

発行元：CTC教育サービス

目次

- キャンペーン Cisco「1コース無料キャンペーン」実施中！！
- 新コース システム仮想化基盤構築トレーニング（仮想デスクトップ分野）
MPLS-VPNファーストステップ
- トピック Inst. Tech View ～第14回“Androidマルウェア”～
- コラム スーパーエンジニアの独り言 第12回“メギドの丘”

メルマガコラムのバックナンバーを公開中です！
<http://www.school.ctc-g.co.jp/melmaback/index.html>

キ ャ ン ペ ー ン

Cisco「1コース無料キャンペーン」実施中！！

Cisco認定コースで、キャンペーン対象のコースを受講すると、
 もれなくもう1コースが無料受講できる、大変お得なキャンペーンを実施中です！
 ネットワークの学習をした後に、仮想化知識を学ぶ。
 はたまたクラウドの概要や、VoIPの基礎について学習する。
 自由な選択肢をご用意しています！
 CCNAやCCNP関連の受講をご検討されている方、必見です！

キャンペーン期間：2012年6月1日から2012年10月31日
 キャンペーン内容：対象コースを受講すると、リスト内のコースが無料で受講
 キャンペーン対象コース：11コース
 無料受講対象コース：7コース

<http://dm.ctc-g.co.jp/c?c=748&m=11740&v=c6cc1e78>

新 コ ー ス リ リ ー ス

システム仮想化基盤構築トレーニング（仮想デスクトップ分野）

社内のITインフラをリモートデスクトップ環境へ移行する企業が増えています。
 リモートデスクトップ環境とは、企業内のスタッフが使用するデスクトップPCを
 サーバ上の仮想マシンとして用意し、リモートアクセスで利用させる環境です。

マシン自体は社内もしくはデータセンター内のサーバ上で集約されていること
 から、IT管理者がスタッフのデスクトップを一元管理することが可能であり、
 スタッフも場所を選ばず、自身のデスクトップ環境を社外から使用することが
 できます。

以前から注目されている、BCP（business continuity plan）対策としても
 適切なソリューションの一つであるといえます。

様々な仮想化ベンダーからリモートデスクトップ環境を構成するための
 ソフトウェアパッケージが販売されておりますが、本コースではVMware社の
 「VMware View」製品を効率的に学習して頂き、リモートデスクトップ環境の
 イメージを掴んで頂くことを目的としています。
 講義では実機を使用した演習も用意しております。

IT500
 システム仮想化基盤構築トレーニング（仮想デスクトップ分野）
 2日間 ￥147,000
<http://dm.ctc-g.co.jp/c?c=749&m=11740&v=63478e76>

MPLS-VPNファーストステップ

ページ(1)

「MPLS-VPNファーストステップ」コースは、MPLS-VPNを構成するための基本知識と技術を紹介する2日間コースです。独学では修得の困難なMPLS-VPNを、弊社オリジナルのテキストを使用して分かりやすく紹介します。実際にCiscoルータを約7台使用して、MPLS-VPN網の構築を行いますので、短期間で実践的なスキルを身に付けられます。CCIEの資格対策としてもお奨めできます。

N682
MPLS-VPNファーストステップ
2日間 ¥126,000(税込)
<http://dm.ctc-g.co.jp/c?c=750&m=11740&v=0dda2b58>

トピック

『 Inst. Tech View ~ 第14回 “Androidマルウェア” ~ 』

今回のInst. Tech Viewは、Androidマルウェアの話題です。

昨今のスマートフォンの急速な普及に伴い、AndroidやiPhoneを標的としたマルウェアが多く発見されています。その中でも「Androidマルウェア」と呼ばれるAndroidに感染して動作するマルウェアによる被害が近年多く報告されています。Androidマルウェアがスマートフォンに感染すると、メールや連絡帳・通話記録・画像ファイルなど様々なデータが取得される、あるいは遠隔から操作して課金が発生する宛先にSMSを送信して金銭が搾取されてしまうといった被害を受けることが報告されています。

Androidマルウェアを作成する手法として代表的なものは、マーケットで公開されている有名なアプリケーションに悪意のあるコードを埋め込み、それを公開する手法です。Androidのアプリは単なるzipのアーカイブファイルのため、アーカイブファイル内のプログラムを抽出し、それを逆コンパイル（プログラムに記載されているコンピュータが理解できるオブジェクトコードを人間が理解できるソースコードに変換すること）してソースコード化します。次にソースコードに悪意のあるコードを埋め込み、再コンパイル後、再びアーカイブファイルにして正規のアプリと見せかけてマーケットに公開します。

結果、そのアプリを正規のものと勘違いした人がスマートフォンにダウンロードしてアプリを起動すると、埋め込まれた悪意のあるコードが実行され、様々なデータが取得される、あるいは金銭を搾取されるといった被害を受けることとなります。

Androidマルウェアに感染しないようにするための対策としては、各キャリアが提供するスキャンソフトの利用や、不審なアプリはダウンロードしないこと、アプリをインストールする際に表示されるアクセス許可の一覧の中で妥当と思われない許可を求めてくる場合はダウンロードを取りやめること等が挙げられます。

PCを標的にして作成されたマルウェアは、長い年月を経て進化を遂げてより巧妙で悪質なものとなっており、Androidマルウェアも同様の歴史をたどることになるかもしれません。Androidマルウェアによる被害を防ぐためには、PCでのマルウェア対策と同様に、スキャンソフトの導入など適切な対策を施し、情報収集を行って安全に使用するよう心掛ける必要があります。

弊社ではマルウェアをはじめとしたコンピュータへの様々な攻撃手法の実態と、その対策方法を紹介するコースを新しく開発しました。セキュリティ関連の知識を向上させたい方にお勧めです。

コースの詳細情報はこちら：

N472: 実践！コンピュータセキュリティ
~ ハンズオンで学ぶハッキング手法とその対策 ~
2日間 ¥105,000 (税込)

コ ラ ム

『スーパーエンジニアの独り言 第12回 “メギドの丘”』

JavaとRubyの例外処理の続きです、前回と併せてお読みくださいませ。
前編は基本構文を比較してみました、後編はfinally(ensure)節での挙動について言及を試みます。

JavaとRubyを例外処理にて比較した場合にも、やはり実行形態（処理系）が異なる点がコーディングにおいても相違点の一因になると考えられます。
Javaではソースコード中にライブラリのメソッドが例外をスロー（送出）し例外チェックが必要な場合に、コード内に例外処理を記述しないとコンパイルエラーになります。例外発生が予期される箇所を事前に教えてくれると理解出来ます。適切な例外ハンドラを用意しなさいという教えるを請うのです。（但し、RuntimeExceptionサブクラス等は例外チェックがありません。）

その反面、Rubyの場合には処理系がインタプリタのためにコンパイル手順そのものはありません。Rubyがスクリプト用途としての簡易実行が可能であり、つまり自由度が大きい分、プログラマにその責務を委ねているとも思えます。（RubyMotionというRubyコンパイラも最近登場です。）

例外発生時には、catch(rescue)節で指定した例外を捕捉して適切な処理を実施すべきですが、catch(rescue)節を省略した場合にはデフォルトのエラー処理がなされます。（よく画面で見かけるあのメッセージです。）その後、プログラムは終了するという事になります。プログラムで例外の発生を回避し処理を継続したい場合には省略せずに捕捉しましょう。

プログラムの終了前に「例外発生の有無に関わり無く」後処理をしたい際、Javaは「お終い（お仕舞い）に(finally)」、Rubyは「確かに(ensure)」とキーワードで指定出来ます。最後に必ず実行してね、と書けるのです。

そこで本題となりますが、
『finally(ensure)節の処理を、いつも実行してくれるのか否か?』

例外を大別すると「致命的なエラー」とそれ以外の二つがあります。
プログラムを実行する母体である仮想マシン（インタプリタ）が異常終了するような「致命的なエラー」が発生した場合、プログラムの実行自体が継続出来ないために、コード内で例外を捕捉し後処理を記述したとしても適うことは稀な事でしょう（タイミングや処理系の実装に拠ります）。

Javaでは、JVM（もしくは外部）で発生した「致命的なエラー」は、Errorのサブクラスで表現され、これらは通常捕捉することが出来ません。
Rubyでは、インタプリタ内部で「致命的なエラー」が発生した場合 "fatal" と表現され、通常プログラムからは参照することが出来ません。

```
// Javaの例外クラスツリー構造（一部）
Object
  Throwable
    Error
      VirtualMachineError
      OutOfMemoryError
      StackOverflowError
    Exception
      RuntimeException

# Rubyの例外クラスツリー構造（一部）
Object
  Exception
    fatal
    NoMemoryError
    SystemStackError
    SignalException
    SystemCallError
```

```
SystemExit
StandardError
  RuntimeError
```

JVM(ruby)自身が異常終了する場合、これを捕捉出来ないのは道理ですが、例えば、JVMがメモリ不足(OutOfMemoryError, NoMemoryError)に陥る現象が発生すると当然「致命的なエラー」の仲間ですが、発生原因に拠っては、この例外を捕捉することも可能な場合もあります。捕まえたい場合は、`catch(OutOfMemoryError e)` もしくは、`rescue NoMemoryError` そして、`finally` と `ensure` も思い切って書くと良いでしょう。捕捉できたならきつと、後処理もしてくれることでしょ。

```
// Javaで OutOfMemoryError を catchして e.printStackTrace();
java.lang.OutOfMemoryError: Java heap space
  at OutOfMemoryTest.main(OutOfMemoryTest.java:9)
```

```
# Rubyで NoMemoryError を rescueして p ex
#<NoMemoryError: failed to allocate memory>
```

スタックオーバーフロー(StackOverflowError, SystemStackError)も同じです。しかしJavaでは別スレッドの処理中にメモリ不足が発生した場合には、そのスレッドが終了してしまうと例外を捕捉出来ません。その際には、UncaughtExceptionHandler (JDK 5 以降) が使えます。ThreadGroupでこのインターフェースが実装されていますので、例外を逃さず捕まえましょう。

最後に、明示的にプログラム終了を指示する場合についてです。Javaで `System.exit()` を呼び出すと、`finally`節は実行されません。Rubyでは `exit!`を呼び出した場合、`rescue`節は実行されませんが、`exit`, `abort`, `at_exit`メソッドなどで終了する場合には、`rescue`節はちゃんと実行されます。

これはUNIXのライブラリ`exit(3)`とシステムコール`exit(2)`のどちらを呼び出しているのかの違いになります。ライブラリではユーザ空間の後始末をしようとしてくれますが、システムコールはカーネル空間で終了するため、`rescue`節が無視されてしまうのでしょ。(UNIXのman頁が参考になります、是非御一読ください。)

尚、詳細については各々の言語仕様を是非ご参照くださいませ。

今回は少しコードをお見せしたいと思ったのですが文字で埋め尽くされてしまいました。またの機会にご紹介したいと思います。次回をお楽しみに。

Ruby関連コースはこちら

<http://dm.ctc-g.co.jp/c?c=752&m=11740&v=9dbc0d05>

お問合せ・ご意見・ご感想は C T C教育サービス 窓口まで
シーティーシー・テクノロジー株式会社 エデュケーションサービス部
E-Mail : kyouiku@ctc-g.co.jp / TEL : 03-5712-8701

外部委託について

弊社はメールニュース配信業務をシーティーシー・ビジネスサービス株式会社 (C T C 100%出資子会社) に委託しております。

本メールマガジン編集・配信責任者

C T C T エデュケーションサービス部 部長 篠原 義一

所在地：東京都世田谷区駒沢1-16-7 ctc_edu_mail@ctc-g.co.jp

個人情報保護方針

C T Cグループの個人情報保護方針につきましては下記URLをご参照ください。

http://www.ctc-g.co.jp/guide/security_policy.html?top=b_security

配信中止及びお問合せ対応について

- ・「C T C教育サービス News&Topics」の配信が不要な場合には、下記URLから配信停止のお手続きを行ってください。
<https://krs.bz/ctc-g/m/ctc-education>
- ・当社では、複数種類のメールマガジンやメールニュースを発行しております。大変お手数ではございますが、C T C教育サービス以外からのメール配信についての受信拒否および個人情報に関するご要求は、各メールに記載の個々ページ(4)

ctc20120622

- の連絡先宛にそれぞれご連絡をお願いします。
- ・受信者ご本人様からの個人情報の開示・訂正・削除に関するご要求は、随時 ctc_edu_mail@ctc-g.co.jpにてお受けいたします。