セキュリティ事始め

OpenSUSEユーザー会
杜若　桔梗

Agenda

- 自己紹介
- ネットワークセキュリティ

自己紹介

- 某ソーシャルゲーム会社の社内ネットワークをみていたエンジニア
- ホーム言語はDelphi
- 好きな分野はOSとハードウェア
- セキュリティ関係を調べたりいじるのが好き
- 現在絶賛お仕事募集中(正社員で)
- Twitter:akiha_tohno
- Blog: http://blog.geeko.jp/author/ciel

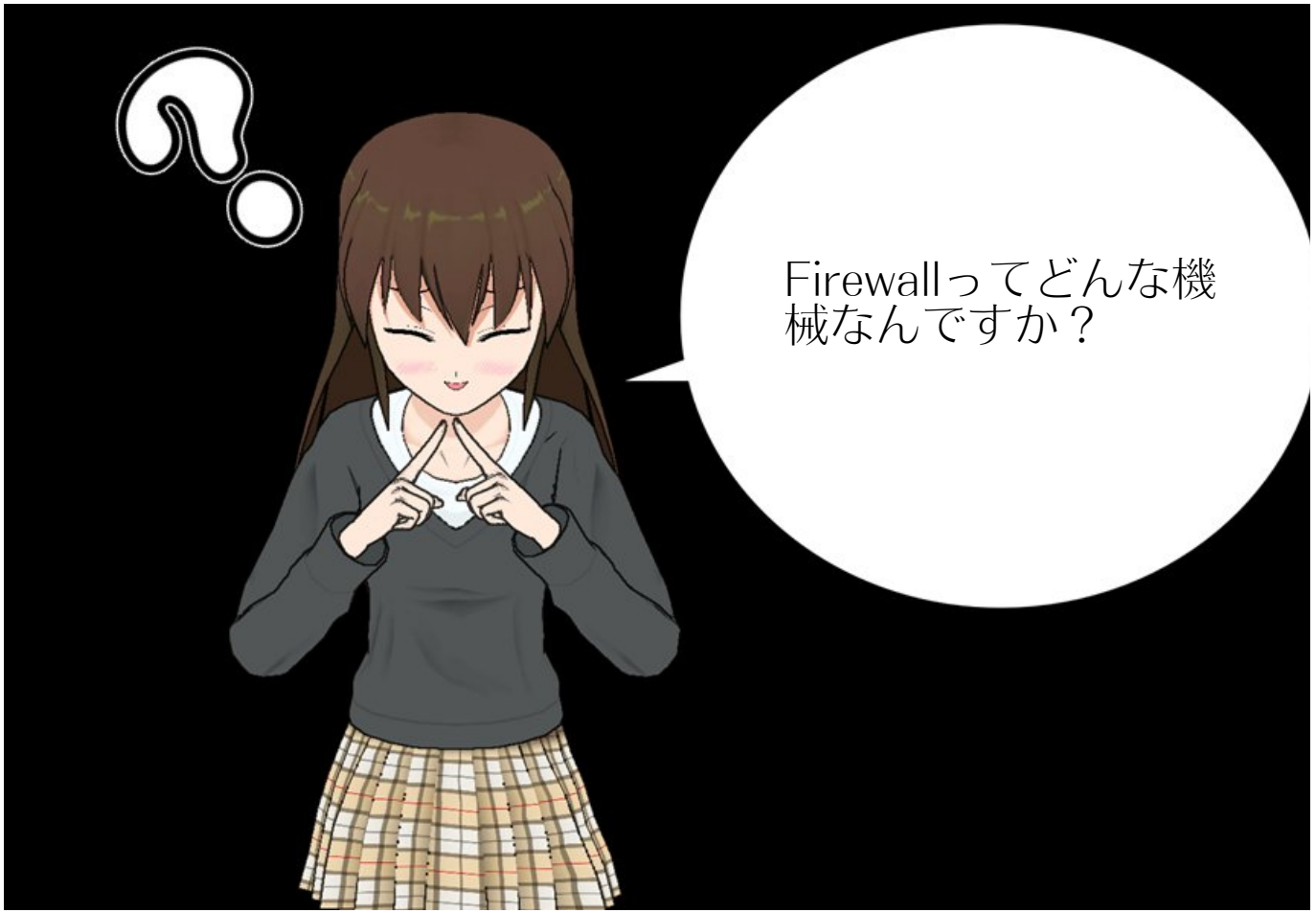
とある現場にて・・・

セキュリティとは？

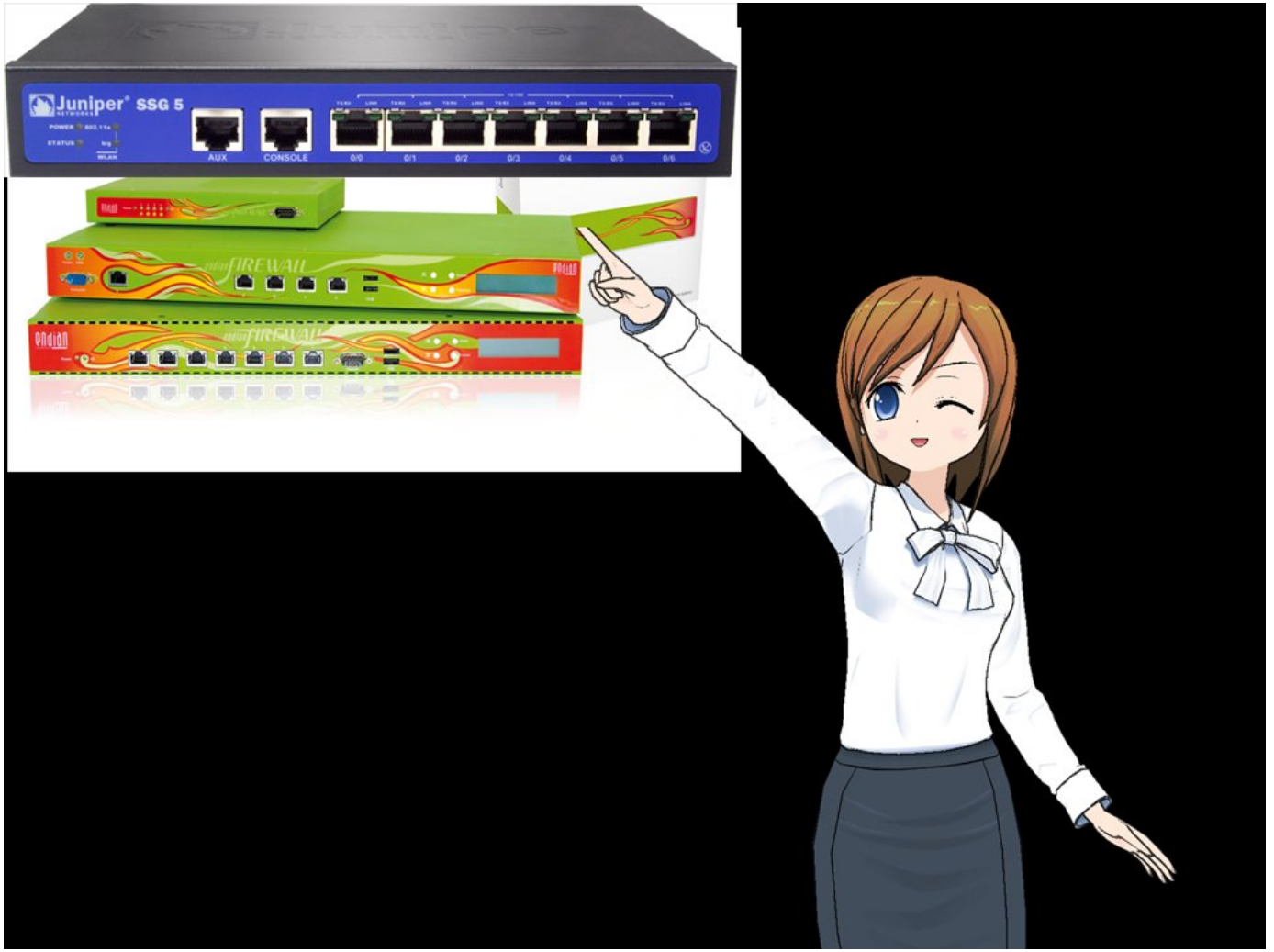
- どのようにして重要な物を守るか？
- OSCなので、どのようにして必要な情報を守るか？
- どのようにしてサイバー犯罪の被害者とならないようにするか？

対策：Firewallの設置

- 内部から外部のアクセスはNATを使う
- 外部から内部にアクセスさせない
- 外部公開サーバーはDMZへ設置する


Firewallってどんな機械なんですか？

Firewallの例


Juniper SSG 5
AUX
CONSOLE
0/0
0/1
0/2
0/3
0/4
0/5
0/6
FIREWALL
FIREWALL

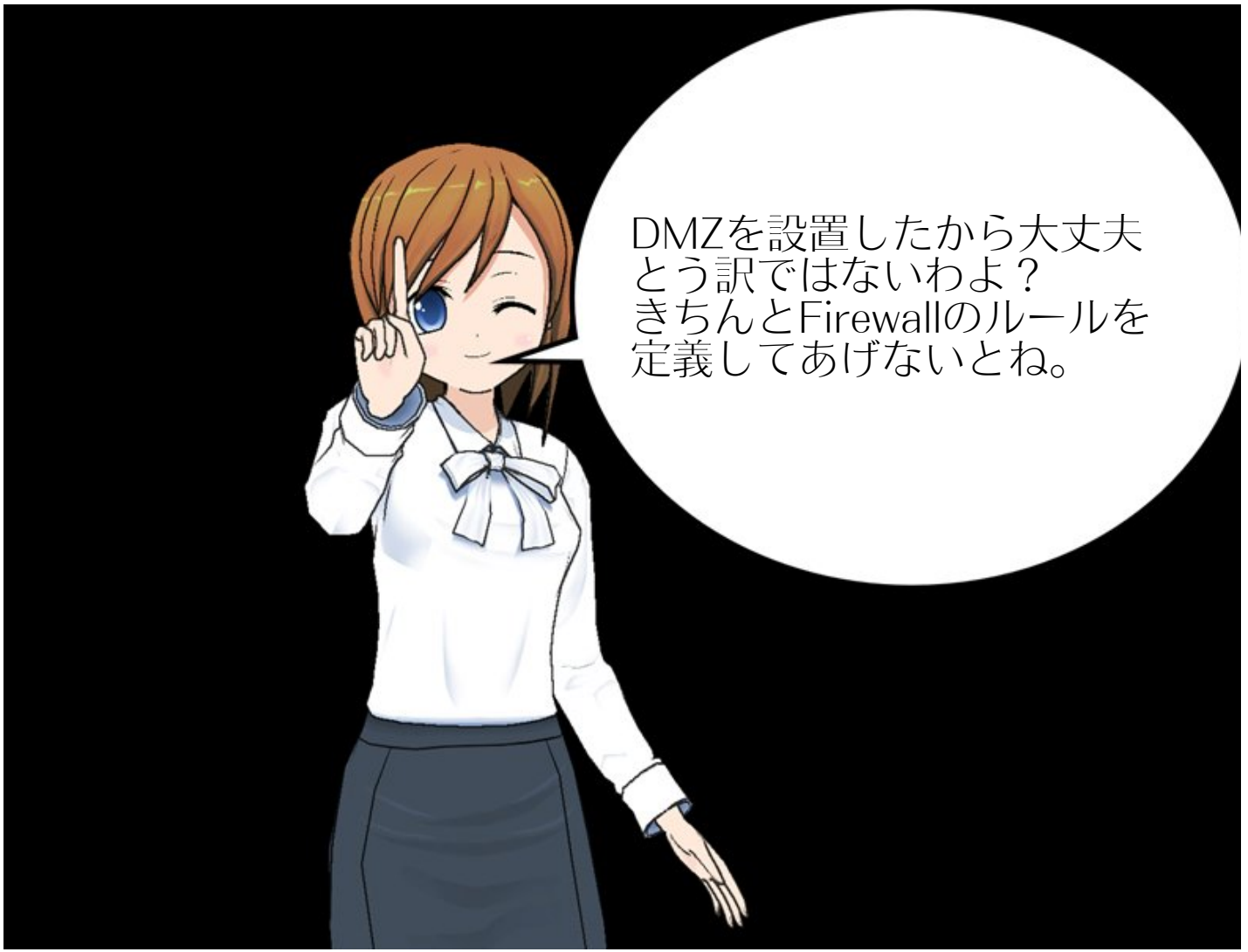
Firewall（DMZ構築のすすめ)

- 役割は単なる防火壁(侵入防止)ではない。
- 外から中へのアクセスをNATで処理することにより、相手に実IPを探らせない
- インターネットに直接アクセスさせない
- 内部ネットワークを隠蔽できる

DMZ設置機器

- Web Proxy Server(EX:Squid,Delegate)
- DNS cache Server(EX:Unbound)
- SMTP Server(EX:Postfix,EXIM)

で



アクセス権

送信元/送信先	イントラネット	DMZ	インターネット
イントラネット	-	○	×
DMZ	×	-	○
インターネット	×	○(一部)	-

Firewallのルール例(Internet直結)

ID	送信元	送信元ポート	送信先	送信先ポート	動作	変換元	変換先
1	Any	Any	Any	http,https	Allow		
2	Any	Any	Any	Domain	Allow		
3	Any	Any	Any	SMTP	Allow		
4	Any	Any	Any	POP3	Allow		
5	Any	Any	Any	IMAP	Allow		
6	Any	Any	Any	FTP	Allow		
7	Any	Any	Any	Any	Deny		

Firewallのルール例(DMZ有)

ID	送信元	送信元ポート	送信先	送信先ポート	動作	変換元	変換先
1	Intra	Any	Proxy	8080	NAT	Proxy	DMZ Proxy
2	Intra	Any	dns	Domain	NAT	dns	DMZ dns
3	Intra	Any	mail	SMTP	NAT	mail	DMZ mail
4	Intra	Any	pop3	POP3	NAT	pop3	DMZ pop3
5	Any	Any	Any	IMAP	Allow		
6	Any	Any	Any	FTP	Allow		
7	Any	Any	Any	Any	Deny		


なるほど・・
で、先生・・
サーバーサイドのセキュリティはどうやるのですか？

サーバーサイドセキュリティ

全般

- Windowsに限らず、脆弱性解決のためのパッチ(パッケージ)は導入する。
- 余計なサービスは起動しない
- 被害を最小限にするためにSE LinuxやAppArmer、TOMOYO Linuxを使おう。

アップデート

- Windows Update(Microsoft Update)
- %sudo apt-get update&&sudo apt-get dist-upgrade(Debian,Ubuntu etc)
- #yum upgrade(Fedora,RHEL etc)
- YaSTからアップデートする。(openSUSE)

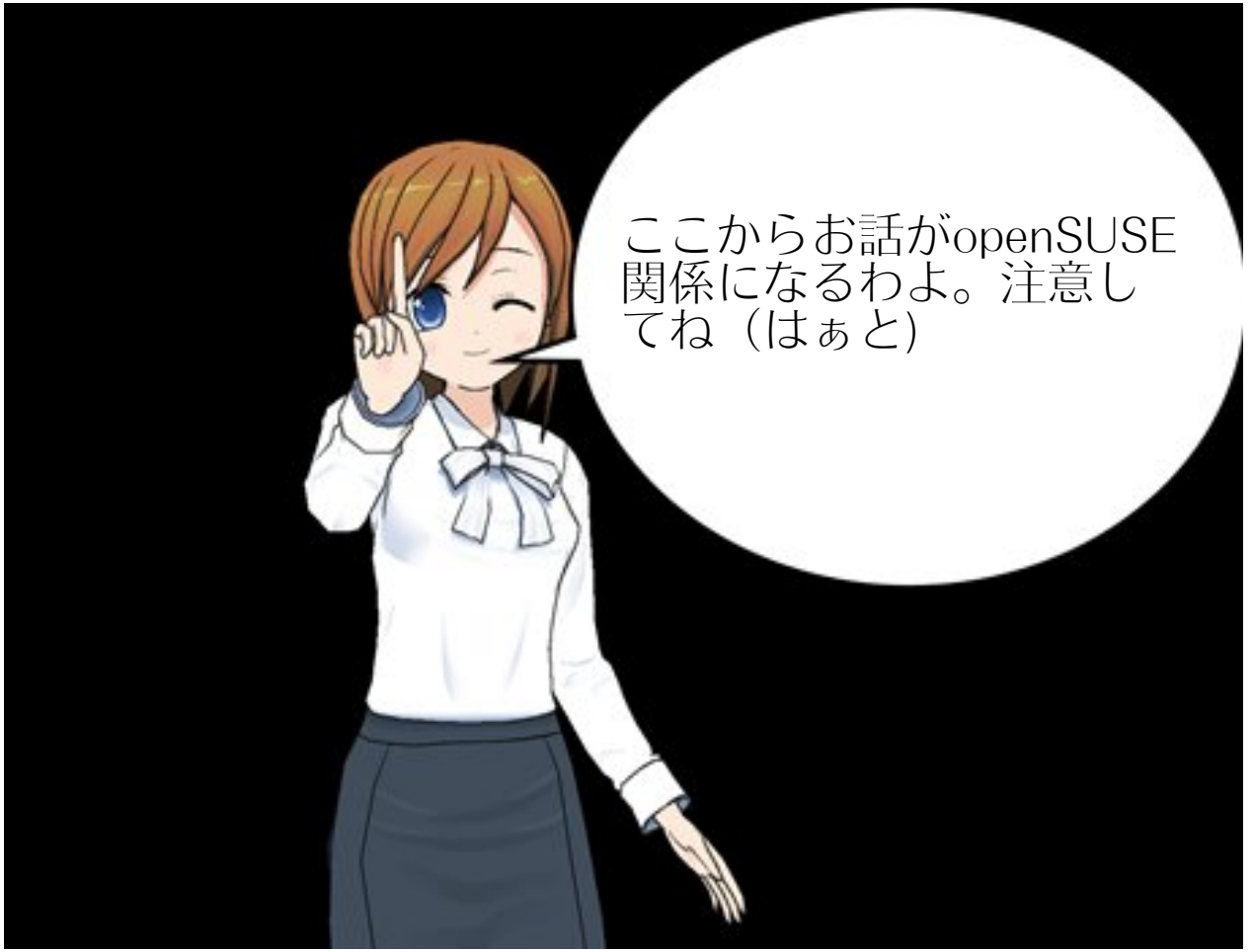
狙われるソフトウェア

- Telnet
- Apache
- SMTPサーバー
- Domain
- MySQL,PostgresQL
- SSH
- Sun JAVA（Oracle JAVA)
- PHP、RubyなどのLL(LightweightLanguage)


なるほど。
どのようにしてソフトのチェックはやるの？

チェックツール

- OpenVAS(脆弱性チェックツール)
- Nessus(同上)
- NMAP(ポートスキャン)
- John The Ripper(Passwordチェック)
- Metasploid Framework（Windows)


ここからお話がopenSUSE関係になるわよ。注意してね（はぁと）

OpenVAS

- オープンソースの脆弱性スキャナ
- http://www.openvas.org/
- インストールは
http://software.opensuse.org/121/ja
から検索して1-Clickインストールでどうぞ。
- リポジトリは
security:OpenVAS:STABLE:v4/openSUSE_12.1です

必要なパッケージ

- openvas-administrator
- openvas-manager
- openvas-scanner
- openvas-server
- greenbone-security-assistant
- gsd

openVAS関係のパッケージは以上です。

使用前の準備

- 管理ユーザーの追加
- feed(チェックリスト)のダウンロード
- openvas-check-setupスクリプトの実行
- openvas-serverの起動

管理ユーザーの追加

- #openvasad -c add_user -n <username> --role=Admin

証明書の作成

- # openvas-mkcert
- # openvas-mkcert-client -n om -i

NVT Feedの更新

- # openvas-nvt-sync
- # greenbone-nvt-sync

openvas-check-setupの実行

https://svn.wald.intevation.org/svn/openvas/trunk/tools/openvas-check-setup

からダウンロードしてくる。

#wget -no-check-certificate https://svn.wald.intevation.org/svn/openvas/trunk/tools/openvas-check-setup

#chmod +x openvas-check-setup

#./openvas-check-setup

特に問題がなければ起動するはずです。

gsdの起動

- gsd（GUIコンソール）を起動します。
- 先ほど設定したユーザーでログインします。
- スキャナでスキャンします。
- 結果はGUIできれいに出てきます。


これからサーバーソフトの話になるわよ♪

ネームサーバーの問題点

- DNSコンテンツサーバーとDNSキャッシュサーバーの分離
- BINDは高機能だが複雑でセキュリティ関係のバグが多い

解決策

- Unbound(キャッシュ)とNSD(コンテンツ)による分離
- 問い合わせエリアの制限

Unbound

- 主にクライアントからのDNSクエリーに応答する
- クライアントから受けたDNSクエリーをDNSコンテンツサーバーに問い合わせて答えをクライアントに送る。
- 答えをキャッシュする。
- DNSキャッシュポイズニング攻撃に強い

DNSキャッシュポイズニング

- DNSキャッシュ汚染攻撃
- 通信にUDPを使うためパケットの偽装が行いやすい
- カミンスキーアタックなどが有名
- 対策にはDNS-SECが有効
(汎用JPドメインのルートサーバーは対応済)

NSD

- DNSコンテンツサーバー
- クライアントからのDNSクエリーには応答しない。
- 自分が管理しているレコードしか応答しない

メールサーバー

- Sendmailの設定が複雑
- スパムメールをどうするか？
- SMTP-AUTHを導入したい

Sendmailの後継

Postfixやeximを使う。

- 利点は設定ファイルの記述が優しいので人為的ミスが発生しにくい
- Sendmailと違って構造が簡素なのでメールスループットが高い
- 日本語の資料もかなり豊富にそろっている
- SMTP-AUTHは敷居が高い？

SMTP-AUTH

- SaslauthdとPostfixの連携で実現可能
- LDAPとPostfixの連携も可能
- 詳しくは次の機会にお話しします

さて、セキュリティって？

今まではネットワークよりのセキュリティ

まだまだ語り足りないところがありますがソフトウェアのセキュリティについて少し語りたいと思います。

ソフトウェアのセキュリティ

- 脆弱性とは大概バグである
- 脆弱性によるアタックを受けたときにどのようにして守るべきか？

脆弱性

- そもそもソフトウェアは人が作るものなのでバグ(脆弱性)ができるのは当たり前である
- Webアプリケーションであれば情報漏洩が一番の問題である。
- 絶対に落とせないサーバーをどのようにして守るか？

バグ撲滅

- 定期的にコード監査ツールを実行する
- コーディングルールの徹底
- ソースコードレビュー
- リバースエンジニアリングをやってみる

Webアプリケーション

- WAFを導入する(mod_securityなど)
- SNORTのルールが使えるので入力チェックなどをWAFで行う。
- ただし、RoRなどのフレームワークが提供している入力値チェックの機能も活用するべき

サーバーの防衛策

- Linux-HAによる二重化
- UltraMonkeyによるWebアプリの負荷分散
- SE Linux、TOMOYO Linux、AppArmerの導入

Linux-HA

- オープンソースの死活監視ソフト
- 詳しくは、http://linux-ha.sourceforge.jp/wp/ へどうぞ

UltraMonkey

- オープンソースのL7ロードバランサ
- つまりIP層ではなくプロトコルレベルで負荷分散を行う
- これとLinux-HAを組み合わせることで強固なシステムを構築可能
- 詳しくは http://sourceforge.jp/projects/ultramonkey-l7/ をどうぞ

UltraMonkeyとLinux-HAの二重化で十分か？

万が一、ハッキングされたときどうする？

被害低減策

- 適切なアクセス権
- CHROOTの応用
- SE Linux、TOMOYO Linux、AppArmerの導入(いわば最後の砦)

アクセス権とアカウント

- WHEELグループの有効利用
- サーバーにログインさせないユーザーのシェル設定(/etc/nologinに設定するなど)
- アカウントに対するパスワードロックアウト
- John The Ripperに引っかからないパスワードを使う

強制アクセス制御

- SE Linux
- TOMOYO Linux
- AppArmer

openSUSE12.1ではTOMOYO Linuxが利用できます。

強制アクセス制御

- ユーザーはプロセス、ファイル、システムデバイスに自由にアクセスすることができない
- ユーザーは管理者が決定したアクセス権より緩くすることはできない。

つまり

アプリケーションごとにユーザーが定義されているので適切に設定することができればかなり被害を防ぐことができる。

だから最後の砦である。

最後に

- コンピューターシステムは人が作るものであってバグは必ずどこかに潜んでいる。
- ネットワークからの侵入に対して考えるだけではなく万が一、侵入されたときのことを考えるべき
- OSSでもそれなりのことはできる。お金がないからできないは言い訳にならない。

課題

- セキュリティに関するドキュメントが少ない。

これについては私の時間が許す限りがんばって充実させていきます。

- セキュリティをわかっている経営者がいない

何をやるにしてもおろそかになってしまう。

何かが起きてから手を打つのは大変です

最後までおつきあいいただきありがとうございました。