導入する必要があると思います。

もう一つの対策としては、最近のファイアウォールにおけるWebのコンテンツをフィルターする機能とかメールのコンテンツをフィルターする機能などを使って危ないコンテンツをフィルタリングするとか、あるいはファイアウォール上に残されたログを後から見てなにかウィルスらしいものがダウンロードされていないかをチェックするとかといった対策を取ることが考えられると思います。

最後の対策は、発想を変えて一台少し古めのあまったＰＣでもいいですから適当に調達してきて、この機械は壊れてもいいやというマシーンにしてしまう、いわば検疫所みたいな形にしてしまう方法もあります。その場合はもちろん危なそうなところはそのマシーンを使って見るとか、その上で時々ウィルスチェックをしたりとかという方法で対策を取ることになるかと思います。ただその方法にしても、ユーザに少し不便だがそういう運用を実施していることを徹底しなければいけませんから、かえって、ユーザそれぞれの注意に任せたほうがいいかもしれません。結局どれを取るかというのは、サイトの実状にあわせて決めていくことになるかと思います。

こういうソフトウェアの問題のある部分は、ソフトウェアのベンダーの方が売れると思ってそういう機能をつけた結果ですので、ユーザの方もこういうセキュリティ対策をしてほしいというような形で何らかの方法で訴えていかないといけないと個人的には思っていますが、なかなかそうは言っても実際にばこういう作業をやるにはこういうソフトを買った方が効率はいいというのはあるわけですから、なかなか難しい面もあるんではないでしょうか。

・WWW とプライバシー

この問題も最初はブラウザを使ってコンテンツを眺める側からという視点で話を進めていきます。最初にコンテンツによるリスクというのは前述しましたので、ここでは、「プライバシーの漏洩の問題」、「ネットワーク盗聴の問題」と「電子掲示板（チャット）ページの問題」について説明します。

まず、プライバシーの問題としては二つあると思います。どうしてこれをセキュリティセミナーで取り上げる問題かとお思いの方もおられるかと思いますが、現実のユーザーはこういう捉え方をしているということで、ここで少し認識していただきたいと思います。

ネットワークの通信はＴＣＰ／ＩＰプロトコルというものでおこなっていて、ＷＷＷ(WorldWide Web)コンテンツを取り寄せるときにはそれに更に HTTP（Hyper Text Transfer Protocol）プロトコルを使って通信をやるわけです。そのときに、当然こちらの情報もサーバ側に送信しているわけです。例えばこちら側のＩＰアドレスはもちろん、今標準的に使われているバージョンの HTTP の仕様では、こちら側が使っているＷｅｂブラウザのバージョンや種類とか、どのようにリンクを辿ってきたとか、そういう情報がＷｅｂサーバ側へ伝達されます。

これは、テクニカルなそういう内面を知っている者にとっては、プロトコルのデザイン上の事実ですが、最近Ｗｅｂをご覧になり始めた普通のユーザがこれを知っているかというと必ずしもそうではないようです。例えばこういう機構を利用して「あなたはここからアクセスしてきました」と表示するページを作ることだってできるわけです。現実に世の中にはそういうページが何個所かあるらしく、知らずにそこにアクセスされたユーザの方が自分のアドレス情報がばれたといって大騒ぎされることもたまにあるようです。

これればかりはどうしてもシステムのデザイン上の事実ですので、手元の情報がばれて困るからどのようにしたらいいかと問われると、相手とやり取りをするのに自分の事情をまったく隠すというのはできないので、そういうのが伝わってもいい相手を選んでアクセスしてください、と答える以外にないわけです。ユーザの皆さんを混乱させないように運用していくためには、そういう原理的なものもある程度、説明しないといけなくなってきています。また、ユーザの皆さんに対して「単にシステムを使ってみるだけではなくて、セキュリティに配慮してください」と注意していただく必要も出てきます。

さらに、例えばこういった情報は「プロキシーサーバ」にも記録を残す事ができます。これは二種類の意味がありまして、どうしてもブラウザの種類を隠蔽したいとかそういう余計な情報をサーバ側に送るのを回避したいという場合に、手元にプロキシーサーバを置いて向こう側に送り出す情報をフィルタリングするという方法が一つ考えられます。逆の、もう一つの視点からみますと、不特定多数に対してトラフィック減とかの理由でプロキシーサービスを公開している方も世の中にはいらっしゃいますが、もしもそういう不特定多数あるいは不特定でなくてもある程度大きな範囲にプロキシーサービスを提供される場合に、プロキシーサービスにアクセスしてきたマシーンについてのログを残しておきます。もしもそのサービスを悪用する不届き者が出てきた場合に追跡をするのにきっと役に立つ事だと思います。

ここまではセキュリティ問題といえなくもないし少し違うかもしれないという感じの話でしたが、次は本当のセキュリティ問題です。

先ほど悪いコンテンツが送られてくるとそれに反応して向こうに情報を送ってしまうということに触れましたが、Ｗｅｂブラウザにあるセキュリティホールを悪用すると、ローカルの（ブラウザが動いている環境にある）情報がどこかに送られてしまうという問題のことを言いたいと

思います。これはすでにコンテンツによるリスクのところでまとめて述べましたとおり、基本的には最新のブラウザにしていただくというのが有力な対策になってきます。また継続的に同じ種類のバグがいくつも発見されておりますので、最新の情報に注意していただく事が大事になってくるのではないでしょうか。また、場合に依ってはブラウザの乗り換えとかそういった抜本的な方法を取られる事を考えてもいいのではないでしょうか。このようにどうしても新しいセキュリティホールとの追っかけ合いになるという面がありますので、どちらかといえばなるべくユーザにも不信なサイトにアクセスしないよう対策の半分を背負っていただいて、管理者の側でも工夫するようにしていかないとなかなかシステム管理が大変になってくるんではないかと思います。

もう一つ「WWWのリスク」として挙げられるのがネットワーク盗聴ですが、こちらは最近のブラウザを購入していただくといろいろな方式でサーバの側と暗号通信を行います。たとえば、クレジッドカード情報を送るときでも安全なように実装されているようです。これに関しては、ブラウザによくあるのがこの情報は盗聴される可能性がありますとかいうダイアログがよく出てくるわけですが、あれがどれくらい役に立っているのかという問題はあるのではないでしょうか。たぶん、一般的なユーザの挙動というのはそこで黙ってＯＫしてしまうとか、何回も出てくるとうっとうしいので以後このメーセージを表示しないというチェックを付けてしまうというようなことがあると思います。そういう場合でも、ダイアログが出た時点でシステム管理者の皆さんにこれはどういう意味ですかと聞いていただいて、その時点で正しい説明ができるといいと思います。そういったことから、常日頃からネットワーク通信について（技術的詳細まで踏み込む必要はないですが）お話していただくとユーザの側としても納得して行動ができるのではないかと思います。

ただそれぞれそういう盗聴を防止するテクノロジーが実装されたとして、なんとなく鍵が画面の左下で繋がっているので安全かなと思っても、本当に相手のWebサーバを信用してもいいのかということでは、完全にインフラが確立されているわけではないのではないのでしょうか。そういったことでネットワーク盗聴を気にする事も大事ですが、たとえクレジットカードナンバーを向こうに送るとかいった場合も普通の通信、会話と同じような危険というのはあるわけです。クレジットカード番号の場合はそうですし、それに限らずなにかプライバシー情報を送っている場合はある程度向こうの事を信じないとしようがない面があります。ですから、これに関してはここで管理者がどうしましょうこうしましょうといっても限界があるのではないかなと思います。

・電子掲示板（チャット）ページ

自由にメッセージの書き込みができて「電子掲示板」みたいに使えるページとか、あるいは「チャットページ」とかいわれて雑談ができるサイトがよくあります。「電子掲示板を運営しているのだがそこにいやなメールとかいやがらせのメッセージがたくさん来る」とか、「誰かのプライバシー情報が載せられてしまった。対策は取れませんかとかそういうようなことを言われたので、そのページのソフトウェアを管理しているプロバイダーには言ってみたのですがどうも対処してくれないのです」とかいうような届け出が来るわけです。プロバイダーにまだ言ってないという段階の届け出ですと「じゃプロバイダーに言ってログがあったらそこから調べるとかしてくださいね、ログがなかったら取るようにしましょうね、アクセス制限かけてないようだったらかけましょうね」というようなテクニカルな対策を解答する事は出きるわけで

す。

しかし、なかなかそうは言っても一つには不特定多数のコミュニケーションができるというのがこのサービスの魅力でもあるわけですからどうもなかなか埒が明かないということで、悩ましい話題でもあります。

もしも不特定多数のコミュニケーションを仮定していないのであれば、技術的にはかなりの対策が取れます。例えば投稿者を事前に登録しておいてパスワードとユーザを打たないと投稿はできないようにしてしまうというのが一つの方法です。この場合、誰でも頼まれれば誰でも登録するというポリシーを取ったとしても、一段階精神的な敷居が入りますので見つけたから気軽に即いたずらという形にはなり難いし、あるいはもしも昔いたずらしていた人が管理者によってアカウントを切られてもう一度別の名前でアクセスしてきても気が付けばまた追い払えるという程度には守れるわけです。もちろんそういうような運営をしている人にとって、不届きな投稿というのがアクセスログに残ればそのホストから逆に辿れば技術的には可能になってきます。逆に、不特定多数でコミュニケーションが取れて楽しいなという場合には、そういうリスクを自覚していただくしかありません。

もう一つは電子掲示板ページでよくある方法として、そこにHTMLタグを使ってHTMLの指令を埋め込めるような仕様になっていますと、なんか怪しげなサブミットボタンが現れて押すと怪しいサイトに行ってしまうというような悪戯ができるようになってしまいます。もちろんそういう事ができれば面白いという人がいれば、それはその人のポリシーなので外からどのようにいえるものでもないのですが。もしも自分でこういうサービスを運営しようと思われている方がいらっしゃるようでしたら、こういう怪しいタグとかはあらかじめ切ってしまうようなシステムを使った方が後でこういう悪戯をされるリスクは当然下がるはずです。

・電子メール

電子メールの方も話題が結構いろいろありまして、こちらもセキュリティ上の問題なのかどうなのかというところから少しづつ入っていきたいと思います。

最初に電子メールでリスクといった場合、現在電子メールのインターネット上の実装に「本質的なリスク」というのがあります。例えば、送信ミスをして内容とか宛先を間違えたまま送るというのは普通に使っていてもありがちなミスですが、その時宛先の文字が少し違ったアドレスが実在していて違う人に送られてしまったということが送った側にはまったくわからなくなってしまうというようなリスクだってリスクなのです。あるいは特定の人にしか送るつもりのなかったメッセージを、メーリングリストにばらまいてしまったという話はよくあります。このようにユーザのミスで少しまずいかなという事態を引き起こしてしまうというようなリスクはもちろんあるわけです。これは不正アクセス対策ではないですが、セキュリティの一種ですのでそれなりの対応をしていただくのがよろしいのではないでしょうか。個人的な話をしますと、よく書きかけのメールを操作ミスで送ることがあったので、キーボードをカスタマイズして送るときの操作をややこしくしたというのを個人的にやっていたりしますが、人それぞれの自分なりの対策を取られるのがよいかと思います。例えばPC用のメールソフトウェアで直接SMTPサーバに送ってしまうのではなく、PC側のキューにいったん残すというようなセッティングができるようなものがあるそうなので、そういうようなのを使ってもう一度確認するフェーズを入れてもいいのでしょう。もちろん、まったくそういうのは気にしないというのであったらもちろんそのまま使っていただいてもいいと思います。

あとはよくインターネット上のメールでいわれている事として、届いたのか届かなかったのかわからないという話がありますが、これればかりは今のデザインですと解決しきれない面もあります。一方で、ちゃんと通知が行くようにしようという機構(DSN)が実装されつつあるようですから、将来的には多少変わってくるのかもしれません。

話が更にずれてしまいますが、Return-Receipt-To というのがあります。大昔の sendmail で(といってもそれが使える sendmail は多分世の中には多少動いているんでしょうが)、ヘッダーに Return-Receipt-To というのがあると一応最後のメールボックスに届いたという通知ぐらいは戻る機能です。もしもそれを設定していた場合、例えば送るときに自動的に付けるように設定してメーリングリストに送ったら、メーリングリストじゅうから通知が届いてきてしまってメールスプールが溢れたという事故を引き起こすというような話もあります。また逆に、外からそういう通知を要求するメールが届いたときに、通知の内容に外部には漏らしたくないなと思う情報が入って出てしまうというようなことが言われていたりもします。メールを出す側も Return-Receipt-To を付けない方がよさそうですし、もしもある程度以上のバージョンの sendmail を使っていたりする場合ですと、通知を返す・返さないはコンフィグレーションで調節できますのでその辺はサイトのポリシーと照らしてご検討いただければいいんではないでしょうか。

話を戻して、電子メールの本質的なリスクとして「トラフィック解析ができる」というのがあります。たとえばメールシステムのログが残るわけですが、誰が誰宛てにメールを送ったというのがいちいち記録されているわけです(あるいは記録するようにできるわけです)。大抵のシステムでは、ログはスーパーユーザしか見られないようにはなっていたりするわけですが、あるいは誰でも見られるようになっていたりするかもしれません。それを見ると誰と誰との間でメールのやり取りが多いなというのも分かるわけです。そうするとそこから例えば男の子と女の子の間でメールのやり取りが多かったら怪しいとかいうのが分かる、というのがトラフィック解析といわれるものです。

つまり内容は分からなくても通信が何処と何処かでおこなわれているのかというのを知ることによって、背後の情報を推測することを「トラフィック解析」といいます。今メールシステムのログと言いましたが、もしもインターネット上のちょうどいいポジションでパケットを観測(盗聴」できたりしますと、始点アドレスと終点アドレスの情報から、あそこのサイトとあそこのサイトとは最近関係があるらしいというのが分かるとかいうのもトラフィック解析です。

もう一つ、「窃視」というのがあります。電子メール自体がそのままネットワーク上を流れているので、適切な場所で盗聴すると中身が見えるわけですが、こちらの方は適切な暗号テクノロジーを使うとある程度解決するわけです。

トラフィック解析のほうは電文が暗号化されているのかどうか関係なく何処と何処が通信しているかという情報で判断できるという話なので、そうすると今のインターネットのお互いに中継しあってパケットが届くという枠組みではなかなか対策は難しいのでないかと思います。そういった意味で、この辺が本質的なリスクということになりますので、これは「デサイン上の事実」として踏まえて使っていただくしかないんじゃないかと思います。そういった意味で、電子メールで書いてもいいことあるいは電子メールでの通信を止めた方がいいことというのを自覚していただくという、そういう対応しかできないのではないでしょうか。

・SPAM、電子メール爆弾

「ＳＰＡＭメール」、「電子メール爆弾」というのは今ホットな話題になっています。基本的には、ＳＰＡＭというのは、多数のメールを多数のアドレスにばらまくことで、電子メール爆弾というのは誰か特定のアドレスにまとめて集中的に送り付けること、という程度の使い分けです。

どちらもなんかの方法でたくさんメールを送り付けてくるわけですが、そこで送り付けられたからそこのＦＲＯＭとかをみてあんたやったのといって抗議すればいいかというとそういうわけではないのです。基本的に今の技術ですと電子メールのヘッダー部分ＦＲＯＭとかそういう部分はみんな「偽造」することができます。その偽造の方法も少しインターネットのプロトコルを勉強すれば分かる程度のものですので、結構敷居は低いということになっています。それでは偽造されていたとすると、やられたらどうすればいいのかなということになるわけですが、ある程度信用できる情報もそこには残っているわけです。例えばメールを送り付けられたシステムで正しくログを取っていれば、一個手前のホストまでは確実に分かるわけです。そこで一個手前のホストに問い合わせをするというのが正しい対応になるわけです。その場合もどちらかというと「そこから送りませんでしたか」ではなくて、「あなたのところも悪い中継に使われませんでしたか」というような形で話をしないとかえって話がこじれる元になりますので気を付けてみてください。

特に電子メール爆弾とかは、個人的な嫌がらせだったりして不正アクセスというのを悪いこととしてやっているわけですが、ＳＰＡＭの場合はよくみるといわゆるダイレクトメールなわけです。その場合やっているほうは悪いことをやっていると思ってやっているかというとそんなことはなくて確信犯的にやっているわけです。ですから、トレースしていって「あんた悪いから止めて」というのは通用しません。そうすると泣き寝入りしかないのかということになりますが、たとえば発信元が使っているプロバイダに連絡してみるのがいいだろうということがいわれています。どういうことかというと、普通のプロバイダの約款を見るとＳＰＡＭを送信するのに使ってはいけないというのが約款に載っていますし、あのプロバイダはＳＰＡＭを許しているのだと言われても困るだろうから、きっと何か対応してくれるのではないかというふうな感じなのです。本当にそれで効果があるかというと、プロバイダによってはなかなかそういうところの調査までうまくできないところもあるらしいですし、もちろんちゃんと対応してくれるところもあるらしいし、という事情のようです。

こういうものが送られてきたときの対応がはっきりしないこととから、「予防」することの方が賢いということで、どう予防すればいいかを考えてみたいと思います。最初は送られてきたメールアドレスというのは、向こうが入手できたアドレスということになるわけですから、自分がアドレスを教えなければ送られてこないんだね、と、そういう対策がまず考えられます。たとえば、メーリングリストなどに参加する場合とか、ネットニュースに投稿する場合とかそういう場合には、あとでＳＰＡＭなんかが送られてきても捨てればいいような「別のアドレスで」参加するというようなやり方が一つ考えられます。もう一つは電子メール爆弾、ＳＰＡＭとかの中継に悪用されてしまう側のシステムの対策として、「自分に関係ないメールの中継を止めてしまう」という方法があります。この方法については、電子メールサービスを提供する側のお話のところで説明します。

基本的にはＳＰＡＭ、電子メール爆弾とかの予防にはまずはアドレスを知らせないということが大事ですし、システムを運営している側にとってはメールを勝手に中継されないように

設定をしておくというのが大事だと思います。それでは個人ユーザがダイヤルアップで参加している場合はどうなるのかというと、プロバイダーの方でメールの中継を制限するというのは、なかなか難しい事情もあるようです。それでも少しづつユーザが登録したアドレスに限って切るとかいうサービスが実装されつつあるようですので、今後の動向に期待したいところです。次に電子メールを使う側のリスクとしてメーリングリストに参加するときに関係する話題に移っていきたいと思います。

・メーリングリスト

メーリングリストに関するリスクにはいろいろあるわけですが、最近注目したいのがWWWでメーリングリストのアーカイブが見られるというサービスです。このサービスは近頃はかなり普及していまして、アーカイブが公開されている事実を知らないでメールを送ってしまいますと世界中から自分のメールが見え見えになってしまうわけです。その結果、例えば、自分の名前をサーチエンジンの検索キーに打ち込むと、検索結果に昔の自分のみっともないメールがぱっと出てきたりとかそういうようなことが起こったりするわけです。それだけならすこし恥ずかしいかなでいいわけですが、それで何がやられるかというと、電子メールアドレスがそこに載っているわけです。自分がＦＲＯＭに書いているアドレスというのはメールを送れば届くアドレスですから、そういうアドレスをＳＰＡＭを送る側の人たちが頑張ってかき集めてきて、それでメールを送ってきてしまうということでやられてしまいます。自分のシグナチャに自分の住所、電話番号などを書いていたら当然世界中から見え見えです。メーリングリストだからクローズドだからいいかと思っていても実は世界中から見えてしまう、そういうようなことになってしまいますのでこれは注意が必要です。

ほかにも、もちろんメーリングリストのメンバーには当然自分のメールアドレスがわかってしまいますし、そのメーリングリストのアドレスにＳＰＡＭが一通来ると自分のところにもＳＰＡＭが来るとか、そのメーリングリストに電子メール爆撃が来ると自分のところにも爆撃が来るというようなそういうリスクは当然負うわけです。その辺りがメーリングリストへ参加するときに目立つリスクかなということになります。メーリングリストも電子メールですから、他の電子メールにまつわるリスクもあります。

こちらの方の「対策」としましては、メーリングリストに参加する前にメーリングリストの運営ポリシーとか運営内容、例えばWWWへのゲートウェイがあるのかとかそういうようなことを確認していただく。もしそのポリシーでは困るということですと参加しないか別のアドレスから参加するかというようなことを対策して行うことになるかと思います。メーリングリスト経由でのＳＰＡＭとか電子メール爆撃に対しては、メーリングリストの運営する側でリストに参加していないアドレスからの投稿は切るとかそういうような対策を施せばある程度回避できます。そちらの方は後でサービスを提供する側から見た整理方法でもう一度説明します。

5. 一般的なサービスのセキュリティ

これまではサービスを使う側からのセキュリティについて述べてきましたが、次にサービスを提供する側のセキュリティについて説明します。

・セキュリティホール

セキュリティホールというのは、どんなサービスにしろ付いてまわってくるものです。セキュリティホール対策というのをやればいいのですが、取り合えずシステムは動いているのでそのまま置いとけばいいのだろうと、私も昔はサボっていましたので皆さんもある程度そう思ってサボっているのではないかと思っているわけです。しかし、そういうふうにしてセキュリティホールを放置しておくとインターネット環境でよく問題になるセキュリティホールが三種類くらいあります。つまり、外から内側にあるホスト上の情報を盗むことができるというようなセキュリティホール、ホスト上である権限で自由にコマンドが実行できるセキュリティホール、一般ユーザとして侵入した後でスーパーユーザ権限が盗れるセキュリティホールの 3 種類です。最初の「設定情報の漏洩」ということですが、パスワード情報が見られたからどうなんだとか inetd.conf が見られたから本当に危ないのかというと、どうも phf プログラムのセキュリティホールでパスワードファイルが盗まれた結果それをクラックされてやられたんじゃないかという事例とかは実際に報告を受けております。ある程度このようなリスクには備えていかないといけないのではないでしょうか。

特にインターネット上でこういうセキュリティホールを破るツールが流通している昨今、基本的にこういうセキュリティホールを破るような話が報告されますと、すでにこうやるとセキュリティホールを破れるよというツールも裏で出回っています。それでそのツールを入手した人がそれでどんどん他のサイトを攻撃してしていくと、攻撃に成功してしまいます。パスワードファイルが盗まれたというくらいでは被害のうちに入っていないと思ってらっしゃる方も結構いらっしゃるようですが、実際にそれを悪用して侵入された事例も報告されていますので、情報が抜かれたらそれを参考にしてやられるかもしれないという認識はしていただきたいと思います。盗んだ情報を使ってあるいは直接セキュリティホールを使って「侵入」するということをおこなったりするかもしれません。侵入した後でまたさらに別のセキュリティホールでスーパーユーザになって、スーパーユーザになればなんでもできますので今度は一発で侵入できるように裏口を設置したりとか、あるいはパケット盗聴プログラムをインストールしたりとか、そういうことを侵入者はやっていきます。これも実際にご提供頂いた方に感謝したいのですが、複数の被侵入サイトに同じような不正アクセスツールが転がっているなというのをこちらでも確認しております。見ないと実感が湧かない方も多かろうとは思いますが、一応世の中このぐらいは危ないんだなという認識が必要です。

「対策」ですが、どんどんバージョンアップして穴の埋まったソフトウェアに取り替えるということをやっていただければ、それはもちろんセキュリティ対策になるわけです。ただそれだけでは大変な面もありますので他の対策と組み合わせていただくのがうまいやり方です。

例えばインターネット全体に対してサービスを提供する必要のないサービスというのがいくつかあるはずです。電子メールスプールをリモートから呼び出すサービス、最近古いバージョンの穴が攻撃されていましたIMAPなどもそうですがそういうサービスですと、正規のユーザがいるマシーンからだけサーバが起動できるようにしておくのが一つの有効な対策になります。もちろんそのサイトのユーザが悪いことをした場合はそれでは防げないわけです

が、インターネットから受ける攻撃への対策には役に立つわけです。

あるいはバージョンアップがいますぐにはできないが、少しあとにはできて、それまでは止めてもいい場合は、バージョンアップまでの間はそのサーバだけ止めておくというようなこともお勧めできる対策の一つです。あるいはこれはもう最近使っていないから止めてしまっていいよというような場合ですと、この際思い切ってそのサーバは止めていただいた方が今後新たなセキュリティホールの登場にも気にする必要もないわけですから、そういうふうにしていらなくなったら使わないようにして止めていくというのも対策の一つです。

セキュリティホールというのは次から次へ出てくるという性質のもので、まじめにこれに対策しようとすると結構大変になってきます。そういった意味では、ファイアウォールで入り口を固めておいて内側は少し更新の間隔を空けてもいいかなというような運営もあるかもしれません。本当にセキュリティに気を使っている環境ではそれでは不足で、やはり中でもいつでも更新していかないといけないのだというような場合があるかもしれません。パッチは、どうしても動いているシステムに手を加えていくわけですから、なかなか当てにくい面もありますので、そういう場合は別途テスト用のシステムを確保するとか、あるいは月に一度メンテナンスの日があるのでその日なら止められるというような運営も考えていかないといけないわけです。

ただ不正アクセス対策上からは、知られているセキュリティホールについては、それを攻撃するツールは出回っているので、すぐ対策を施すさないとまずいということは言えます。

・パスワード認証

前でもユーザ認証ということで簡単に説明しましたが、一般的なインターネットに直結してシステムを運用している環境では、かなり重要なトピックだと思いますのでもう一度説明したします。パスワード認証の場合の典型的なリスクとしては、この前ログインしたときのパスワードをそのまま繰り返されて入られてしまうというリスク（「リプレイ攻撃」）がまずあります。あるいは「辞書攻撃」といって普通のユーザが思い付くのは辞書に載っている単語なので、辞書を総当たりで調べると本当に可能な文字の組み合わせを全部当たるよりは早い時間で破れるという攻撃方法があります。次が「ブルートフォース攻撃」（「力まかせ」攻撃）なわけですが、可能な文字の組み合わせを全部当たるという攻撃方法のことです。これにとって8文字というのは少しづつ短い存在になりつつあるようです。ものの本を当たるとこれぐらいの計算機パワーで8文字の組み合わせ全部当たるのにどれくらいかかるとか、6文字だとれだけかかるとかというようなデータが出ておりますが、基本的には6文字とか7文字ぐらいぐらいのパスワードは、もうブルートフォース攻撃の射程に入ってきています。8文字でも全文アルファベットですと、危ないかなという感じのようです。8文字のパスワードの場合は、基本的には大小文字を混ぜて、数字と記号も混ぜるという形で運用していただきたいと思います。辞書攻撃とかブルートフォース攻撃というのは、もう本当に古典的な方法ですが、ただ古典的な分だけ基本的な攻撃方法ですからこれに対して備えをしておくことはある程度セキュリティの確保にとって重要なことです。よくある方法としては管理者があらかじめ「crack」ツールを使って自分のユーザのパスワードを破っておいて、破れたら「あんたパスワード変えてくださいね」というようなそういう運用があるわけです。そういうことをやってもいいでしょうし、ユーザにパスワードを定期的に変えるようにうまく指導していただければと思います。世の中にはもう一つ便利なものがあって、パスワードを設定する段階で危ないパスワードをはじくツール

があります。何種類かあるようですがなかなか標準のパスワードコマンドを入れ替えるあるいはパスワード機構を入れ替えるのは普通の管理者にとって勇気のいることですので、その辺はテスト環境で試されてから少しづつ導入されてもよろしいのではないでしょうか。何といっても一番恐いのは、「パスワードファイルの漏洩」です。クラッカーがシステムに侵入しようとする場合、パスワードファイルが盗まれると侵入に有利になってしまいますので、セキュリティ十分注意をしていただきたいと思います。

対策のまとめとしては、基本的にパスワード認証というのはユーザの自覚が大切です。とにかく強いパスワードを付けるように指導頂くというのが大事になってきます。

パスワードファイルの漏洩に関しては、シャドウパスワード機能というのがあります。暗号化されたパスワードであってもスーパーユーザしか見られないようにするという設定のことをいいます。一部のシステムではオプションで利用できますし、一部のシステムではフリーソフトウェアをインストールすれば可能ですし、一部のシステムでは標準の機能でセッティングを変えれば生きるとか、最初からディフォルトで生きているとかいろいろ違いはありますがそういう機能があります。これを実施しておきますとどういう利点があるかといいますと、例えば最近流行の phf アタックの場合 phf というプログラムは一般ユーザ権限で最初起動されますので、そのプログラムからは「シャドウパスワードファイル」に記載されたパスワードは見られないのです。その結果、いきなりパスワードを持っていかれた結果侵入を受けるというリスクは低減させることができます。ただシャドウパスワードになっているシステムでもユーザ名が登録されているファイル、シャドウでないほうのファイルがあるわけですが、そちらはやはり盗むことができます。その場合、ユーザが安易なパスワード、たとえばアカウント名と同じパスワードを使っていたりすると、もうそれだけでシャドウパスワードにした意味がないなということになってしまいますので、シャドウパスワードにしたからといって絶対安全かというとそうでもないわけでが、やらないよりはやっていたほうがはるかに安全だと思います。

もちろんシステム管理者によってあらかじめクラックを実行しておいて、ユーザに警告をする、あるいはどうしても聞き入れてくれなければしかるべき手段に訴えるというようなそういう運用もあるかとは思います。あとシステム管理者によるクラックというのは、誰もが認めている運用方法ではないというのは一応心には留めておいてください。組織の運用ポリシーとして管理者はこういうことをやるんだということをユーザが納得していればいいわけですが、これをやってしまいますともっていきようによっては管理者が少し気まずい思いをするだけだったということもありますので、その辺は気を付けてうまくやっていただければと思います。

もちろんパスワード認証では危ないので止めてしまおうというのは、結構有力な対策方法ですので必要に応じてそういう「認証機構」、例えば「ワンタイムパスワード」、お金がなければフリーソフトウェアのワンタイムパスワードもありますし、お金があればいろいろな製品もありますのでそのようなものを選んでいただくのもいいわけです。また将来ひょっとしたら指紋で認証とか網膜検査をやるとかそういうものが民生用で気軽に使えるようになるかもしれませんので、そういう時代が来たらそういうのもご検討されてはいかがでしょうか。

・ネットワーク盗聴

基本的に今のIPネットワークではネットワークの途中にいるマシーンでパケットを傍受することは可能です。その対処法となるとなかなか難しい側面もありますが、基本的には何処で盗聴されても構わないようにしたいのであれば、暗号通信を使うという対策があります。そうではなく、例えばローカルなシステムに侵入されてパケット盗聴プログラムを設置された事

実を検出する方法として、常時ネットワークインタフェースを監視しあらゆるパケットを受信するモードになっていないかを調べる方法もあります。たとえば、ＢＳＤ系のネットワークの実装ですと、「ifconfig」というコマンドを打ち込んだときに、promiscuous の promisc という表示が出てくると、それが怪しいかもしれません。これはネットワークの実装によって差があり、自分のシステムが promiscuous モードのときに promisc というのが出るという事実をあらかじめ確認しておく必要があります。あるバージョンのシステムですと、どうやってもコマンドでこの promiscuous モードになっている、つまりすべてのパケットを拾うモードになっているのを検出できないシステムも世の中にはあるようです。そういうシステムですと、この方法は取りにくいかもしれません。

もう一つの有効な方法としては、「スイッチ」いわゆるイーサネットスイッチとかスイッチングハブとかいって、パケットを必要なところにしか届けないということをやってくれるハードウェアをハブの替りに導入します。そうすると、同じイーサネットセグメント上のノードにパケット盗聴プログラムが仕掛けられていたとしても、パスワードなどが盗まれないで済みます。この場合は、「1マシーンに対し1ポート」を割り当てる必要があります。最近イーサネットスィッチは昔に比べて値段が下がりつつありますので、場合によっては検討してもいいのでないかと思います。

一方で本当にパスワード盗聴に限った対策になりますが、認証方式として盗聴されても大丈夫な方式というのがあります。それは、ワンタイムパスワードであったりあるいは電子メールサーバのＰＯＰで使われている「ＡＰＯＰ」という方式であったりしますが、そういう方式を選んでいただければセッションの内容は盗聴されるかもしれませんがそれによってパスワードは盗まれないということになります。

- **IP spoofing**

Spoof という単語には、偽造する、偽証するあるいは詐称するとかの訳語が当たります。なんでもＩＰパケットを偽造すれば IP spoofing なのですが、基本的には IP spoofing といった場合、偽造するのはＩＰアドレスの始点側のアドレスです。これをやられるとどういうことが起こるかというと、例えばＩＰアドレスでアクセス制御をかけているサービスに対してこの IP spoofing を施したパケットを突っ込めば、そのアクセス制御を通過させることができます。最近「Land Attack」というのが世間で騒がれていますが、これも IP spoofing と特定のＴＣＰの実装の穴をうまく組み合わせた攻撃の一種です。

いろいろなレベルの対策がありますが、一番楽なのは自分の組織のアドレスは自分の組織の外にはないというのを利用して、入り口で「パケットフィルタリング」によってこういう怪しいパケットを落してしまうことです。このとき、ほかのアクセス制御の設定においては、外部のアドレスを指定しないほうがよいでしょう。自分のところのアドレスは自分の外にはないという規則では、外界のアドレスに偽造されたパケットは落せないのです。

パケットフィルタリングのセッティング方法は、個々のルータ、ファイアウォールによってみんな違いますから、具体的にどのようにすればよいかを言うことはできないのですが方法だけを言っておきます。外部と接続されている専用線などのインタフェースからパケットが入ってくる側にフィルターをおき、自分の組織のアドレスを始点側に持つパケットを拒否します。この時に、内側のアドレス以外に、いわゆるプライベートＩＰアドレスと言われている10．0．0．0/8、172.16.0.0/12、192．168．0．0/16 を一緒に落しておくようにするのもいいことだと

思います。あとは127．0．0．0/8というのもこのあたりで落してしまっていいアドレスです。

こういう形でパケットフィルタリングをしておくことは、今後 IP spoofing をうまく使った攻撃ほかの攻撃もかなりの部分防げることになりますので、各組織の基本の設定だと思います。しかしこれにも少し問題がありまして、「モバイル IP」で使っているようなプロトコルですと、自分の組織のアドレスなのに外側に移動してしまうということになるはずなのですが、そちらの方はそれに対応するプロトコルが別途デザインされつつあるようです。今後モバイル IP が気軽に使えるようになる頃には、解決策が普通に利用できるようになるのでないかと思います。

・ DNS spoofing

DNS spoofing とは、ネームサーバの登録情報にうその情報を紛れ込ませるものです。例えば、ホストネームベースのアクセス制御を不正に通過させたり、特定のWebサーバをアクセスしたつもりが別のサイトを見にゆくようにさせたりというようなことを目的として行われます。ネームサーバの実装上の問題、DNSサービスあるいはDNSのクライアントリゾルバ側の実装上の問題を突いたものとかいろいろあります。

こちらの方の対策は、基本的にはなるべくセキュリティパッチを当てることあるいはなるべく最新の BIND に入れ替えること以外にないと思います。一方で自分のところの認証をこれで破られるのは困るという状況がもしありましたら、うまく設定すると手元のオーソライズドサーバつまり自分が設定して自分が書いた情報しか知らないオーソライズドサーバの情報だけ認証に使う設定も可能かと思われます。ただもちろんそれをやるためには、BIND 関係あるいはネームサーバ関係に対して詳細な知識が必要になってきます。

・ログ

ここでは、ログを取ることによって起こるリスクについて説明します。一般的なUNIXシステムですとログを取るために syslogd というプロセスが動いています。そのプロセスにログの情報を他のマシーンからおくることによってリモートでログの蓄積を行うことが可能になっているわけです。しかし、例えばそういうサービスを運用していると逆に偽のログ情報をそのデーモンに送り付けることによって「偽のログ」を作ることもできます。また、そのときにたくさんログを送り付けますとログ領域の不足を意図的に招いたりといったようなことをすることができます。

また少し syslogd の話から離れまして、あらゆるシステムのあらゆるログに共通の話になりますが、何らかの方法で管理者の特権を入手したあるいはログが記録されているファイルに対するアクセス制御がうまく設定されていなかったという事態がありますと、そのログが侵入者によって「改ざん」されてしまうというリスクがありえます。この結果具体的にどういう問題が起こるかと言いますと、例えば改ざんの方では、自分のシステムに侵入があった痕跡を侵入者によって消されることが起こり、それによって追跡あるいは対策が難くなるという問題があります。またログ領域の不足のほうでは、ログ自身がこれ以上記録できなくなったりとかログを格納している領域が溢れた結果それと同居している機能に支障が出ることが起こります。ログが溢れた結果、例えば、よくあるセッティングですが、メールスプールが同居していたのでメールが届かなくなって送れなくなるトラブルが起こったりします。

この対策としては、ログ領域をログ用に特別に確保するとか、古いログを放棄していく期

間を自分の手持ちディスクの範囲内で調整していくことが考えられます。もちろん、ログファイルのアクセス権をうまく設定することも大事ですが。一方、UNIX系のシステムで syslogd デーモンが動いていてそれでログを集めているような場合ですと、その syslog のパケットをどこかでフィルタリングして外からログのデータを送り付けられないようにするという対策を一つ候補に入れておいていただきたいと思います。最近のOSについてくる syslogd ですと、オプションを指定しないと外のマシーンから送り付けられたログを記録しないものがありますので、そういう設定を状況によってうまく使い分けて頂くのも一つの方策ではないかと思います。

さらにセキュリティ管理に気を使うのであれば、例えば何らかの方法で収集したログをプリンターに印刷して紙にでも記録しておくとか、あるいは遠隔ログインを一切禁じて最低のサービスしか提供しないログ取り専用マシーンを用意して、そこに全部ログを集めて管理するという方法もあります。ただ一般的な環境ですと、そこそこ普通にシステムのログを取るというような感じでも十分ではないかと思います。

一応ログ取りサービスもサービスの一つですので、サービスを動かしていればセキュリティ上のリスクはあるということで例としてあげておきました。あらゆるセキュリティ対策の基本の一つとして記録を残しておくというのがありますので、記録が十分に残るようにディスク領域の確保には気を付けていただきたいと思います。

・メール配送サービス

続いて、メール配送サービスにまつわる説明をします。

メール配送サービスそのもののリスクとして今気を付けていただきたいのは、中継地点として使用されるつまり前にSPAMとか電子メール爆弾のところでも述べましたように、電子メールの不正な中継に利用されるリスクと、フリーソフトウェアである「sendmail という特定ソフトウェアへの攻撃」です。sendmail は昔から多くセキュリティホールが発見されていて、常にバージョンアップをしていかないと危険なソフトウェアの一種でもあります（逆に言うとそれだけみんなが使っているので穴があってもすぐ見付かってすぐ対応ができるという見方もできますので、sendmail が危ないんだという解釈には必ずしもならないと思います）。

Sendmail をお使いの場合には、必要なバージョンアップとかパッチ当てをしていただかないと危険ですので注意してください。ベンダー版の sendmail、OSにそのまま付いてくるものをそのまま使っている場合は、適宜ベンダーから出ているセキュリティパッチを当てていただくというのでいいと思います。フリーソフトウェア版の方をインストールしたサイトの場合には、随時バージョンアップ情報に気を付けていただいて、適宜インストールする必要があります。このように、メール配送サービスの方も随時新しいソフトウェアにしてくださいというのがあります。

もう一つの中継利用についてですが、JPCERT／CCでは、中継利用を禁止するように設定をしてください、という趣旨の技術文書を出しています。少し探しにくいのですが、JPCERT/CC の Web サイトでは「JPCERT／CCからのお知らせ」というところに一通り発行文書が並んでおり、そこに「電子メール配送プログラムの不正利用（予期しない中継）に関する技術メモ」というタイトルで載せておりますのでそちらも参考にしてください（http://www.jpcert.or.jp/tech/97-0001/）。

簡単に説明しますと、一般的に sendmail をはじめ多くの電子メール配送システムは、とく

に何も設定しない限りどこからどこ宛てのメールでも中継するような仕様になっています。昔のインターネットですとこれでも良かったわけですが、今のSPAMが横行しているインターネットではこのまま放置しておくと自分のサイトの計算機資源が不正に中継に利用されてしまうことが起こります。これが具体的にサイトの管理にどれほど影響を与えるかということですが、あまりこういう事を言ってはいけないのかもしれませんが、十分に能力を持っているサイトですと計算機資源が消費したからどうこうとか、ネットワーク資源を塞いだからどうとかということが起こらないサイトがあるかもしれません。あるいは、「回線の幅が細いのに一杯送りやがって」というサイトがあるかもしれません。

どちらかといえばこの攻撃による不利益というのは、計算機資源が浪費されるというよりはむしろ（送られていったサイトからは直接送信元としてみえてしまうのは自分の組織のマシーンとなりますから）、外のサイトから苦情が殺到してしまうとか抗議のメールが殺到するとかになります。または、SPAMというのは結構いい加減にやっていて、例えば数字のアドレスがあったら数字を少しづつ変えていけばそれもきっと有効なメールアドレスかなという感じで送ってしまったりしますので、その結果大量のエラーメールが出ます。そして管理者のところにエラーメールが山のようにきて仕事にならないという被害が起こります。こちらの方がもっぱら起こる被害だと思います。

予防対策ですが、最近のソフトウェア（たとえば sendmail のバージョン 8.8.6 以降）ですと特定のホストからの接続を拒否したりとか、あるいは自分が中継したくない送信元、受信先の組み合わせは中継を拒否したりといった設定が可能になりますので、そういう設定をします。

もう一つ、こちらはもっとふるい sendmail でもできることですが、とりあえず sendmail.cf の設定をすることによって Received ヘッダに記録する項目を増やすことができます。既にたくさん記録している場合はそのままで結構なのですが、例えば接続してきた相手のIPアドレスとかいろいろな情報が記録できますので、いろいろ調節しておくことによって攻撃に使われた場合のトレースがしやすくなるということになります。

どういうふうに設定すればいいのかという話ですが、残念ながらここは sendmail のチュートリアルではないので割愛します。その辺のところはJPCERT／CCのWebページにある技術メモを参照してください。

以上のように中継利用禁止をしてしまおうという対策もありますが、もっと簡単な方法としてメールを受け取る必要のない、つまりメールスプールを自前で持っていないホストがある場合には、そのマシーンでSMTPサーバつまりメールの受け付けを止めてしまうという方法があります。一般的な設定では、sendmail –bd –q …というコマンドラインで sendmail が起動され、SMTP サーバとして動いています。そのうちの–bd というのを取るとそのホストに対してSMTP接続をすることができなくなります。この場合そのホストからは、「メールは出せるが、受け取れない」設定になります。その場合でも–q の後ろに時間を付けて起動しておくと、デーモンとして起動され、キューに未配送のメールが溜まっていた場合の再送とかはやってくれます。使わないものは止めておくという原則に従えば、そういう設定も一つの方法だと思いますので少し記憶にとどめてくだされればと思います。

・メールの自動処理

メールの自動処理をした場合のリスクについて説明します。メールの自動処理といいま

すとメーリングリストを提供するときのリスクというのが含まれるわけですが、ここではメーリングリストサーバに拘わらず、メールを自動処理プログラムに入力した場合のリスクに関して説明します。

この場合よくあるリスクとしては設定ミスに起因して問題が発生する、例えばエラーメールをたくさん出すとか、本来そのプログラムを動かそうと思っていた権限ではない権限でプログラムが動いてしまうという危険があります。そしてそれがセキュリティホールとなり、情報の流出に繋がってしまうとか、自分のファイルを消してしまうとかそういった事故に繋がります。とちらかといえばこれを狙って不正アクセスが行われるというよりはむしろ単純に設定を間違えた結果、自分のシステムを自分で破壊してしまうという方のリスクの方が高いとは思いますが一応ここでは挙げておきたいと思います。

もちろん起動されるソフトウェアに含まれるセキュリティホールも気を付けるべきです。例えば、有名なメールの処理ソフトである「Majordomo」のあるバージョンですと、あるメールを送ると Majordomo の権限で、不正にコマンドを実行できるということができます。新しいバージョンでは対策が施されていますが、そういうセキュリティホールもありますから、有名なソフトウェアだと狙われる可能性もあります。

対策としては、新しいバージョンのソフトウェアにどんどん上げていくというのが一つ重要な対策です。設定ミスに関してはよく確認していただくしかないわけですが、それとは別に特定のソフトウェアでないと起動できないようにするというような制限を加える方法があります。つまりユーザが勝手にプログラム起動をセッティングしてその結果そのユーザの設定が問題を引き起こすということはこれで防止できるわけです。たとえば「smrsh」というソフトウェアがあります。あちこちのセキュリティツールのアーカイブにもあります。フリーソフトウェアとして配られている sendmail の最近のリリースにも入っていますので、入手は比較的たやすいと思います。この smrsh を設定しておくと、特定のディレクトリに置いたソフトウェアしか sendmail から起動できないようにできます。昔よくあったセキュリティホールに「decode」というエイリアスがありました。/etc/aliases に decode というのがあって「uudecode」というプログラムに入力できるように設定されている場合、それはセキュリティホールです。普通の設定だと(いまさらこれが有効になっているところはあまりないとは思いますので特に気にしなくてもいいとは思いますが)、もしも decode:root という設定があった場合には、これを使った攻撃へのトラップとして用意されいるものだと思ってください。decode の宛先が管理者になっている場合はそのまま置いておいてかまいませんが、プログラムになっている場合はまだこういう穴をつぶしていなかったのかと思いながらコメントアウトしてくださるようにお願いします。こういうところから地道に対策を実施していただくことが大事なのではないかと思っています。

・メールボックスサービス

妥当な用語が思い浮かばなかったのでこういうメールボックスサービスという用語を当ててしまいましたが、つまり具体的にいうとＰＯＰとかＩＭＡＰのサービスのことを言っています。このサービスによるリスクは二つあります。セキュリティまず第一のリスクとしては、メールを読み出すときにおこなわれる認証時における「ネットワーク盗聴」による「パスワード漏洩」というリスクがあります。もちろんパスワード以外にも電子メールの文章もそのまま盗聴されてしまう危険もあります。もう一つは特定のソフトウェアのセキュリティホールへの攻撃です。

最近流行しているものに「IMAPサーバのある古い版のセキュリティホールを突いた攻撃」があります。セキュリティホールへの攻撃の対策としては、最新版のソフトウェアを使うことと、その上で特定の本当にサービスをする必要のあるホストに対してだけアクセスを許可する設定にする、さらにアクセスが行われた時点で記録が残るようにしておくことが考えられます。JPCERT／CCでもIMAPサーバに関係するアナウンスを流しております(http://www.jpcert.or.jp/info/97-0004/)。基本的にはUniversity of Washington版のIMAPサーバの古い版に含まれているセキュリティホールが大元になっています。したがって、これを元に作られたシステムには同じセキュリティホールが含まれているおそれがあります。対策としては、このIMAPサーバと、セットになっている(普通のPOPサーバではない)IMAPへのゲートウェイとなるようなPOPサーバをバージョンアップするか使っていなければ止めてくださいということを言っています。あとは一般的な対策として、もしもやられたのであればパスワードを変更し、勝手なアカウントが作られていたら早急にそのアカウントを消すこと、ある程度やられた可能性があってどれほど裏口を仕込まれたのかわからない場合は、再構築することをお奨めしています。また、監視用ソフトウェアのインストールという項目では、「tcp_wrappers」というのをここではお勧めしています。tcp_wrappers 以外にも、同じ目的のソフトウェアとして xinetd というのがあります。どちらも、アクセスのログとかアクセス制限に有効ですので紹介しておきたいと思います。これらのソフトウェアはフリーで配布されていますので、UNIX系のシステムをお使いの方はお試しください。

このIMAPサーバのセキュリティホールに関係してここで注意しておきたいのは、「IMAPは使っていないから多分システムに入っていないので多分安全だろう」と思っている方は、本当に動いてないことを知っている方を除いて、もう一度本当に動いていないか確認して見てください。OSによっては、その University of Washington 版のIMAPの危ない版がいつのまにか入っていることがあります。いつのまにかというのは、インストールのときにソフトウェアを選ぶと入っているとか、インストールのときに「contrib」を選ぶと入っているとか、そういう種類のものです。もしもそういう物がセットに含まれているとしたらいつのまにか起動している状態になっているかもしれません。関係ないと思っても本当に動いていないかどうかだけはもう一度改めて確認くだされればと思います。

・情報提供サービス一般

Webサーバそのものの説明は後回しにして、最初に Anonymous FTP や、(少し下火になりましたが) WAIS、gopher のような、情報提供サービス一般にあるリスクのほうから説明します。

最初に思い浮かぶ典型的リスクとしては、こういう外向けに広く公開している情報提供サービスというのは大抵 wwww...とか ftp...とかいうホスト名を付けますから、その結果そのマシーンでは明らかに Web サーバが動いているとか ftp サーバが動いているので狙おうかという形で標的にされやすいというリスクがあります。しかも、情報提供サービスを動かそうとしているので、基本的にある程度アクセス制御をオープンにせざるをえない面があります。例えばFTPサービスを提供しているのにFTPを遮断するということは絶対できないわけですからそういった意味で侵入のリスクは大きくなると思われます。もちろんインターネット全域を対象に公開している場合は世界中がそういう相手になってくるわけです。

もしもセキュリティホールを突かれて侵入を許すとどういう結果になるかと言いますと、例

えば自分のサイトが知らないうちにアンダーグラウンド情報公開用サイトにされてしまったり、あるいは一般公開しているコンテンツが改ざんされてしまったりというリスクを負います。Ｗｅｂサイトがやられてしまうと必要以上に世間の注目を集めてしまうというのが現在の状況ではないかと思いますが、その結果単純に、やられたから復旧したという以上のダメージを社会的に負わされてしまうという側面があります。単に侵入を受けた、単にサービスを悪用されたという程度ではすまないところが情報提供サービスにあるわけです。しかも守りにくいということですから、十分に気を付けていただきたいと思います。

少し不正アクセスとは話がそれますが、「WWWのロボット」いうジャンルのシステムによって自分の意図しない情報開示が行われてしまうというようなリスクもあります。このロボットいうのは具体的にどういうものかといいますと、皆様がよく目にするサーチエンジンの裏で動いているものです。たとえば特定の検索サイトに自分の名前を打ち込んだら、自分の住所が出てきたというようなしゃれにならないことが起こったりしているみたいです。情報提供サービスを動かすと、自分が想定した範囲以外にも情報が開示さられてしまう、使われてしまうというリスクが存在します。

まず一般的な不正アクセスのリスクを減らす対策の話から進めていきます。くどいようですが、最新のソフトウェアにバージョンアップしてください。それが基本です。ＷｅｂサーバにしてもＦＴＰサーバにしてもセキュリティホールが定期的にというほどではないにしても、コンスタントに見付かってきていますので、バージョンアップしていただくのがまずは第一歩かと思います。ここでよく Anonymous FTP を運用する場合はなんとかがいいという話があるわけですがその話は後の方に譲って、ここでは取り合えずベンダーパッチを当てるとか最新版のソフトウェアにするとかの程度で話をとどめておきます。

よくあるのがその情報提供サービスのサーバから何らかの外部プログラムを呼び出してユーザに対して付加サービスを提供するというのがあります。これはまじめに設定をしないと必ずセキュリティホールになるといっていいほど危ないものです。基本的にクライアントからの要求で何か外部プログラムを呼び出すような設定というのはどんなものでも注意が必要なのですが、特にこれらの場合、有名なセキュリティホールがありますので最新の注意を払って設定していただければと思います。まずは「要らなかったら消す、要らなかったら止める」というのがここでも基本の行動になってきます。作成するあるいは設定するという場合は良く注意して、攻撃を受けても大丈夫なように作っていただきたいと思います。

もう一つの対策としてアクセスログを取っておこうというのがあります。たとえば、何処のサイトでもアクセスログをよく見ると、phf アタックが試されていたということがあるみたいですが、時々アクセスログに危険なアクティビティがないかどうかというのを調査していただくのがいいかと思います。もちろんアクセスログを取っておけば前向きな目的にもいろいろと使えますので、結構どのサイトでもＷｅｂサーバなどでは比較的よく保存しておられるではないかと思いますが、あえてここで強調しておきたいと思います。

ここまでの対策は特定のサービスにおけるサーバ設定の話でしたが、逆にそのサーバを動かしているマシーンの方のセキュリティも固めておくというのも大事になってきます。例えばインターネット中のマシーンから遠隔ログインできる設定よりは、特定のメンテナンス用のクライアントからしかアクセスできない、さらにはコンソールからしかアクセスできない設定の

方がより安全になります。またこういう情報提供サービスというのは、とりあえずホストにこのパケットを送ったら落ちるかなといって安易に試されがちので、本来提供されているサービス以外のポートへのパケットを全部落してしまうといった防御方法も検討される価値があるのではないかと思います。またもっとそれを真剣にやるのであれば、ファイアウォールの後ろ側のサーバでサービスを提供し、必要なものだけをファイアウォールで通過させるという設定もあると思います。この場合、ファイアウォールとしてどういうものを持ってくるかによりますし、またどのようにサービスを中継するかという実装によっても違ってくるかと思いますが、もし自分のサイトのサーバが人気のあるサーバでしたらシステムのパフォーマンスに注意が必要かもしれません。

またこれと前述の対策と組み合わせて考えると、他のサービスと同居させていると情報提供サービスとして狙われやすいホストを守りにくくしてしまいますので、マシーンに余裕があればWebサーバはWebサーバ専用、Anonymous FTP サーバは Anonymous FTP 専用というようにそれぞれ専用のサーバ機へ少しづつ機能を分散させてゆくというのも一つの方法だと思います。これもサイトの事情によって判断が難しいかと思いますが、例えばサーバのマシーンを二台にしてしまいますと、二台分まじめにセキュリティ管理をやるという手間が増えます。その手間にかかる時間と人手を捻出できる状況ですと安全に運用できると思うのですが、そうでない場合、安全に保つのが難しくなってきますから、ここも少し判断が難しいところかと思います。あとは「ファイアウォールの背後に隠蔽」といいましたがこの場合良く出てくる単語に「DMZ(demilitarized zone)」というのがあります。ファイアウォールメーカがよく外と内側以外にもう一個セグメントを付けられますといった機能の紹介でDMZという用語を使っていますが、インターネットルータ側とは別にWebサーバ向けに別のセキュリティポリシーをもって配置できるという機能です。

・ WWW サーバー

WWWサーバというのは HTTP のサーバのことですが、その固有の事柄について説明します。

こちらの方の対策としまして、もう大分くどくなってきたわけですが、phf プログラムがまだ残っていたら[起動不可能にする]か、しかるべきセッティングにすることを強くお勧めしたいと思います。最近の届け出のメールを見ていますと真面目にカウントしているわけではないのですが、IMAP、phf、SPAMが大勢を占めているような印象があります。単にそれらが多いと思っているからそう見えるのかもしれませんが、とにかく多いです。まずは、phf は止めていただくというのをお薦めします。

基本的にCGIプログラムは本当に自分のところで使うものだけを置くように心がけるほうがより安全な設定のはずです。

一般的な httpd ですと cgi-bin というディレクトリがあって、そこに cgi プログラムを置いてゆくようなセッティングになっています。そのときの注意事項ですが、そのディレクトリには本当にWebサーバが呼び出す実行形式ファイルだけを置くようにします。特に絶対に置いていけないものの例としてよくあげられるものに、perl プログラムとか sh プログラムがあります。cgi-bin ディレクトリにスクリプトを置く場合、このディレクトリにスクリプトを解釈するインタプリタも一緒に置いてしまうというのはありがちな設定かもしれませんが、これをやってしまいますと直接 perl や sh そのものがが起動されてしまいます。そうしますとなんでも好きなことができるという状態になってしまい、とても危険です。これは結構前の CERT Advisory でも報告されている(CA-96.11)話ですが一度見直されてみたらいかがでしょうか。

・ Anonymous FTP

Anonymous FTP のほうは昔からセキュリティホールが多かったというのと、設定ミスを犯しがちで、きちんと設定するのが難しかったという感じがありましたので、そういうところを狙って攻撃をされていました。HTTP のない時代はともかく、今なぜWebサーバではなく anonymous FTP を設定するかというと、書き込みができるディレクトリを置いて手軽にファイル交換をしたいというのがあるかもしれません。ただそのディレクトリも繋げれば誰でも置けてしまうという点がありますので、悪用されるリスクは伴います。

こちらの方の対策もいろいろあるわけですが、anonymous FTP で有名なソフトウェアに wu-ftpd というのがあります。wu-ftpd も、最近少しずつセキュリティホールが修正されているようですので、お使いであればバージョンアップをしていただいた方が安全かと思います。最近の更新がWUアーカイブの方から、「academ 版」というのに移っていること、またβが長い間続いていてドキュメントでも紹介しにくいというような状態がありました。古いFTPデーモンは新しいものに変えていかれることをお勧めしたいと思います。またある版の wu-ftpd にはトロイの木馬を仕組まれて一時期出回っていたというような話が公になっておりますので(CERT Advisory CA-94.07)、もしもだいぶ前に wu-ftpd を設定したという方がいらっしゃいましたら新しいバージョンへのバージョンアップを検討されることをお勧めしたいと思います。

次に、書き込み可能なディレクトリを作りたい場合に、それをある程度安全に運用する方法を見ておきます。例えば書き込み可能なディレクトリを特定のパーティションに一個丸々

割り当てておいて、そのパーティションが潰されてもいいようにしておくとか、ファイルは毎晩掃除するようにしておくとかいろいろな方法があるわけですが、基本的には良く監視していただくということになろうかと思います。

単純にファイルを配りたいだけでしたら、anonymous FTP に必ずしもこだわることはないのではないでしょうか。もしもサイトの事情が許せば HTTP のサーバに切り替えてゆくというのも一つの対策かとは思います。その場合、例えば ftp コマンドで接続して mget とかできなくなって不便だと言われてしまえばそれまでですが、そういう形の守り方もあるのではないかと思います。

・ **WWW ロボットによる情報開示**

Webのロボットと言われるシステムによる情報の開示、不本意な流出について述べます。こちらも対症療法というものはあるにはあって、基本的に何処のロボットサイト、サーチエンジンをやっているところでも、「検索結果に出ては困るページがもし出ていたら連絡してください」というような連絡アドレスが公開されているようです。そこに連絡するのも一つの手です。しかし、もうすこし本質的な対策がありまして、「robots.txt」というファイルをドキュメントルートに置いておくと、そのファイルの指令に応じてこのディレクトリは読まないでくれとかこのディレクトリは選んでもよいとかの制御をすることができるようになります。次のＵＲＬに、robots.txt の書き方があります。

http://info.webcrawler.com/mak/projects/robots/norobots.html

ただし、一般的なサイトが使用しているロボットはこの robots.txt の指令に従いますが、従わないケースが世の中に出てきてしまうかもしれません。そういうロボットを使うサイトがどのような評価を世間から受けるかは別として、本当にそういう情報の管理に気を配るのであればファイル単位のアクセス制御を設定しておく必要があります。ただこれも問題があって、よくあるのがそこにアクセス制御をかけていたつもりなのにロボットにアクセスされてしまって情報が出てしまったということが起こるおそれもあります。そういう運用ミスを回避するためには運用上の工夫が必要になってきます。例えば、少しでも内緒の情報があるサーバは別に独立させて動かしておくとか、明らかに別の運用をしてそういう操作ミスを未然に防ぐといった対策をした方がいいかもしれません。しかしサーバを別に分けてしまうと、一方でそのサーバをメンテナンスして安全に保つという負荷が増えてきますので、やはりサイトの人手、体力、事情というものでどういうソリューションを選択するかは選んでいただきたいと思います。

・遠隔ログイン

多くのＵＮＩＸ系ホストで動いている遠隔ログインについて説明します。遠隔ログインは侵入口の有力な一つで、ユーザ認証さえパスしてしまえばそのまま入れるわけです。これはセキュリティ対策上非常に注意が必要になってきます。また、遠隔ログインのセッションは盗聴される可能性がありますので、アカウントやパスワードが盗まれる危険があります。

こちらも対策としては遠隔ログインできるホストを制限する、記録するとか「ネットワーク盗聴対策」を実施してアカウントやパスワードを盗まれないようにするというようなことが挙げられると思います。あるいは遠隔ログインをする標準の telnet、rlogin ではなくて暗号化通信ができるソフトウェアを導入するのも一つの方法かと思います。ただそれも簡単にコンパイル・インストールできるような代物かというと、そういうものもあったりそうでないものもあったりしますので、少し管理的な負担が増えるかもしれません。そういうものの例として、telnet over SSL をやるようなソフトウェアとか、あるいは SSH（セキュアーシェル、こちらのプロトコルは RFC でも公開されています）とかあります。SSH は最近よく使われるようになってきたアプリケーションですが、このように暗号通信をおこなうより安全なソフトウェアもありますので状況に応じて導入を検討されたらいかがかと思います。

遠隔ログインに関して古い話を一つ挙げておきますが、ＰＣ用の telnet クライアントの何種類かにはなぜかＦＴＰサーバの機能が付いています。もちろんＦＴＰサーバというぐらいですから telnet で他のホストにログインしているときに、telnet クライアントが動いているマシーン上のファイルを外からＦＴＰでアクセスできるという機能なわけです。それではそれはどうやって保護されているのかといいますと、パスワード認証が簡単に付いているか空のパスワードつまり認証なしか、いずれにせよそういうセッティングになっているはずです。そういうような機能というのはデフォルトでは動いていないというのが普通なのですが、うっかりとそれを動かしてしまうとローカルのＰＣのファイルが危ないという問題があって、これももう何年も前にひとしきり騒ぎになりました（CERT Advisory CA-91.15）。大抵のソフトウェアはデフォルトではこの機能は消えているはずですが、うっかりユーザが触ってしまったりとかそういう危なさを認識せずに使っていたりするケースがあると思いますので一応念のために申し上げておきたいと思います。

これは何年も前にあった話ですがというのを二つほどお話しましたが、古い話なのでこれらの対策はしなくてもいいのか、忘れてしまっていいのかというとそんなことはありません。何年か前の設定のままずっと何年も使い回しているマシーンというのは各地に散らばっているみたいで、私も学生時代にそういう何年も前のマシーンを見ていましたし、きっとその事情は変わらないのではないかと思います。そういうところですと、大昔の管理者がその当時安全と思われていた設定のまま、何年もメンテナンスされずに動いていたりするかもしれません。そうすると今の基準に照らすと危険かもしれませんし、今のユーザが今の基準で、もうそんな何年も前だから知らないという感じで、いつのまにかセキュリティホールがまた有効な状態になっていたりすると、もしかしたらそれもやられてしまうかもしれません。こういう形で、大昔のセキュリティホールを忘れていいかというと必ずしもそんなことはなく、もちろんソフトウェアのセキュリティホールに対しては新しいバージョンを入れればいいわけです。しかし、このように設定によって変わるものというのはいつまで経っても忘れることができないので、管理者の負担はどんどん増えていくわけですがそれに負けないように不正アクセス対策あるいはセキュリティ対策の方がんばっていただきたいと思います。

6. 各種サービスのセキュリティ

・メーリングリスト・サービス

メーリングリストをサービスする側のリスクと対策について説明します。基本的にメーリングリストサービスというのは、もちろんメールで使われるわけですのでメールに関連したリスクは被ることになります。ただ単なるメールサービスではなくてメーリングリストサービスの場合、参加者の数に比例してあるいはそれ以上に大変になってくるということが挙げられます。それに加えてメーリングリストをサービスするために使っているソフトウェアのセキュリティホールがリスクの重要な部分になってくると思います。またもちろんメーリングリストが電子メール爆弾に使われてしまうとかあるいはＳＰＡＭの送信に使われてしまうといったリスクもあると思われます。

それに対しての対策ですが、リストへの投稿をリストのメールが送られるアドレスに制限するというのが一つの方策だと思います。これですとあらかじめ一応確認の採れている人だけが投稿できるということになりますので悪用に使われる危険が少し減ります。さらに登録するときに自動的に登録するのではなくて、一旦リストへの参加要求のメールを送ってきたアドレスに確認のメールを送って、その確認メールに対して妥当な返事が戻ってきたら、その時点で初めてメーリングリストに登録するというようなやり方もあります。それによって悪用を防ぐことができます。このような参加確認する手続きをやっておきますと、知らない人にいつのまにかメーリングリストに登録されていて、急にメールが増えて困ったとかいうようないたずらに使われることを防ぐこともできますし、また投稿を登録アドレスのみに限定しているときに偽造アドレスを登録してその上で投稿というような悪用を防ぐことにもなります。

さらにリストサーバのバージョンによっては既知の弱点があるかもしれませんので、適宜バージョンアップをおこなっていただくというようなことが対策として考えられると思います。

・ Samba

Ｓambaというのは、UNIX系のシステム上のファイルシステムを Windows95 などからアクセスできるようにするというようなサービスですが、最近よく使われています。こちらの方はセキュリティ問題というのが時々見付かっていますが、他の理由もあって、よくバージョンアップされています。ですから、適切なタイミングでバージョンアップをしていただくことがより安全な運用に繋がります。また、適切に外界からのアクセスは遮断しておいた方が、(認証を破られるとサイトのファイルの情報が奪われる危険性もありますので) よいと思います。関連するパケットをファイアウォールで落しておくとかいろいろな方法があります。

Ｓambaサービスそのものとは少し離れますが、このＳambaサービスを動かすために、クライアント環境の方で NetBIOS over TCP/IP あるいは NBT といわれているものを使うように設定しないといけません。これを設定すると、そのクライアント環境がインターネット環境に直結しているＬＡＮ上にある場合、この NetBIOS over TCP/IP が有効になっているクライアント側を攻撃する方法がいろいろあります。使うなとか使ってもいいとかいうものではなくて、例えばこれも適切なレベルでパケットフィルタリングなどをすれば気にする必要もないかもしれません。もう一つ問題があり、こういうクライアントを使っていますと、一度パスワードを入力するとずっとそれをＰＣの側で覚えているというパスワードキャシュという機能があります。これは何度もパスワードを打たなくてもいいので便利ですが、ホスト側が用意している認証のパスワード機構とは別にパスワード関係のものを覚えているものが増えるという意味でセキュリ

ティに配慮している環境にはそぐわないのかなと思います。このパスワードキャシュの機能ですが、一応殺せるようにはなっていて、毎回パスワードを聞かれるように設定することはできます。簡単に言うと、直接レジストリを変更するか、システムポリシーエディターを使います。システムポリシーエディターの標準のセッティングでこのパスワードキャシュを無効にする項目があります。

・ r コマンド群

ＢＳＤ系のＵＮＩＸに由来する r コマンド群について説明します。r コマンド群と一言で言ってしまいましたが、そのなかには rsh、rcp、rlogin、rdist といった類のコマンドがあります。基本的な問題としては、古いバージョンにはセキュリティホールがあるということがあります。もう一つは r コマンドの特徴として/etc/hosts.equiv とか ~ /.rhosts というファイルを設定しておきますと認証をバイパスすることができます。これを適切に運用していれば逆にパスワード盗聴のリスクを減らせるというメリットもありますが、安易に設定していたり設定を間違えていたりしますとこれを使って攻撃をされてしまう可能性があります。たとえばホスト単位の認証をＤＮＳの偽造とか IP spoofing を使って認証をバイパスして侵入されるというリスクもあります。また/etc/hosts.equiv とか ~ /.rhosts のファイルを侵入者に読まれた場合、自分の持っているほかのアカウントについての情報源にもされてしまいます。これを使って他のホストに次々ログインしていくというのも昔からある手口ですので、実際に利用される場合にはある程度配慮が必要になってきます。また、rexecd というのがあります。これは他の r コマンド群とは少しだけ動作が違うデーモンで、たとえばX端末から窓を開くためにこのサービスが使われていたりしますが、このデーモンには昔からちょっとしたミスフィーチャーがあります。普通 Login:プロンプトに登録されていないユーザ名を打ち込んだ場合も登録されているユーザ名を打ち込んだ場合もどちらも同じ Login incorrect というメッセージが返ってくるわけですが、rexecd ではその両者のメッセージが違うためにシステムにどういうユーザ名があるかを probe することができます。すこし細かい話になってしまいましたが、そういうのが気になる方は止めておくのがいいと思います。

ほかの対策としては、アクセス制御をして特定のマシーンからしかアクセスできないようにしておくか、サービスそのものを止めてしまう。あるいは別のツールを導入するかというのがあると思います。前から紹介しています「tcp_wrappers」というのもありますし、「logdaemon」といって遠隔ログイン用のデーモン一式の置き換え版がパッケージされたツールもあります。また、r コマンドの中でも特にセキュリティホールが騒がれている rdist というものに関しても、そっくり入れ替え用のパッケージが出回っていますのでこれも適宜さがしていただければ見付かるのではないかと思います。あるいはまるっきり違うサービスで代替してしまうものの例としては「ssh」があります。ただのや、logdaemon のように標準のデーモンを置き換えるものや、サービス自体を置き換えてしまう ssh を導入すると標準の状態からかなりシステムの状態を変えてしまいます。特にコマーシャルベースのＯＳを使っている環境ですとかえって管理しにくくなる場合もあります。自分のサイトの事情あるいは自分の力量にあわせてどういう対策を取られるかをよく検討してください。

・ Sun RPC （NIS, NFS, ...）

いわゆる Sun RPC に関係する説明をします。よくあるのは NIS とか NFS とかの話です。

良く言われているトピックとしては、サイトの外から NIS のマップを取得できるとか、単純に rup サービスとか rusers サービスを使って情報が得られるというリスクがあります。あるいはサイト外からの情報取得ということであれば、finger デーモンなども同じようなリスクを持っているわけです。finger デーモンの場合は、大昔のインターネットワーム事件とかいうぐらいまでさかのぼれば、finger デーモンも止めましょうといっているところですがそちらの方はここでは割愛します。Finger も含めて、サイトの情報を取られて困る場合は止める、アクセス制御する、という程度にとどめたいと思います。

少し脱線してしまいましたが話をもどします。大抵のシステムではデフォルトでは動いていないとは思いますが、「rexd」というデーモンがあります。これが動いている場合、認証が甘いのでこれを使って任意のコマンドを実行されるおそれがあり、危険です。万が一、rexd デーモンが動いている、有効になっているようなシステムでは、たぶんお使いにならないとは思いますので止めてください。

NFS サーバの設定を間違えている結果、ファイルシステムがマウントされてしまうということがあるかもしれません。設定をよく見て正しくアクセス制限がかかっているような状態にしておいてほしいと思います。

以上の対策としては、パケットフィルタリングやアクセス制御をするというのもあります。この場合、ポートマップサービスのポート(111 番)を一つ潰しておくだけで RPC の一般的な攻撃は防げますし、ＮＦＳのポート(2049 番)を塞いでおくのも有効な対策だと思います。あるいは「portmap」や「rpcbind」というプログラムが動いていて、これらのプログラムというのはＲＰＣを利用するためには欠かせないデーモンとして動いているわけなのですが、これを置き換えバージョンのものに置き換えることによってＲＰＣのアクセス制御を加えることができます。portmap や rpcbind を止めてしまうと他にＲＰＣを使っているものはあらかた再起動しないといけなくなってしまいますから、稼働中のシステムで入れ替えるというわけにはいかず、少し手間がかかるかもしれません。

NIS については、/var/yp/securenets というファイルを書いておく対策があります。このファイルはＮＩＳサーバを運用している場合にＮＩＳサーバにアクセスコントロールをかけるというファイルです。ただし、このファイルの設定が有効なＮＩＳサーバ (ypserv) は最近のものにかぎられます。たとえば、昔の SUN OS4.1.x というのはパッチを当てないとこれが有効になりません。SUN OS4.1.4 ではパッチを当てなくても平気なはずです。他のＯＳでもパッチを当てれば有効になるものもあると思いますので、こういうのを書いてＮＩＳサービスを守るということも重要だと思います。

さらに、特定のＮＩＳの実装の話になってきますが、よくある設定として、ＤＮＳを使うためにＮＩＳを動かすという設定があります。もしもそのためだけに NIS を設定される場合には、本来それを動かすためには hosts.byname というファイルと hosts.byaddr というファイルだけをマップとして用意すれば大丈夫です。つまりパスワードなどをＮＩＳ管理下に置かなくてもかまわないということです。セキュリティに気を使いたい環境で、ＮＩＳをやめたいということであっても、hosts.byname と hosts.byaddr ファイルしかマップは置かないという体制に変えるだけでも効果があります。

・ＮＴＰ

ＮＴＰはインターネットで標準的につかわれている、ホスト間で時計を同期させるために

使われているプロトコルのことです。一応ＮＴＰという見出しにはなっていますが、それぞれ時間情報を一致させるサービスが他にいくつかあるはずですから、どれにも通用するノウハウの話をしておきます。

まず、これには偽のサーバあるいは偽の情報をＮＴＰサービスに送り付けることによってマシーンの時間を混乱させるというリスクがあります。そうやって時間情報が狂うとどういう不都合があるかと言いますと、例えばログの記録の時間が狂うので後から調査がし難くなるということがあります。もっと高度な話になりますとタイムベース認証をごまかすという操作にも使えます。「タイムベース認証」とは、ある時間の間だけ認証が通った状態になるというような種類の認証とかあるいはある時間に認証を通すためにはこのパスワードを使ってという形で時間に応じて認証情報を変化させるような認証方式のことだと思ってください。例えばある一定の時間内にこの情報を送れば認証が通るというような場合、前に他の人がそれを送ったのを見て同じことをするには時間が過ぎているからできなくても時間を狂わせてそれをもう一度試みると認証が通るというような不正アクセスができるようになります。

これに対する対策ですが、こういう時刻サーバに対して何らかの認証プロトコルを設定してあげるということがまずあります。ＮＴＰの場合は何種類か認証方式がありますので、それで自分の管理しているサーバ同士であるいは知人が管理しているサーバとでお互いに認証しあってそういう偽の情報を送られても平気な状態にすることができます。また不必要なマシーンがそういう情報を送り付けてこないようにアクセス制御をおこなったりあるいは一般的に非常に有名で信頼のできる時刻サーバをたくさんサーバとして指定しておいて、そうすればすべてが同時にやられることはないだろうから結構安定して時間情報が得られるのではないかというようなやり方もあるはずです。

・ X Window System

X Window に関してはサーバ側、クライアント側とも、多数のセキュリティホールが発見されています。X Consortium リリース(かつてのMITリリース)に含まれているクライアントやライブラリには、かなりセキュリティ関係の修正が行われ、セキュリティホールが埋められていますので、なるべく新しいバージョンのソフトウェアを使うことが大事です。とくにxterm、kterm、xload のようなクライアントは、set-uid,set-gid されてインストールされています。また、シャドゥパスワードを使っているシステムですと xclock というプログラムも set-uid root されています。現実に、そういうプログラムに関連した更新が行われています。適宜最新のものにバージョンアップされるか、もし使っていなければせめて set-uid とか set-gid をはずしておくと良いと思います。

プログラムそのもののセキュリティホールとは別に、認証をおこなっていないサーバに対してXクライアントを接続してそれでセッションを盗聴するというようなことが昔からできると言われていました。こちらはユーザ認証を設定するか、接続できるクライアントのアドレスをサーバに設定するつまりホストベースの認証を設定するかというような対策ができると思います。また普通のXサーバ単体ではやりにくいのですが、もしどうしてもある程度の範囲に対してX Windowの接続を提供していないといけないという場合に、ファイアウォールなどで中継したときの記録を取っておくとかというようなことを、あるいはそのレベルで何らかの別の方法で規制するということも対策としてあげられるのではないかと思います。

これまでいくつかサービスを取り上げて説明してきましたが、どれもサービスがあればセキュリティホールが見付かっているので新しい版にしましょうという話になります。また他のリスクがあったりという形で、基本的にサービスを増やせばリスクが何かしら存在するものだということになってきます。大切なのは、大抵の場合はそれに見合う対応策などがありますのでサービスを導入する前にそれに対してどう対応するかというのをあらかじめよく検討の上、場合によってはリスクが高いという判断があれば、サービスを止めたりあるいはこういう対策をすれば防げるというのがあればいろいろな対策をおこなっていただいて、その上でサービスを設定していただくということに気を付けていただきたいと思います。

7. 非常事態への対応

ここでは問題が起こったときに、どのように対応していったらよいのかということに関して説明してゆきます。

・監視

まず非常事態が発生していることに気が付かなかったら対応することもできませんから、やはり常々自分のサイトが攻撃にさらされていないかどうかということをチェックすることが大事になってきます。

そのためにはまず、通常のシステム状態あるいは通常のハードウェア、ソフトウェアの稼動状況というのを把握しておく必要があります。その中には例えば本当の正規のユーザが通常どおりの作業をしているかというようなことも平時の通常状態に含まれています。例えば普段昼間にしかログインしていないユーザが、午前5時頃にログインしたとかという記録が残っていたとすると、普通はそこでそのユーザのアカウントが盗まれたのではないかと気が付きたいところです。ただそれもそのユーザが普段昼間しかアクセスしていないという事実を知らないとそういう判断もできませんし、ましてやログを見ていないと何があってもわからないということになります。

やはり常日頃から正常なシステムの状態を把握しているということがまず第一歩になってきます。把握していますと、急に負荷が増えたので何か悪いことが行われているかもしれないということにも気が付きますし、あるいは単に調子が悪いと思ったらハードウェアが故障したということにも早めに気が付けるかもしれません。いずれにしても常日頃から状態を把握してその上で何かシステムがそれから外れたことがあったら、それに対して注目して確認をするというのが大事です。

調査の第一歩としてアクセスの記録を取りましょうと前に何回か述べてきましたが今度はその記録を具体的にチェックしてゆくということが大事になってきます。これは本当に大変な作業です。これまではなるべく詳しくログを取りましょうというお話をしてきましたがそうやって詳しく取ったログというのはえてしてセキュリティとはまったく関係のないログで埋め尽くされていたりします。それではどうしたらいいのかということですが、例えば一日一回そういう問題のないエントリーをフィルターで取り除いて注意したいエントリだけを残したものを管理者にメールで送る仕組みを作るとか、そういう運用上の工夫で対応してゆけます。ログを詳しく見ないといけない状態があれば、そういうフィルターをした結果のログにも前兆が出てくるはずです。そのあとで、詳しく取っておいたログを分析すると異常事態が発見できるというようなストーリーになります。自分でそういうツールを作ってもいいわけですし、既にあるものを利用していただいてもかまいません。こういうツールはいろいろと出回っています。例えばWebサーバですと少しずれた使い方ですが、アクセス統計ソフトとかいうものを使ってログを分析すると、phf にアクセスがあったというのが分かる場合があります。そういうツールを使ってできるだけ負担のかからない方法で、しかしそれを継続してシステムの稼動状況に注意を払っていただきたいと思います。

・ IRT、他サイトからの連絡

そうやって自分でログをみて見つけた異常とは別に、他のサイトあるいは我々のような緊急対応組織(IRT)から連絡が来る場合があります。たとえばJPCERTが連絡を差し上げる場合には「こういう情報が寄せられています。もしかしたらあなたのサイトが悪用されている

かもしれませんので早急にご調査頂くようお勧めいたします」というような形の文章を送信しています。ちなみに宛先は、大抵ドメイン名を元にJPNICデータベースのほうを検索しております。大規模な組織になりますと、部門間の対立などがあって、そこに書かれている連絡者に連絡を取られると困るとかいうようなことがあるのかもしれませんが、さすがに我々はそこまで細かく対応いたしかねます。

こういう連絡が入った場合は、慌てないで冷静に対応してくださるようお願いします。こういう連絡が入った場合それががせネタであったということも十分にありえることです。意図的にそういう欺まん情報を流す場合も有りますし、あるいは何かログを改ざんされた結果勘違いしているのかもしれませんし、いろいろな原因があります。ただ、こういう攻撃元がありましたという連絡をJPCERT／CCが受けたとしますと、それが真実であった場合の損失ということを考え、連絡を中継することにしております。いずれにせよ、冷静に事実関係の調査からスタートしていただきたいと思います。特に他のサイトから攻撃をされているとか攻撃をしているとかという連絡を受けた場合、そのサイトのマシーンは踏み台として他のサイトへのアタックに使われている恐れも十分にありますので適切な対応を取られるようにお願いしたいと思います。

・攻撃者の追跡について

被害を受けたので何とかしてこの侵入者、攻撃者を捕まえて恨みを晴らしたいと思われる方も結構いらっしゃるようですが、ここで攻撃者を追跡するためには何が必要かをここで少しお話しておきます。

攻撃者を追跡するためには、例えば追跡をたやすくするために、他のサイトをアタックしている攻撃者を放置して監視するということをよくやるわけですが、それをやるともちろん被害が拡大したり、その間にやられたサイトから苦情とか抗議とかが来たりというリスクを負うことになります。

一方、こういう追跡を成功させようと思ったときに何が必要かというと、例えば運がかなり効いてくるのではないかなと思います。あるいは、交渉力、技術力、労力、時間とかいろいろあります。基本的に今の日本の枠組みですと、警察が乗り出さない限り、この侵入者を捕まえたいので協力をしてくださいとか協力をお願いしますというのは、お互いにお願いをするあるいはお互いに善意でそれに応じるということでしか成り立ちません。ですから、そういう意味で交渉力とか運とかいうのは絶対欠かせないでしょう。またその上で順番にトレースしてゆくためにはしかるべき不正アクセスあるいはインターネットに関する技術、知識がないと辛かろうと思います。

そういうことで、実際にそれぞれ被害を受けたところが攻撃者を追跡して犯人を特定して捕まえたいと思っても、なかなかリスクの方が大きいのではないでしょうかというのが現状だと思っています。JPCERTのほうでは、攻撃元サイトや攻撃先サイトに連絡をとるのは攻撃者を捕まえるためではなく、対策を取って再発を防止していただくという目的でやっております。個別に連絡したいがちょっと間に立ってくれといわれれば、こちらの労力の許す限りでお手伝いしております。

もしもうちのサイトにとっては犯人に損害賠償を請求することがとても大事だということでしたらがんばって追跡していただきたいと思います。ただ、「カッコウはコンピュータに卵を産む」とか「テイクダウン」とかを見てあのとおりにやろうと思われる方は結構いらっしゃるとは思

いますが、あれは本当にもう大変なことです。捜査機関とかの協力もあって行われたお話ですのでなかなかあのとおりには行かないのではないかというのが正直なところです。

・状況の把握

怪しいと思ったら、まず具体的にどれくらいやられているのか、どういう状況なのかを把握してゆくところから始めるのが重要です。その時にもしも組織の標準的なセキュリティポリシーとかそれに基づいた運用マニュアルというのがあればまずはそれを参照しながら行動を取ってください。そのなかには技術的な調査手順、対応手順、復旧手順そういうのがいろいろ書かれていると思われますし、対外的な手順、例えば報道関係、あるいは会社ですと顧客関係、株主関係とかそういうような対策というのが手順に含まれているかもしれません。

どうしてこういうことが大事かといいますと、攻撃されたものが外界から見える状態になっていたり、どこかに載せられたりということになりますと、情報システム部の担当者やシステム管理者が独自に対応した結果、被害が拡大したり、再発したりということになりますと立場上辛いわけです。そういうときに組織の標準的な手順があればそれにしたがった結果結局こうなってしまいましたということもできるということでもあります。もしもポリシーやマニュアルが無いようであれば、早急にしかるべき責任者の方と相談の上随時対応していかれた方がよろしいかと思います。

そうやってマニュアルを見ながら行動してゆくわけですが、基本的な対応の路線というものはあるわけで、それを次に紹介してゆきます。大抵不都合なことがあると攻撃を受けたのでないかといって慌てた状態になりますが、実際にはただのシステムの障害だったりあるいは操作ミスだったり事故だったり、そういうこともあります。まずは落ち着いて本当に攻撃をされているのか侵入をされているのかというあたりから調査を始めます。どうも侵入を受けているらしいということになると、今すぐ行動しないといけないか否かというのが重要な判断ポイントになってきます。例えば、侵入者が今すぐシステムを壊そうとしているとかいう状態になっていたりしますと、有無を言わせず侵入者のプロセスを殺したり、ネットワークからケーブルを抜いたり、そういうことをやらないといけないかもしれません。が、基本的には侵入をしてまだ侵入者が活動しているようでしたら「取り合えずこのシステムは踏み台として使うために壊しはしないだろう」という程度の認識は持っていいと思います。うかつに侵入者を刺激するような行動、例えば発見したというメッセージを相手の端末に送ったりとかそういうようなことをしますと、それまではシステムに壊滅的な打撃を与えてなかった侵入者がその瞬間にシステムを壊して逃げて行ってしまうというようなこともあります。そういうような意味で、どのような対応をするかどう行動するかというのは判断をしながら進めていく必要があります。

・一般的な対応

一般的な不正アクセスを受けたという状況ですと、最初の対応としては組織の責任者とか不正アクセス対応担当のエンジニアあるいは担当の人というのに連絡をするというのが最初にくる手順かもしれません。つづいて責任者や担当者の監督の元で、あるいはご自分が責任者や担当者であれば自ら、被害の拡大を防止するような行動を取ります。

基本的には、この時点でネットワークからケーブルをはずすとか電源を切るとかその類のことをやります。

この時に大事なのは、「不正アクセスの証拠をなるべくたくさん残しておく」ことです。例えば、どんなホストとネットワーク接続があったとか、どんなプロセスが動いていたかとかそういう情報は後で被害状況を調査するときにきわめて重要な参考情報になります。例えば、知らないホストからネットワークアクセスがあったという場合ですと、どこのホストから侵入をおこなっているというような判断ができますのでこの時点でなるべく取れるだけの記録を取ります。

この時に侵入者が残していくファイルとかが、今後関連する踏み台にお互いにされているサイトの調査に役立ったりとか、アンダーグラウンドの活動状況の把握に役立ちますのでさまざまな証拠として今後役に立つと思います。このため、ディスクなどの記録の内容のバックアップを取っておかれることをお勧めしたいと思います。

この場合バックアップと一口で言ってもいろいろな取り方があると思います。例えばファイルに最終参照時刻が記録されるようなファイルシステムを持っているシステムがあります。そういうシステムですと、普通のアーカイブコマンドでコピーしますと、侵入者がどのファイルをいつ見たかというのに役立つ最終参照時刻の情報が失われてしまいます。単純にディスクの内容をそのままテープにコピーしてしまうとかいろいろ方法は有りますが、なるべく多くの情報が残るようなバックアップを作成しておかれることをお勧めします。システムによっては呼び出し禁止モードでファイルシステムをマウントしておくと、そういう情報が残るものもありますから、運用でも工夫することができると思います。

・攻撃元サイトへの連絡

もしも自分のサイトが踏み台にされていて、さらに攻撃先も分かっている場合は、ぜひ攻撃先サイトへも連絡をしてください。見出しは一応攻撃元サイトだけになっていますが。

基本的に攻撃元サイトは、踏み台にされている可能性がある、というよりも、むしろその可能性の方が高いのです。あるいは踏み台というより、ダイアルアップ接続サービスの個人アカウントを盗んだという場合も有りますが、いずれにしても攻撃元サイトの管理者=悪者という立場で臨まれると話がこじれる可能性があります。基本的には「踏み台にされているかもしれませんので調査してください。こちらからは一応ご連絡さし上げます」というような形でご連絡いただければと思います。

注意事項があります。自分の組織のプライバシー、機密を他のサイトにわざわざ教えてしまうこともないわけですから、その中に記載される内容には十分ご注意ください。他のサイトにどういう場合に連絡を取るか、あるいは何処までを機密として何処まではこういう場合に攻撃元サイトに開示するかという判断基準についてですが、これも理想的な状態ですとやはりセキュリティポリシー、運用マニュアルといった形でサイトのなかで文書化されていることと思います。そういうのがない状態ですと、何処まで出せるかともめているうちに時間が過ぎてしまって効果が薄れてしまうかもしれません。できれば不正アクセスを受ける前にこういう不正アクセスを受けたときにどうするかということを想定して対応手順を考えておかれるようにお勧めしたいと思います。攻撃元サイトへの連絡、攻撃先サイトへの連絡のときに重要な注意として、攻撃者、侵入者がその状況を何らかの方法で盗聴あるいは監視している可能性があります。特にメールシステムに侵入を受けているばあいはご注意ください。

・ IRT への連絡

攻撃元、攻撃先とは別にわれわれJPCERT／CCのような緊急対応機関に対してご連絡いただければ幸いです。私たち私たちの機関は、基本的にはたくさんのサイトから報告が寄せられているセキュリティホールについて、インターネット全体に対してこれに気を付けてくださいという勧告文書を出したり、複数のサイトが関連する不正アクセスに対してサイト間の秘密を維持しながらお互いに連絡を取るお手伝いをするというようなことをやっております。また、不正アクセスを受けたところに対して簡単では有りますが、一般的な対応策とか参考文献の場所などをお教えしております。

攻撃元、攻撃先というのは直接連絡を取り難いところも有りますが、そういう場合お互いに連絡を取り合って穴を埋めていくことが将来自分のサイトに踏み台を経由したアタックがもう一度来るというようなことを防ぐ原動力になります。なるべくインターネット全体のセキュリティが向上するように、私たちの活動にご協力いただければ幸いに思います。

組織によって、私たちのような機関に情報を出せる出せないというのはあるとは思います。私たちに連絡を取った結果、社内で気まずい立場になったとかいうようなことがあってもひっかかるところですから、組織のポリシーに注意していただければと思います。

・復旧作業

関連のサイトと連絡を取りつつ復旧作業をおこなっていくわけですが、ここまでの調査で被害状況は把握できていると思います。

この時に具体的にどういう侵入経路で侵入を受けたかというのをある程度特定しておかれることが大事になってきます。この時に侵入経路が特定できていれば、それに対して再発の防止をおこなっていくという形で復旧をしていただければ同じ攻撃を今後受ける心配がなくなっていくわけです。そうでない場合は闇雲に対応を行わないといけないため、結構大変です。できるだけ何かあった場合のために普段から証拠になるようなログをたくさん取るようにしていただいたり、そういう準備をしていただくというのがここでは大事なのではないかと思います。

被害状況を把握した後、大抵はバックアップからシステムを回復したりとかOSを配布メディアからインストールする作業をやることになります。その時に特にバックアップからシステムを復旧する場合は、そのバックアップの中身がすでに不正アクセスを受けていないかどうかをよく確認してください。例えば、既にバックドアが仕込まれているシステムをバックアップから復元してしまっては、何のための復旧作業なのかわからなくなってしまいます。配布メディアとの比較とか配布メディアからの復旧を優先させたりとか、いろいろ方法は有りますが、そういうところには注意してください。

システムを回復した後で今回やられたセキュリティホール、侵入経路を塞ぐということをやっていただきます。また、あるいはパスワードファイルが盗まれているような恐れがある場合はパスワード変更をするとかそういう一般的な対策が必要になってきます。

・アフターフォロー

普通システムの動作が中断すると復旧作業が優先になってしまいますので、なかなかその後の反省というところには手が回らないと思いますが、復旧作業が一段落したらもう一度反省をするというチャンスを設けていただくのも今後の不正アクセスを防止するのに効果的だと思います。

保存した証拠に基づいて被害状況の詳細な調査を行う作業や、組織に対してのレポートを作成する作業がこのあたりにくると思います。たとえば、不正アクセスによって単純にシステムが破壊された以上の被害を万が一受けてしまった場合などは、そういう調査分析レポートが要求されることでしょう。

アタックに関する記録とかその後の再発防止策についてドキュメンテーション、記録をして今後システム管理を引き継ぐ方に記録として残してゆくのが良いとされています。また侵入を受けたということは、やはり運用上まずい点があったということですから、ハード、ソフトにとどまらず運用の方もシステムの一環としてここでよく見直しておいていただきたいと思います。間違っていたのが組織のセキュリティポリシーだったのか、それに基づいて作った運用マニュアルだったのかあるいは何か設定ミスをしていたのか、例えば、何か設定ミスをしていたとして、なぜあの運用体制で設定ミスに気がつかなかったのかということを見直してゆくうちにポリシーを修正するのがいいのか、運用を修正するのがいいのか、実際にマニュアルには準拠していたが人間が守っていなかった、じゃどうすれば守れるようになるのか、というようなあたりをすべて反省するいい機会にしていけば不正アクセスの損失からも、上手に立ち直れるのでないかと思います。

8. まとめ

現実に不正アクセスは存在します。あらゆるサイトが不正アクセスを受ける可能性があります。不正アクセスを受けることが多いサイトはあっても少ないサイトはなく、必ず一定量の不正アクセスを受ける可能性は存在しているという認識が必要になります。心当たりの有無にかかわらず、絨毯爆撃的に攻撃をされる可能性があるので、すべてのサイトで適切な対策を取ってください。ソフトウェアのバージョンアップとかアクセス制御といったあたりの対策を中心にうまくセキュリティホールのパッチ当てとかを運営体制に盛り込んでいただきたいと思います。

・情報提供のお願い

どんなセキュリティー問題が実際に存在し、どういうセキュリティに対するアタックが流行っているのか、どういう警告を出せばいいのか、というようなことはすべて皆様方からの届け出にかかっております。このため、是非JPCERTまでご連絡をお願いします。特によく聞かれることですが、「うちのサイトにはアタックが失敗しているログだけ残っているが報告していいのですか、そういう場合はしないほうがいいのですか」というようなことをよく尋ねられますが、そのような場合も是非ご連絡をください。一般的な設定ですと不正アクセスに成功するとログは残らないが、失敗するとログは残るとかいうような、そういうセキュリティホールも実は世の中にはあるわけですが、そのような場合、攻撃に失敗しているというサイトがたくさんあれば、やられているからこちらも緊急報告を出そうかというような対応を取ることができます。実害がない場合には、単に私たちにご協力頂くという形になってしまいますので少し心苦しい点はありますが、是非ご一報くだされればと思います。

届け出フォームはつぎのURLにあります。

http://www.jpcert.or.jp/form.html

フォームにはおおむね次の項目を埋めていただくことになっています。

1.不正アクセスを受けたサイト
2.あなたの連絡先
3.影響を受けたホストの情報
4.不正アクセスの内容

9. 参考情報

・ JPCERT/CC 提供情報

- Web

  http://www.jpcert.or.jp/

- 情報提供用メーリングリスト（登録方法）

  http://www.jpcert.or.jp/announce.html

  これは私たちから緊急報告や技術メモといった不正アクセスに関しての情報を送る場合のメーリングリストです。投稿はJPCERT/CCのものに限定されていますので、これに登録すると危ないということがないように十分心して運用しておりますのでぜひご登録く

ださい。

- 参考文献リスト

  http://www.jpcert.or.jp/ref.html

  どうしてもセキュリティ関係の文献というのは、実際よりも以前の状況を元に書かれたものばかりになってしまいますが、だからといって重要性がないかというとそんなことはなく、どうしてもここでご説明し難いようなセキュリティに関する網羅的な情報とか教科書的な考え方が記述されておりますので一度はどれかお読みになっていただくことをお勧めしておきたいと思います。

- 初心者のためのセキュリティ講座
  http://www.jpcert.or.jp/magazine/beginners.html

  これはインターネットマガジンさんのご厚意で、一般ユーザ向けに連載したものをＰＤＦファイルで提供しているものです。

・ FTP ミラー

FTPのミラーサイトを以下に何個所か設けています。

ftp://ftp.jpcert.or.jp/pub/組織名

というＵＲＬで統一しています。日本国内からアクセスしやすいのではないかと思います。

- CERT/CC

  ftp://ftp.jpcert.or.jp/pub/cert/

- AUSCERT

  ftp://ftp.jpcert.or.jp/pub/auscert/

- CIAC

  ftp://ftp.jpcert.or.jp/pub/ciac/

・ツール

セキュリティに関するツールのアーカイブがあるＵＲＬを以下に示します。

- CERT/CCアーカイブ（ミラー）

  ftp://ftp.jpcert.or.jp/pub/cert/tools/

- CIACアーカイブ

  http://ciac.llnl.gov/　にある（Tools）というリンクを辿る

- COASTアーカイブ

  http://www.cs.purdue.edu/coast/archive/index.html

  こちらは昔からセキュリティ関係のアーカイブの集積として有名でしたが、最近は置かれているもののバージョンが古いという感じがあります。

・ Assigned Numbers

これは例えばログの中に tcp の何番にアクセスがあったという場合に、これは何だろうと思ったときに調べるためのリファレンスです。もともとはＲＦＣとしてリリースされていて、その最新のものがＲＦＣ １７００ですが、RFC はかなり前の古い版です。最新情報を希望する場合は以下を参照すれば見付かるはずです。

http://www.iana.org/iana/ (General Assignments)
ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers

・ RFC

セキュリティに関するＲＦＣのＮＯ. を以下に示します。

- Internet Draft
  http://www.ietf.org/ (Internet-Drafts Index)
  “Site Security Handbook”
  “Users' Security Handbook”

- RFC 2196
  “Site Security Handbook” 現行版

  セキュリティポリシーの話もかなり取り上げられていまして、組織のセキュリティに責任をもつ人向けのものです。

- RFC 1281
  “Guidelines for the Secure Operation of the Internet”

  サイトのセキュリティ管理者には必ずしも全部当てはまらないような題材で構成されていますが、一応参考として挙げました。

以上