

**GLOBAL
INFORMATION
ASSURANCE
CERTIFICATION**



CANDIDATE BULLETIN

**GIAC
Secure Software
Programmer
(GSSP)**

***Secure Programming Skills
in Java/Java EE***

**GIAC Secure Software Programmer (GSSP)
Java/Java EE 受験要綱**

SANS

In Cooperation with the SANS Software Security Institute



GIAC Secure Software Programmer (GSSP) Java/Java EE 受験要綱

目次

A. 序章	2
GIAC 認定試験	2
B. GIAC SECURE SOFTWARE PROGRAMMER (GSSP) 試験	2
C. 受験資格	3
D. GIAC 倫理規程	3
E. 試験情報	4
受験申込み期限・試験日程	4
受験料	4
受験申込み手続き	4
出題内容	5
試験時間	5
試験当日	5
受験規則	6
成績	6
認定	7
再受験	7
F. サンプル問題と解答	8
付録 A : コンテンツブループリント:	13
Java/JAVA EE におけるセキュアプログラミングスキル	13
付録 : 参考文献	18
ソフトウェアセキュリティに関する参考書	18
ソフトウェアセキュリティに関する Web サイトや Podcast	19

A. 序章

開発者やプログラマーがより安全なコーディングスキルを身につけない限り、政府や企業、教育機関は、ソフトウェア脆弱性対応という終わりなき戦いを強いられる運命にあります。そこで、セキュアプログラミングスキルおよび知識を促進する各イニシアチブが進行中です。シマンテック、オラクル、マイクロソフト、ほか数社のソフトウェア企業は、自社プログラマー向けに短期研修を行っていますし、SPI Dynamics や Fortify Technology は、大学と連携してプログラミングコースの受講生にリアルタイムフィードバックシステムを提供しています。また、多くの大学がセキュアプログラミングに関するオンラインコースを設けています。しかし、たとえこうしたイニシアチブの全てが功を奏しているとしても、その影響が、既に実務に就いている、もしくは今後 5 年間に開発者やプログラマーになると想定される 150 万人のうちの 2%にも及んでいるかどうか疑わしいのが現状です。

このような状況に対応するため、SANS Institute は Global Information Assurance Certification(GIAC)を通じて、ソフトウェアのプログラマーおよび開発者向けの試験認定制度を整備し、セキュリティ脆弱性につながる一般的なプログラムエラーの確認や修正に要する技術の習熟度と専門知識の有無について、信頼できる基準の策定を行ってきました。

SANS は、情報セキュリティのトレーニングおよび認定試験において、世界で最も信頼できる最大の組織です。SANS による様々な情報セキュリティ調査研究結果の膨大な資料は、無償で入手することができます。また、インターネットの早期警戒システムである Internet Storm Center の運営も行っています。SANS は、盛んに弱点をつかれる恐れのある最大の脅威から、アプリケーションやシステム、ネットワークの防御に要する実践的な技術を習得することができる集中トレーニングを提供しています。

GIAC 認定試験

GIAC は Global Information Assurance Certification の略であり、1999 年、IT セキュリティの専門家や開発者の実質的なスキルの有効性を証明するべく設立されました。コンピュータ、ネットワーク、アプリケーション、そしてソフトウェアのセキュリティに関する主要分野の実践的な知識とスキルの保持者の認定を目的としています。この認定資格は、情報セキュリティおよび安全なソフトウェア開発の実務に伴う様々な責任を担うに相応しいものです。GIAC は、総合的な知識よりも、具体的に特定された知識を推し量るという点に独自性があります。

B. GIAC SECURE SOFTWARE PROGRAMMER (GSSP) 試験

GIAC Secure Software Programmer (GSSP) は、ほとんどのセキュリティ問題につながる一般的なプログラムエラーの対処に必要な知識とスキルの習得者を認定します。対応言語は以下のとおりです。

- Java/JEE
- PHP
- C
- Perl

・ C++

・ ASP .NET and .NET

※ 「Java/JEE」「C」以外の言語については開発中

この試験は、まず第一に開発やコードの維持に携わるプログラマー／開発者を対象としています。すなわち、ソフトウェアコードの検証／監査、品質保証や検査のスキルも重要となる業務の担当者です。試験では、脆弱性につながる一般的なエラーに焦点を当てます。日々のコーディング作業に直接関わる安全なプログラム規則のみならず、各プログラム言語の遂行課題の解決も目標としています。

机上の知識を測るのではなく、コーディングエラーの認知力、妥当な規則の適用力、安全なコーディングの核を構成できる提案力、といった実践的な能力を測る試験です。試験問題の多くが、コードサンプルからのエラーの発見と除去方法の決定を問うものになります。

C. 受験資格

GSSP の受験資格は特にありません。

D. GIAC 倫理規程

情報セキュリティの専門家と開発者の影響範囲と責任領域は多岐にわたります。提供するサービスは、組織の成功を左右するものであり、IT およびソフトウェア開発業界のセキュリティ全体に対する姿勢に関わる重要なものです。そのような責任は、情報セキュリティおよび安全なアプリケーション開発規則の実践倫理基準をクリアした認定資格者に対する重大な期待の下に位置づけられます。

GIAC 認定は、取得・維持に値する荣誉であると認知されています。GIAC 認定資格者は、当該倫理規程の提唱、順守、支持を誓約します。

当該規定のいかなる原則に対して故意に違反する GIAC 認定資格者は、GIAC より懲戒処分を受けます。

社会に対する責任

私は、コミュニティのセキュリティや福祉に関係する決定を下すに当たり、相応の責任を負います。

私は、コミュニティ、私のプロフェッショナルとしての評価、または情報セキュリティの原則に悪影響を与えるような非倫理的あるいは違法な集団と関係を持ちません。

認定に対する責任

私は、GIAC 認定手続きに関する機密情報を他に漏らしません。

私は、私の認定および認定に関連する物や情報（認定証やロゴといったもの）を、GIAC 認定資格者である私自身を象徴するため以外、いかなる個人または団体の用にも供しません。

雇用主に対する責任

私は、私の認定資格と地位に対する期待に見合うレベルの高いサービスを提供するよう努めます。

私は、知り得た機密情報の守秘義務を負います。

私は、リスクマネジメントを実践することにより、IT ソリューションの機密性、完全性、可用性に対するリスクを最小限に抑えます。

自身に対する責任

私は、利害関係の衝突を回避いたします。

私は、私の立場上保持するいかなる情報および特権も悪用いたしません。

私は、私の技能、コミュニティにおける職務、雇用主、同僚について偽装いたしません。

E. 試験情報

受験申込み期限・試験日程

GSSP 試験の受付は年中行っています。直近の実施情報はこちらです。

<http://www.sans.org/gssp/>

申し込みは、希望試験日の 30 日前までに済ませてください。

受験料

日本国内で開催される GSSP 試験の受験料は 57,000 円です。

支払い方法は、口座振込みのほか、Visa、マスターカード、JCB、American Express が利用できます。日本国内開催の受験料は、すべて円建てのお支払いとなります。受験料はいかなる理由があろうとも払い戻しできません。試験運営事務局の入金確認が済んだ時点で、お申込み受付の完了となります。

受験申込み手続き

1. 申込みに先立って受講要領に目を通し、記載内容に従ってください。手続きに失敗すると申込みができない場合がありますのでご注意ください。
2. 受験申込みサイトで申込みを完了させます。登録項目の全てを埋めてください。ここで入力された情報が、受験者との連絡に必要となります。手続き迅速化のため、郵便番号、電話番号、E メールアドレス全てに漏れない完全な情報を提供してください。申込みおよび認定の手続き期間中は、常に最新情報の受信を可能にするため、SANS に提供する受験者の属性情報は最新の状態であるよう留意してください。

以下の理由により、GSSP 受験の申込みが拒否されたり、GIAC 認定が取り消されたりすることがあるので注意してください。

- ・不備のある申込み
- ・登録情報の改ざんまたは虚偽の登録
- ・GIAC 倫理規程違反による罰則の適用

3. 試験日の約 3 週間前までに、GSSP 試験実施事務局より各受験者に受験票を送付いたします。試験当日、必ず持参してください。

出題内容

多岐選択式問題が 100 問出題されます。

2 つ以上の選択肢を回答する問題もあり、その旨明記されています。

各試験に数種類のバージョンがありますが、全受験者に間違いなく公平を期すため、実施バージョンによって合格可能性に偏りはありません。上記「試験沿革」の項にあるとおり、専門家によって識別された全試験問題の難易度レベルが、難易度比較可能な試験内容に反映されています。

試験時間

試験の制限時間は 6 時間です。

終了した方は、随時退席可能です。

試験当日

試験当日は、時間に余裕をもって会場にお越しください。

試験開始の 15 分前に受験に関する注意事項を説明します。試験開始時刻の 15 分前には試験会場に入場してください。

また、必ず顔写真入りの身分証明書を持参してください。

身分証明書となるもの：

運転免許証

パスポート

その他政府公認の身分証

身分証明書とならないもの：

スポーツジムの会員証

店舗の会員証

学生証

クレジットカード

顔写真のない証明書

※試験場への入室前に、本人確認をさせていただきます。

鉛筆、メモ用紙、回答用紙、問題冊子は試験会場で配付します。これらは全て試験室より持ち出し禁止となります。

受験規則

1. 試験の保証事項と手順

- いかなる書籍、書類、その他参考資料も試験室へは持ち込めません。
- 電話、ノート PC、カメラ、信号を発する媒体、ポケベル、アラーム機器、計算機、および録音／再生機器（iPod や mp3 プレイヤーを含む）など、いかなる電子機器も試験室へ持ち込めません。これらのものは試験会場に持参しないようにしてください。
- 音を遮るための耳栓は持ち込み可能です。
- 試験を終えた受験者は試験室より退出してください。
- 受験しないいかなる同伴者の入場も認めません（親、子供、配偶者、友人、介助動物以外のペットほか、いかなる人物や動物も不可）。
- 試験に要した道具、資料や全てのメモは試験室より持ち出せません。試験で使用した物は退出の前に全て提出してください。
- 試験施行中は、他の受験者と議論したり、参考資料や試験情報についていかなる情報共有も行わないでください。試験室退出後の情報共有も認められません。
- いかなる理由があっても、試験問題のコピーはできません。
- 飲食物は指定された場所でのみとることができます。
- 試験施行中は、試験に関するいかなる質問も受け付けません。試験監督の説明をよく注意して聴いてください。
- 試験は、受験票記載の日時にのみ受験することが可能です。
- ドレスコードはビジネス・カジュアルです。
- 空調は可能な限り調整いたしますが、念のため寒暖に備えられる服装で臨んでください。

2. 参考資料

適正な試験運営のため、試験監督より配付された試験資料以外の物へは何も書き込まないでください。参考資料は試験室へ持ち込めません。受験者の私物の持参は最小限にとどめてください。

3. 解答用紙

解答用紙には、受験者の氏名およびその他指定の必要事項を記入してください。解答はすべてこの用紙に記入してください。解答が終了したら、試験監督が受験者の試験関連物の回収を行うまで座席で待機してください。問題冊子に記入された解答は採点対象となりませんので、解答を問題冊子から解答用紙に書き写す作業は制限時間内に終わってください。

成績

試験結果は、試験日より 6 週間後に米国より郵送いたします。合否のほか、総得点と各試験分野ごとの得点内訳が記載されます。電話や E メール、FAX による照会はできません。

認定

合格者は **GIAC Secure Software Programmer (GSSP)** 認定を付与されます。選択した開発言語に応じた認定内容となります。例えば、**Java** の試験に合格したプログラマーは **GSSP-J** 認定取得者となります。

GSSP 認定は **4** 年間有効です。有効期限最後の **1** 年間、更新試験受験資格が与えられます。更新試験の受験申込み受付期間は **この 1 年間** です。更新試験に合格しなければ、認定資格失効となります。

再受験

不合格の場合、試験日より **4** ヶ月の待機期間後より再度同様の申込み手続きが可能となり、受験できます。受験料は同額です。

F. サンプル問題と解答

サンプル問題

- 1) Javaのsynchronizedキーワードがセキュリティ上重要である理由として、当てはまるものを選択してください。
- A. 2つの異なる関数を同時に実行できるようになるため。
 - B. 複数の開発者による同じコードブロックへの書き込みを防止できるため。
 - C. JREの起動直後にクラスをロードできるようになるため。
 - D. 複数のスレッドによる同じコードブロックへの同時アクセスを防止できるため。
- 2) Java内部クラスにおけるセキュリティ上の問題の原因として、正しい記述を選択してください。
- A. 内部クラスと外部クラスのプライベートメンバに、パッケージ可視のアクセサメソッドが作成されてしまう。
 - B. 内部クラスと外部クラスのいずれにおいてもfinalメソッドを拡張できてしまう。
 - C. 内部クラスはJVMによって認識されないため、その内部クラス自体が含まれるクラスのサブクラスに変換されてしまう。
 - D. 最適化のため、内部クラスのデータメンバはJVMによりメモリにキャッシュされてしまう。
- 3) SSLを利用してJAVA EEサーバアプリケーションを動作させるには、何が必要でしょうか。
- A. Verisignなど、第三者の認証局によって発行された証明書の利用
 - B. 認証局による証明書もしくは自己証明書の利用
 - C. Sunの暗号化プロバイダパッケージのJVMへのインストール
 - D. SSL通信の前のオフラインでのクライアントとの鍵交換
- 4) JSESSIONIDを使用するJAVA EEアプリケーションで、安全なセッション管理として適切でないものを選択してください。
- A. JSESSIONIDを送信する通信を暗号化する。
 - B. JSESSIONIDをパーマネント（永続的な）Cookieでなく、セッション（一時的な）Cookieとして保管する。
 - C. JSESSIONIDの妥当性チェックを行い、メタ文字が含まれないことを確認する。
 - D. JSESSIONIDを長くランダムに生成する。
- 5) 入力チェックを実行するJAVA EE層を1つだけ選択しなければならない場合、最適な層は次のうちどれですか。
- A. 適切に実行すれば、どの層でも問題ない。
 - B. エンタープライズ（バックエンド）層。SQLインジェクションなど、バックエンドシステムに対するインジェクション攻撃を防御できるため。

- C. 中間層 (Middle Tier)。アプリケーションのリソースへの入口であるため。
- D. クライアント層 (Client Tier)。アプリケーションへの入口であり、妥当性チェックをできるだけ早期に実行する必要があるため。

6) Javaが提供するPreparedStatementクラスは、適切に使用することでSQLインジェクション攻撃を防御できます。しかし、下記のコードでこのクラスを使用すると、インジェクション攻撃の犠牲となるおそれがあります。問題点を選択してください。

```
PreparedStatement stmt = con.prepareStatement
    ("SELECT ssn FROM usersTable WHERE "
     + "name = " + getParameter("username") );
ResultSet rs = stmt.executeQuery();
```

- A. ユーザがusernameを指定できるため、別のusernameに変更してそのユーザのssnにアクセス可能になること。
- B. 危険なコンテンツに対してusernameパラメータの妥当性チェックが行われていないこと。
- C. クエリを保護するため、SQL文字列はあらかじめ定義されてからPreparedStatementに渡されるべきであること。
- D. PreparedStatementが通常の動的SQLクエリとして使用されており、セキュアな置換方法でデータが入力されていないこと。

7) JAVA EEのFilterは、アプリケーションのセキュリティ対策を強化できます。その理由として誤っているものを選択してください。

- A. Filterを使用してJDBCデータベースコネクションでデータの妥当性をチェックし、インジェクション攻撃を防御できる。
- B. Filterを使用してJAVA EEアプリケーションをラップし、既存のコードを変更することなく新たなセキュリティ対策を追加できる。
- C. アプリケーションへのリクエストをFilterで変更して、入力チェックなどのセキュリティ操作を実行できる。
- D. アプリケーションによるレスポンスをFilterで変更して、ヘッダの書き換えなどのセキュリティ操作を実行できる。

8) 次に挙げるセッション固定攻撃の例について、設問に答えてください。

攻撃者がWebサイトをブラウザして、ログイン前のJSESSIONIDを受け取ります。次に、このJSESSIONIDをリンクに埋め込み、電子メールで被害者に送信します。被害者はリンクを

クリックしてログインを行います。ただし、このとき利用するJSESSIONIDは、攻撃者にとって既知のJSESSIONIDになります。このため、攻撃者は被害者になりすますことができます。

この脅威を最も低減させるものを選択してください。

- A. 電子メールに埋め込まれたリンクをクリックしないようにユーザを指導する。
- B. 各ユーザが認証するたびに、アプリケーションは新しいJSESSIONIDを割り当てるようにする。
- C. ツールではなく人間がアクセスしていることを確認するために、テキストを不明瞭に表示した画像を示し、読み取ったテキストを入力させる。
- D. JSESSIONIDの有効期限を極力短くするようにアプリケーションサーバを設定変更して、攻撃の機会を低減させる。

9) JAVA EEの宣言的アクセスコントロールをweb.xmlファイルでセットアップする場合について、誤った記述はどれでしょうか。

- A. web.xmlファイルに宣言されたリソースとロールとの間のアクセスを制限するため、開発者は継続してisUserInRoleメソッドを使用しなければならない。
- B. ワイルドカードパターンを使用して、複数の異なるリソースに対するアクセス制限を一度にセットアップできる。
- C. 複数のロールをリンクさせることで、ロールを他のロールと等価にするか、またはスーパーセットにすることができる。
- D. EJB内の個々のメソッドを特定のロールに制限できる。最終的には、そのメソッドにアクセスできるエンドユーザを制限可能となる。

10) 下記コードにはfinallyブロックが必要です。その理由を選択してください。

```
File fid = new File("C:/testfile");
try{
FileInputStream fis = new FileInputStream( fid );
...
fid.close();
} catch ( Exception e ) {
    Logger.log("Exception accessing file.");
}
```

- A. 例外が発生した場合に、プログラムでデバッグ情報にアクセスできるようにするため。

- B. JVM内の他のコードによる同じリソースへのアクセスを防ぐことができるため。
- C. `fid.close()`関数を実行する最適な場所であるため。
- D. このコードセクションで発生する可能性のある、あらゆるタイプの例外をプログラムで確実に検出できるようにするため。

11) 下記のコードは、システムオブジェクトにアクセスするWebユーザのアクセスコントロールチェックを実行します。このメカニズムを大幅に強化できる手段を選択してください。

```
//get account for which user wants details from the Request
String rqstdId = request.getParameter("ACCNT_ID");

//check that user is privileged to view the requested object
boolean accessAllowed = false;
String[] userAccountList = getAccountIds(user);
for (int i=0; i<userAccountList.length; i++) {
    if (userAccountList[i].equals( rqstdId ) accessAllowed
    = true;
}

if ( accessAllowed == true ) {
    displayDetails( rqstdId );
} else { logAccessControlFailure(); }
```

- A. Webインタフェースを設計し直して、要求されたアカウントIDごとに固有のURIを使用できるようにし、アカウントIDがパラメータとして扱わないようにする。これにより、ユーザによるパラメータの改ざんを回避する。
- B. ユーザが実際のアカウントIDのリファレンスを送信しないようにする。代わりに、そのユーザがアクセス権を持つ実際のアカウントにマップされたインデックスから選択させる。これにより、ユーザによる実際のアカウントIDの改ざんを回避する。
- C. `displayDetails()`のコードを単一のメソッドにする。アクセスコントロールチェックはここで実行されるため、リソースへの実際のアksesも同様に実行する。これにより、アクセスチェックでリソースが直接コントロールされていることをコード監査で確認できる。
- D. `boolean`変数`accessAllowed`の初期値を`true`にする。`false`であることが判明してから、`false`を設定する。これにより、オブジェクトへのアクセスが意図せずに制限されてしまうのを回避する。

サンプル問題

- 1) D
- 2) A
- 3) B
- 4) C
- 5) C
- 6) D
- 7) A
- 8) B
- 9) A
- 10) C
- 11) B



本資料は、必要に応じて更新が行われる可能性があります。
最新版はwww.sans-ssi.orgよりダウンロードしてください。
ご意見ご要望はspa@sans.orgまでお送りください。

GSSP (GIACセキュアソフトウェアプログラマ)

Java/Java EE実装の課題

www.sans.org

2007年3月26日

編集者:Ed Tracy、Booz Allen Hamilton

寄稿者/校閲者:Ryan Berg、Ounce Labs; Andrew Van der Stock、OWASP、およびAspect Security

課題1 –入力値処理

Javaプログラマは、各インタフェースから入力値(コマンドライン引数、環境変数、入力ストリームなど)を読み取り、適切に検証および処理するプログラムを作成できなければならない。これらの入力ソースは実際にはユーザ入力であったり、その他の信頼できないソースであったりするため、入力値の処理にはセキュリティが影響する。

- **規則1 : 入力値検証の原則**

Javaプログラマは、HTTPリクエスト、アプレットソケット、直列化ストリーム、設定ファイル、バックエンドのデータストアなど、どのインタフェースからであっても、入力値は信頼できないものであることを理解する必要がある。また、ホワイトリスト方式とブラックリスト方式を理解し、それぞれを使用する場合のトレードオフを把握しておく。

- **規則2 : 入力値検証のソース**

Javaプログラマは、Javaアプリケーションの一般的な入力ソースを把握しておく必要がある。これにより、入力値検証が正当であるかどうかを判断するために、特定のデータの信頼度をいつ確認すべきかを認識できる。

- **規則3 : 入力値検証手法**

Javaプログラマは、Stringなどの一般的なデータ型と、一般的でない入力構造を検証する方法を理解しなければならない。正規表現、`doValidate()`、および入力値検証のためのJava/Java EEツールを熟知している必要がある。

課題2 – 認証およびセッション管理

Javaアプリケーションプログラマは、JavaおよびJava EEの認証APIの基本的な知識を有しており、ローカルアプリケーションおよびリモートアプリケーションの認証方針を熟知している必要がある。本試験においては、セッション管理が、エンドユーザの認証識別情報を長期にわたり維持するプロセスとして見なされる。Javaプログラマは、一般的な認証およびセッション管理操作を適切に保護するため、これらの操作に対する脅威を理解する必要がある。

- **規則1： 認証が必要な状況**

Javaプログラマは、認証がエンドユーザだけでなく、サードパーティーのサービス、バックエンドシステムなどにも必要であることを理解する必要がある。

- **規則2： 認証保護**

Javaプログラマは、暗号化と証明書を使用して各種認証プロセスを保護する方法を把握しておく必要がある。たとえば、機能強度、認証情報の期限、回復/リセット、再認証などについて理解する。

- **規則3： セッションの保護**

セッショントークンを保護するため、Javaプログラマはさまざまな要素(暗号化、機能強度、トークン存続期間、再発行など)の影響を理解する必要がある。

- **規則4： 認証手法**

Javaプログラマは、JavaおよびJava EEで広く利用されている認証手法とAPIに精通していなければならない。たとえば、Java認証承認サービス(JAAS:Java Authentication and Authorization Services)、バックエンド信用証明書ストレージ、および各種フロントエンド認証手法(証明書、フォーム、基本認証など)が挙げられる。認証手法とAPIについて十分な知識を持つプログラマであれば、各手法における脅威ならびにトレードオフを理解できる。

- **規則5： 認証への対応**

Javaプログラマは、一般的なAPIを使用する場合のサービスと保護機能について、利用できるものとできないものを完全に把握している必要がある。たとえば、最大セッション長、再認証、および暗号化は、自動的に有効になる保護機能ではない。

課題3 – アクセスコントロール (承認)

Javaアプリケーションプログラマは、ユーザデータの機密性を保証できるアプリケーションを開発できなければならない。また、このようなアプリケーションでは、特定の機能についてユーザが実行できないように制限する必要もある。アクセスコントロールは積極的に実施されなければならないが、省略されたりバックエンドシステムに依存したままであったりしてはならないことを、開発者は理解する必要がある。

- **規則1： リソースへのアクセスの制限**

Java開発者は、システムリソースに対する明確かつ完全なアクセスコントロールポリシーの必要性を理解しなければならない。たとえば、ユーザデータオブジェクトはデータ所有者のみがアクセスする、などのポリシーである。

- **規則2：機能へのアクセスの制限**

Java開発者は、権限付き関数や権限付きURIなどの機能へのアクセスを制限する必要性を理解する必要がある。

- **規則3：宣言型のアクセスコントロール**

設定ファイルを使用したアクセスコントロールに対応している一般的なAPI(およびそれらのトレードオフ)を理解しておく。

- **規則4：プログラミングによるアクセスコントロール**

Java開発者は、開発したカスタムコードにてアクセスコントロール検査を手動で実行する方法とタイミングを理解する必要がある。

- **規則5：JAAS**

Java開発者は、Java認証承認サービスを使用したアクセスコントロールの実装方法を理解していなければならない。

課題4 – Javaデータ型とJVM管理

Javaプログラマは、基底のデータ型およびJava固有のメモリ管理とセキュリティとの関係を理解する必要がある。

- **規則1：java.lang.String**

Javaプログラマは、Stringクラスの不変性と、Stringオブジェクトの比較方法を完全に理解しておく必要がある。

- **規則2：IntegerおよびDoubleのオーバーフロー**

Javaプログラマは、Javaの数値データ型の制約事項と、これに伴うセキュリティへの影響を理解する必要がある。

- **規則3：ガベージコレクタ**

Javaプログラマは、Javaガベージコレクタの仕組みと、これによるセキュリティへの影響を理解する必要があります。

- **規則4：ArrayListとVector**

Javaプログラマは、ArrayListとVectorの違いと、これに伴うセキュリティ上の考慮事項を理解する必要があります。

- **規則5：Classのセキュリティ**

Javaプログラマは、アクセス修飾子、final修飾子、クラス比較、直列化、クローン機能、内部クラスについて熟知している必要がある。

- **規則6：コード権限**

Javaプログラマは、コードの権限の管理方法とさまざまな保護ドメインについて理解しなければならない。これには、セキュリティマネージャとそのポリシーファイルの理解が含まれる。

課題5 – アプリケーションエラーとログ記録

すべてのJavaアプリケーションプログラマは、アプリケーションエラーを適切に処理できなければならない。

- **規則1：例外処理**

Javaアプリケーション開発者は、アプリケーションとシステムの例外を適切に処理するため、Javaのtry/catch/finally構造を理解する必要がある。

開発者は、例外検出時に、その例外の特性に基づいてどの程度の情報をログに記録するかを決定しておく。

- **規則2：ログへの記録**

開発者は、セキュリティ関連イベント(ログイン、ログオフ、認証情報の変更など)のログへの記録の原則を理解しなければならない。また、Javaのロギングパッケージであるjava.util.loggingに精通している必要がある。

- **規則3：エラー処理の設定**

Java EE開発者は、HTTP 404エラーと500エラーに対しデフォルトエラーページを戻す設定を熟知している必要がある。

課題6 – 暗号化サービス

Javaプログラマは、暗号化を用いて重要なデータを保護する方法を理解しなければならない。

- **規則1：通信の暗号化**

Javaアプリケーション開発者は、Java Secure Sockets Extension (JSSE)パッケージと、Java EEアプリケーションのSSL通信の設定方法を熟知している必要がある。開発者はまた、開発するアプリケーションの外部リンクのうち、どれが暗号化により保護されるべきかについても把握している必要がある。

- **規則2：保管データの暗号化**

Java開発者は、機微情報を暗号化して保管する方法を理解しなければならない。

課題7 – 並行性とスレッド処理

Javaプログラマは、マルチスレッドプログラムの適切な構造化方法を理解する必要がある。

- **規則1：レースコンディション**

すべてのJavaアプリケーション開発者は、レースコンディションと、これがシステムのセキュリティに及ぼす影響を理解しなければならない。

これには、複数のスレッドからアクセス可能なセキュリティ関連情報の、キャッシュの回避などがある。

- **規則2：シングルトンと共有リソース**

Java開発者は、Javaでのシングルトンパターンの実装方法と、複数スレッドからアクセスされるその他のリソースの保護方法を理解する必要がある。

課題8 – 接続パターン

Javaプログラムは、他のアプリケーションとセキュアな方法でインタフェースをとることができなければならない。開発者は、パラメータライズドクエリ、出力エンコード、およびフェイルセーフな接続パターンについて熟知している必要がある。

- **規則1：パラメータライズドクエリ/PreparedStatement**

Javaプログラマは、動的クエリの使用に伴うセキュリティリスクと、ユーザ入力に基づいてデータベースと適切かつセキュアに対話するためにPreparedStatementを安全に使用方法を理解する必要がある。

- **規則2：出力エンコード**

Javaプログラマは、ユーザインタフェースでのデータ表示に出力エンコードを使用する方法と使用する状況を理解しなければならない。これは、出力エンコードはUIインジェクション攻撃(クロスサイトスクリプティング攻撃など)に対する主な軽減策であるためである。

- **規則3：フェイルセーフな接続パターン**

Javaプログラマは、リソースリークを防ぐため、Javaのtry/catch/finallyを使用して接続パターンを適切に形成する必要がある。外部システムとの接続処理中に発生したエラーが原因で、リソースリークが発生することがある。

課題9 – その他

- **規則1：クラス/パッケージ/メソッドのアクセス修飾子**

すべてのJavaプログラマは、Javaアクセス修飾子(public, private, protected)を使用してクラスのメンバーとメソッドを保護する方法を理解する必要がある。

- **規則2：クラスファイルの保護**

Javaプログラマは、JARシーリングの使用法を理解しなければならない。

- **規則3：Java EEフィルタ**

Java EEプログラマは、Java EEフィルタと、これらのフィルタを使用して前述の課題の多くに対応する方法に精通している必要がある。

付録：参考文献

以下のリストは、プログラマーやアプリケーション開発者、セキュリティ従事者にとって、セキュアプログラミングについてさらなる学習の一助となる資料の例です。アセスメントに役立つ資料も含まれます。これは、試験対策の関する完全なリストでも GIAC からの推薦リストでもありません。単に、学習される方の興味を引くきっかけになり得るものとして紹介しています。

ソフトウェアセキュリティに関する参考書

19 Deadly Sins of Software Security

Michael Howard, David LeBlanc, John Viega

Building Secure Software: How to Avoid Security Problems the Right Way

John Viega, Gary McGraw

Exploiting Software: How to Break Code

Gary McGraw, Greg Hoglund

Foundations of Security: What Every Programmer Needs to Know

Neil Daswani, Christoph Kern, Anita Kesavan

Introduction to Computer Security

Matt Bishop

J2EE & Java: Developing Secure Web Applications with Java Technology (Hacking Exposed)

Art Taylor, Brian Buege, Randy Layman

Secure Coding in C and C++

Robert Seacord

Secure Coding: Principles and Practices

Ken Van Wyk, Mark Graff

Hacking Exposed: Web Applications

Scambray, Shema, Sima

Secure Programming Cookbook for C and C++

John Viega, Matt Messier

Security and Usability

Simson Garfinkel, Lori Faith Cranor

Software Security: Building Security In

Gary McGraw

The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities

Mark Dowd, John McDonald, Justin Schuh

The Security Development Lifecycle

Michael Howard, Steve Lipner

Web Security, Privacy & Commerce, Second

Simson Garfinkel, Gene Spafford

Writing Secure Code, Second Edition

Michael Howard, David C. LeBlanc

[ソフトウェアセキュリティに関する Web サイトや Podcast](#)

情報処理推進機構 (IPA) 「安全なウェブサイトの作り方」

<http://www.ipa.go.jp/security/vuln/websecurity.html>

OWASP - Open Web Application Security Project

http://www.owasp.org/index.php/Main_Page

Microsoft Corporation - Security Developer Center

<http://msdn2.microsoft.com/en-us/security/aa570401.aspx>

MITRE - Common Weakness Enumeration (CWE)

<http://cwe.mitre.org/>

CERT - Secure Coding Initiative

<http://www.cert.org/secure-coding/>