Javaセキュアコーディングセミナー東京
第1回
オブジェクトの生成とセキュリティ
演習

2012年9月9日(日)
JPCERTコーディネーションセンター
脆弱性解析チーム
戸田 洋三

演習[1]

```
class Dog {
  public static void bark() {
    System.out.print("woof");
  }
}

class Bulldog extends Dog {
  public static void bark() {}
}

public class Bark {
  public static void main(String args[]) {
    Dog d1 = new Dog();
    Dog d2 = new Bulldog();
    d1.bark();
    d2.bark();
  }
}
```

A. どのような出力が得られるか?
B. bark()メソッドがstatic宣言されていない場合の出力は?
C. メソッドがどのように実行されているか説明せよ

ヒント
Java言語仕様 §15.12 Method Invocation Expressions

演習[2]

A. どのような出力が得られるか?
B. なぜこのような出力が得られたのか説明せよ.
C. 適切な出力が得られるようにコードを修正せよ.

```
class Point {
  protected final int x, y;
  private final String name;
  protected String makeName() { return "[" + x + "," + y + "]"; }
  public final String toString() { return name; }
  Point(int x, int y) {
    this.x = x; this.y = y;
    this.name = makeName();
  }
}
public class ColorPoint extends Point {
  private final String color;
  protected String makeName() { return super.makeName() + ":" + color; }
  ColorPoint(int x, int y, String color) {
    super(x, y);
    this.color = color;
  }
  public static void main(String[] args) {
    System.out.println(new ColorPoint(4, 2, "purple"));
  }
}
```

ヒント: Java言語仕様 §15.12 Method Invocation Expressions

演習[3]

```
class Purse {
  private int i;

  public Purse(int arg) {
      i = arg;
  }
  public int get_i() { return i; }
  public void set_i(int iarg) { i = iarg; }
}

class User {
  private Purse p;
  public User(Purse arg) {
      p = arg;
  }
  public Purse get_p() {
    return p;
  }
}
```

A. User クラスのインスタンスを生成し，その private フィールド p が参照する Purse インスタンスの持つ値を変更する攻撃コードを書け.
B. A. でつくった攻撃コードが動作しないように元のコードを修正せよ.

演習[4]

```
class Authlet {
  int i;
  Authlet(int i0){
    if (checkarg(i0)) { this.i = i0; }
  }
  boolean checkarg(int i) throws IllegalArgumentException {
    if (i<0 || 100<i) {
      throw
        new IllegalArgumentException("arg should be positive < 100.");
    }
    return true;
  }
}
```

```
class Auth {
  private static Authlet a0;

  public static void checkAuth(Authlet a){
    if (a0 == null){
      if (a == null){
        System.out.println("invalid Authlet!");
        System.exit(1);
      }
      a0 = a;
    }
  }
}
```

演習[4]

```
class useAuth {
  public static void main(String[] args){
    Authlet au;
    try {
        au = new Authlet(Integer.parseInt(args[0]));
    } catch(SecurityException ex){
        au = null;
    }
    Auth.checkAuth(au);
    System.out.println("Authenticated!");
  }
}
```

A. checkarg() による入力値チェックを回避する攻撃コードを書け.
B. A.で書いた攻撃コードに対する対策を行え.