CWE/SANS最も危険なプログラミングエラーTOP 25

最も危険なプログラミングエラー25項目とその解決策に関する共通認識を専門家が発表
この認識が組織のソフトウェア購入基準を変化させる

プロジェクトマネージャ:Bob Martin, MITRE
問い合わせ先:top25@sans.org
日本語訳:NRIセキュアテクノロジーズ株式会社

米国時間2009年1月12日、ワシントンDCで米国をはじめとする各国のサイバーセキュリティ会社に所属する30名以上の専門家が、セキュリティバグの原因になったり、サイバースパイ活動やサイバー犯罪につながる恐れのある最も危険なプログラミングエラー25項目のコンセンサスリストを共同発表した。驚くべきことに、こうしたエラーのほとんどがプログラマにきちんと理解されていない。コンピュータサイエンスの課程でもこうしたエラーの回避策を教えておらず、商用ソフトウェアを開発している企業でさえその存在を見逃してしまう場合がある。

これらのエラーが与える影響は広範囲に及ぶ。2008年には25項目中たった2つのエラーが150万件を超えるWebサイトのセキュリティ侵害を引き起こした。この種のセキュリティ侵害は、問題のWebサイトにアクセスしたユーザのコンピュータを、攻撃者が容易に操作できるゾンビコンピュータへと次々に変身させてしまう。

このプロジェクトでは、次の専門家および組織から多大なご支援をいただいた。いずれもセキュリティ分野の権威とされる専門家であり、SymantecやMicrosoftをはじめ、DHSのNational Cyber Security Division、NSAのInformation Assurance Division、OWASP、Japanese IPA、University of California at Davis、Purdue Universityなどの主要な組織の代表者に協力いただいた。このTOP 25エラーはMITREおよびSANS Instituteが中心となって作成されたが、プロジェクト自体に推進力を与えたのはNational Security Agencyであり、MITREのプロジェクトに参加したエンジニアへの米国Department of Homeland SecurityのNational Cyber Security Divisionによる財政面での支援である。NSAのInformation Assurance DivisionおよびDHSのNational Cybersecurity Divisionはこれまで、行政機関や基幹的な国家インフラストラクチャが購入するソフトウェア製品のセキュリティ強化を図るべく、政府のリーダーとして主導的な役割を果たしてきた。

今回の試みで注目すべき点は、専門家の議論が白熱したにもかかわらず、またたく間に統一見解が出されたことだ。SANS DirectorであるMason Brownは、次のように話している。「プログラミングエラーについて大筋の合意が得られた。これをいかに修正するかが今後の課題だ。まずすべてのプログラマにTOP 25エラーを排除したコードの記述方法を周知させた上で、各プログラミングチームに、問題の検出、修正、または回避を図るためのプロセスを導入する。また、自動化されたツールでも確認できるように、コードにこれらのエラーが無いことを立証するためのツールの使用を条件付ける必要がある」

Office of the Director of National Intelligenceは、TOP 25への支援について次のように説明している。「ハードウェア/ソフトウェア製品の完全性は、サイバーセキュリティで不可欠の要素であると考えている。セキュリティを強化したソフトウェアの作成は、連邦政府や国家の基幹的なインフラストラクチャが市販製品に依存して業務を運営するにあたり、システムやネットワークのセキュリティを確保する上で不可欠な条件である。TOP 25は、米国における総合的なセキュリティを指導する際の重要な要素となる。当機関では、こうした努力を歓迎すると同時に、サイバー教育など、さまざ

まな場面を通じてこのツールの活用を奨励している」

これまでのガイダンスは、プログラミングエラーが原因で生じる「脆弱性」を中心としたものだった。これはこれで意義がある。これに対しTOP 25は、脆弱性を作ってしまう開発者が実際に犯しやすいプログラミングエラーを重点的に取り上げている。また、TOP 25のWebサイトでは、エラー回避に関する信頼できる情報が詳しく提供されている点も重要である。『US National Strategy to Secure Cyberspace』の主要著者であり、Software Assurance Forum for Excellence in Code (SAFECode)のエグゼクティブディレクタを務めるPaul Kurtz氏は、このリストの効用を次のようにたとえている。「TOP 25を利用すれば、空き巣に合ってから警察の捜索を待つのではなく、侵入される前にドアにしっかり鍵をかけられるようになる」

今回の発表の内容

TOP 25エラーリストの作成に寄与した専門家と組織

Robert C. Seacord, CERT
Pascal Meunier, CERIAS, Purdue University
Matt Bishop, University of California, Davis
Kenneth van Wyk, KRvW Associates
Masato Terada, Information-Technology Promotion Agency (IPA), (Japan)
Sean Barnum, Cigital, Inc.
Mahesh Saptarshi and Cassio Goldschmidt, Symantec Corporation
Adam Hahn, MITRE
Jeff Williams, Aspect Security
Carsten Eiram, Secunia
Josh Drake, iDefense Labs at VeriSign, Inc.
Chuck Willis, MANDIANT
Michael Howard, Microsoft
Bruce Lowenthal, Oracle Corporation
Mark J. Cox, Red Hat Inc.
Jacob West, Fortify Software
Djenana Campara, Hatha Systems
James Walden, Northern Kentucky University
Frank Kim, ThinkSec
Chris Eng and Chris Wysopal, Veracode, Inc.
Ryan Barnett, Breach Security
Antonio Fontes, New Access SA, (Switzerland)

Mark Fioravanti II, Missing Link Security Inc.
Ketan Vyas, Tata Consultancy Services (TCS)
Lindsey Cheng, Ian Peters and Tom Burgess, Secured Sciences Group, LLC
Hardik Parekh and Matthew Coles, RSA -Security Division of EMC Corporation
Mouse
Ivan Ristic
Apple Product Security
Software Assurance Forum for Excellence in Code (SAFECode)
Core Security Technologies Inc.
Depository Trust & Clearing Corporation (DTCC)
第1回OWASP ESAPI Summitの作業グループ
National Security Agency (NSA) Information Assurance Division
Department of Homeland Security (DHS) National Cyber Security Division

MITREのCWE Project LeaderであるRobert Martin氏は、次のように語って以上のような貢献を歓迎している。「このように熱心なセキュリティの専門家による協力とエネルギーを結集した成果として、確度の高い権威あるリストが完成したことは大変喜ばしい。すばらしいことだ」

TOP 25エラーの利用方法

TOP 25エラーは、主に次の4つの面で影響を及ぼす。

- ソフトウェアのユーザが、より安全なソフトウェアを購入できるようになる。
- プログラマに、自分が作成するソフトウェアのセキュリティを一貫した方法で測定するツールが提供される。
- 大学ではより確信をもって安全なコーディングを指導できるようになる。
- 雇用者は、より安全性の高いコードを記述できるプログラマを確保できる。

まず、ソフトウェアのユーザは、より安全なソフトウェアを購入できるようになる。

ユーザの求めに応じて、ソフトウェアベンダーは販売中の製品のコードに25項目のプログラミングエラーが含まれていないことを文書で証明する必要が出てくる。保証の対象が、プログラミングエラーの修正やこうしたエラーが原因で生じる損害に対してベンダーが負うべき責任という領域に移行する。ニューヨーク州やその他の州の行政機関が策定中の標準調達基準ではすでに、TOP 25エラーを適用する方向で調整が進められている。将来的には多国籍向けの共通基準プログラムにも、米国政府が購入する製品のコードにTOP 25エラーが含まれていないことを保証するアプローチの一つとしてTOP 25が採用されるだろう。

また、プログラマには、自分が作成するソフトウェアのセキュリティを一貫した方法で測定するツールが提供される。

ソフトウェアテストツールでは、評価基準の中にTOP 25を組み込んで、テスト対象となるソフトウェアのコードの安全性を採点できるようになる。同1月12日、この発表に並行する形で、ある大手ソフトウェアテストベンダーによる発表も行われた。同社のソフトウェアではTOP 25に分類される大半のエラーについて、その有無の検証とレポートが行えるようになるという。アプリケーション開発チームでは、開発プロセス中にこうしたテストソフトウェアを使用するようになる。

大学ではより確信をもって安全なコーディングを指導できるようになる。

大学や、プログラマを育成するその他の教育機関では、重大なプログラミングエラーを回避するための方法を指導するにあたり、カリキュラムの基礎としてTOP 25エラーを取り入れることができる。TOP 25の策定に関与した大学、University of California at Davisではすでに学生が作成したソフトウェアを対象として、重大なセキュリティの脆弱性に結び付く主なプログラミングエラーの有無を検証するセキュアコーディングクリニックが開設されている。このクリニックではTOP 25の基準に従って、検証する際エラーに優先順位を付けている。その他、セキュアコーディングクリニックを見習った類似機関の開設を進めている大学もある。

雇用者は、より安全性の高いコードを記述できるプログラマを確保できる。

雇用者側は、プログラミングの人材を雇用したり、アウトソーシングしたりする場合に、スキルの評価や育成の指針としてTOP 25エラーリストを利用できる。セキュアコーディングスキルを測定するための一般的な査定ツールとしてすでにGSSP (GIAC Secure Software Programmer) を導入している企業は100を超える。GSSP試験には、TOP 25エラーの検出や除去に必要なプログラミング知識が全面的に反映されており、その修得度を集中的に評価する目的で実施される。GSSPの詳細についてはhttp://www.sans-ssi.org/ を参照のこと。プログラマの数が500人を超える組織では、最大100名のプログラマを対象としてセキュアコーディングスキルを内々にしかも経費負担なしで査定できる。詳細の問い合わせ先: spa@sans.org

C/C++、Java、および.NETの各言語におけるセキュアコーディングスキルを指導するコースも用意されている。詳細の問い合わせ先: http://www.sans-ssi.org/courses/

TOP 25エラーはなぜ重要か

プロジェクトの参加者に、この試みに対して相当な時間と専門知識を投じる価値があると思った理由をたずねてみた。その回答の一部を以下に示す。本文書を読み終えた読者は、ここで示す回答より多くの重要さに気付くはずだ。

National Security AgencyのInformation Assurance Directorate

「サイバースパイ活動やサイバー犯罪に結び付く可能性のあるプログラミングエラーのリストの公表は、当機関のネットワークとテクノロジーの脆弱性を管理するための重要な第一歩となる。数千個所にも及ぶ個別の脆弱性に逐一対応するのではなく、全般的な根本原因を元に脆弱性を生み出している比較的数の少ないソフトウェアの弱点をより重点的に解決できるような方向を目指さなければならない。このようなリストがあれば、ソフトウェア開発に伴う実践、ツール、および要件に関連する改善の目標が設定でき、ライフサイクルにおける早期の段階からこうした問題を管理できるようになるため、大規模でかつコスト効果に優れた解決が可能になる」

-Tony Sager, National Security Agency's Information Assurance Directorate

US Department of Energy:

「CWE/SANS TOP 25の構想はきわめて貴重なもので、多くの組織にとって、ソフトウェアセキュリティの問題に対応するための具体的な方策となる」

-Michael Klosterman, SCADA Operations, Western Area Power Association, US Department of Energy

Depository Trust:

「CWE-SANS TOP 25エラーは、組織がソフトウェアセキュリティに対するリスクベースのアプローチに基づいて環境内の具体的な脆弱性と、業界で権威ある専門家が下したリスクに関する総合的な見解を比較する際に

利用できる不可欠なツールとなる」

-Jim Routh, CISO, The Depository Trust & Clearing Corporation

Microsoft:

「2009 CWE/SANS TOP 25プログラミングエラープロジェクトは、ソフトウェア開発者が問題の認識、防止、および修正を実践する上で、最も重要なセキュリティの脆弱性は何かを特定する際の優れたリソースとなる」

-Michael Howard, Principal Security Program Manager, Security Development Lifecycle Team, Microsoft Corp.

OWASP Foundation:

「弱点の種類が700を超えるような数千ものインスタンスを含む大規模なアプリケーションポートフォリオを扱うとなると、どこから着手すべきかを判断するだけでも大変な作業になる。適切に行えば、CWE TOP 25エラーを一掃することにより、セキュリティが大幅に強化できるだけでなく、ソフトウェアの開発コストを大幅に削減できる」

-Jeff Williams, Aspect Security CEO and The OWASP Foundation Chair

Symantec:

「2009 CWE/SANS TOP 25プログラミングエラーには、アプリケーションソフトウェアに見られる問題が反映されていることに加え、ソフトウェアセキュリティの継続的な改善を目指した実行可能な方向性が示されている」

-Wesley H. Higaki, Director, Software Assurance, Office of the CTO, Symantec Corporation

Software Assurance Consortium:

「このリストは、消費者を保護する存在として、あらゆるユーザに向けてセキュリティを提供するための大きな第一歩となる。あらゆるソフトウェア製品の日常使用に与える影響に焦点を合わせることにより、多様なレベルのソフトウェアセキュリティを認識できるようになる。CWE/SANS TOP 25は、SwACのツールボックスの機能を強化する一方、業界と行政の連携を通じてあらゆるソフトウェア製品のセキュリティと信頼性の改変を目指す当機関が使命を遂行するにあたって大きな支援となる」

-Dan Wolf, Director, Software Assurance Consortium.

EMC:

「TOP 25リストは、プログラマをはじめ、ソフトウェアの設計や開発に携わるすべての担当者が利用できる強力なツールとなる。このようなリストが存在するだけで、ソフトウェアがより効果的に保証されるようになる」

-Dan Reddy, Consulting Product Manager, EMC Product Security Office

Purdue:

「CWE TOP 25は、最も厄介なプログラミング上の誤りを対象としているため、脆弱性の発生や国家レベルでの露出が抑制されると同時に、パッチへの不必要な依存を低減できる効果があるため、注目すべき方策となる」

-Pascal Meunier, CERIAS, Purdue University

Secunia:

「TOP 25は、ソフトウェアの脆弱性を生む可能性のある一般的なコーディングエラーを最も効果的にまとめたリストであることは間違いない。セキュリティ業界における数多くの専門家からのフィードバックを元に作成したこのリストは、重大度や影響範囲といった選択基準を重視しているため、今日のアプリケーションに持ち込まれる最も重大なエラーについても幅広く対応している。TOP 25は、おもしろく読みやすい構成でまとめられているため、一般的なコーディングエラーについて理解が深まるほか、こうしたエラーを回避する方法がよくわかる。ソフトウェアの設計にかかわる人が、これまでに犯してきた過ちを2009年に再び繰り返さないよう、強く推奨される資料である」

-Carsten Eiram, Chief Security Specialist, Secunia.

Ken van Wyk:

「このプログラミングエラーのリストが業界に与える効果は計り知れない。コードに含まれる一般的なセキュリティ上の欠陥を認識した上で全体的な状況の理解が深まる。ちょうどOWASP Top 10によって、こうした欠陥への攻撃がよく理解できた状況に似ている」

-Kenneth R. van Wyk, KRvW Associates, LLC

Veracode:

「セキュリティの問題について優先順位の付いたリストは、限られた資源と納期という制約の中でのソフトウェアセキュリティを現実的な側面から捉えるための開始点となる。TOP 25リストには、顧客がソフトウェアを使用する前に開発者が一掃しておくべき最低限のコーディングエラーが記載されている」

-Chris Wysopal, Co-Founder and CTO of Veracode, Inc.

Core Security Technologies:

「これは、ソフトウェアセキュリティにかかわる問題の特定、予防、修正、緩和に伴う実践的な応用を中心として、セキュリティの脆弱性や欠陥に関する体系的な分類を図った初の試みである。現在、緊急性と関連性が最も高いソフトウェアセキュリティ上の問題に対応するための合理的な手法を求めているソフトウェア開発やセキュリティの業界にとって、共通の言語を確立するために不可欠なものとして長い間待ち望まれていた第一歩にほかならない」

-Ivan Arce, CTO of Core Security Technologies Inc.

Breach Security:

「CWE/SANS TOP 25リストは、今日の攻撃を許容している根本原因に優先順位を付け、修正を図るために組織が効果的に利用できる戦術的な資源となる。安全なコーディングの基本原理を要約した基礎として、開発者向けの必読資料となる」

-Ryan C. Barnett, Director of Application Security Research, Breach Security

McAfee:

「2009 CWE/SANS TOP 25プログラミングエラーの試みはまさに正鵠を射ている。最も重要な問題についてソフトウェア開発者を教育し、セキュリティバグを避ける方法を具体的に説明するこの試みにより、プログラマがコードの問題を解決しておくことで、セキュリティの問題にまで発展するのを阻止できるようになる」

-Kent Landfield, Director, Risk and Compliance Security Research, McAfee, Inc.

Ounce Lab:

「このリストはソリューションの活性化を図るための手段として利用できる。2009年は実行の年として、長年にわたって現存している問題の解決を目指すべきだ。市場には、ありがちなエラーへの対処に役立つという余計なソリューションが、あまりにも溢れている。このリストは、ソフトウェアの安全性を確保するために今すぐにでも利用できる。過去何年かを振り返ってみると、明日では遅すぎることがわかる」

-Ryan Berg, Co-Founder and Chief Scientist, Ounce Labs

Grammatech:

「ソフトウェア中のバグは当社にとって悩みの種であり、ビジネス上あってはならないものである。かと言って避けて通ることもできない。特に製品が市場に出た後、どのバグが最も重要なのかを理解することは難しく、コストもかかる。CWE/SANS TOP 25の試みは、起こり得るさまざまな種類の欠陥に関する意識を高めるほか、プログラマがアプリケーションの品質やセキュリティにとって最も重要な事項に焦点を絞る上で役立つ」

-Paul Anderson -Vice President of Engineering, Grammatech Inc.

TOP 25に含まれるエラー

TOP 25エラーは、次の3つのカテゴリに分類される:

- Insecure Interaction Between Components(セキュアでないコンポーネント間通信)（9 項目）
- Risky Resource Management(リソース管理の問題)（9 項目）
- Porous Defenses(不完全な防御策)（7 項目）

リスト中の「詳細」をクリックすると、MITRE CWEの対応するスポットを確認できる。リンク先には次の項目が掲載されている。

- すべての CWE エントリデータへのリンク
- 弱点の拡散度合いとその影響力を表すデータフィールド
- 修復コスト
- 検出のしやすさ
- 攻撃の頻度と攻撃者の意識
- 関連する CWE エントリ
- この弱点に関連する攻撃パターン

TOP 25エラーサイトの各エントリにも、開発者が脆弱性の緩和や除去に向けて実施できるさまざまな防止対策や修復方法が紹介されている。

カテゴリ: Insecure Interaction Between Components(セキュアでないコンポーネント間通信)

CWE-20: 不適切な入力の妥当性チェック

これは、健全なソフトウェアが犠牲となる原因の第一位であるため、所定の基準に従った入力を条件付けておかないと、自ら災難を招くことになる。

詳細: http://cwe.mitre.org/top25/#CWE-20

CWE-116: 不適切なエンコード、または出力のエスケープ

コンピュータは、ユーザが本来意図するところではなく、実際の入力内容に従って動作する。不適切な出力エンコードは、不当入力の妥当性チェックに比べ軽視されやすいが、現在猛威をふるっている大半のインジェクション攻撃の原因となっている。

詳細: http://cwe.mitre.org/top25/#CWE-116

CWE-89: SQLクエリ構造が保護されない(「SQLインジェクション」)

ユーザがデータベースとのやり取りに使用するSQLを、攻撃者が操作できるようになる。

詳細: http://cwe.mitre.org/top25/#CWE-89

CWE-79: Webページ構造が保護されない(「クロスサイトスクリプティング」)

クロスサイトスクリプティング(XSS)は、Webアプリケーションで最も頻繁に起きる、容易に解決できない危険度の高い脆弱性の一つである。

詳細: http://cwe.mitre.org/top25/#CWE-79

CWE-78: OSコマンド構造が保護されない(「OSコマンドインジェクション」)

オペレーティングシステム上の別のプログラムを呼び出すときに、プログラムの実行時に生成するコマンド文字列に対し信頼性のない入力を取り入れると、攻撃を呼び込む原因となる。

詳細: http://cwe.mitre.org/top25/#CWE-78

CWE-319: 機密情報の平文転送

個人的なデータや認証の証明など、機密情報をネットワークで送信する場合、この情報はさまざまなノードを経由して—。

詳細: http://cwe.mitre.org/top25/#CWE-319

CWE-352: クロスサイトリクエストフォージェリ(CSRF)

クロスサイトリクエストフォージェリでは、攻撃対象者が意図に反してユーザのサイトに宛てた要求を起動させられる。スクリプティングとWebの全般的な動作形式によって攻撃対象者は—。

詳細: http://cwe.mitre.org/top25/#CWE-352

CWE-362: 競合状態

攻撃者が故意に競合状態に乗じて混乱を起こしたり、アプリケーションから貴重な情報を暴露させたり—。

詳細: http://cwe.mitre.org/top25/#CWE-362

CWE-209: エラーメッセージ情報の漏洩

余計な情報を含むエラーメッセージを使用すると、ここから攻撃者に機密情報が開示され、ソフトウェアが悪用されることがある。機密情報の範囲は多岐にわたり—。

詳細: http://cwe.mitre.org/top25/#CWE-209

カテゴリ: Risky Resource Management(リソース管理の問題)

CWE-119: メモリバッファの範囲内での操作が制限されない

バッファオーバーフローには「ある容器に、許容量を超える物質を入れようとすると、混乱を招く」という物理学の法則を示唆する自然の摂理があてはまる。

詳細: http://cwe.mitre.org/top25/#CWE-119

CWE-642: クリティカルな状態データの外部制御

データベースのオーバーヘッドを伴わずにユーザの状態データを保存する方法は数多くある。しかし、攻撃者が操作できるような場所にこのデータを保存すると—。

詳細: http://cwe.mitre.org/top25/#CWE-642

CWE-73: ファイル名やパス名の外部制御

ファイル名の作成時に外部の入力を使用するのは、危険を伴う。

詳細: http://cwe.mitre.org/top25/#CWE-73

CWE-426: 信頼性のない検索パス

リソースの検索パスが攻撃者の制御下に置かれると、攻撃者が選んだリソースにパスを変更することができる。これによりソフトウェアは、時期や適正を考えずにリソースにアクセスするようになり—。

詳細: http://cwe.mitre.org/top25/#CWE-426

CWE-94: コード生成が制御されない(「コードインジェクション」)

開発の手間を考慮した場合、数行のコードを多様な機能に応用せざるを得ない場合がよくある。コードを動的に管理できれば—。

詳細: http://cwe.mitre.org/top25/#CWE-94

CWE-494: 完全性検査なしのコードダウンロード

あるコードをダウンロードして実行する場合、このコードのソースに悪意がないことを前提としていることは誰にでもわかる。しかし攻撃者はさまざまな仕掛けを駆使して—。

詳細: http://cwe.mitre.org/top25/#CWE-494

CWE-404: リソースの不適切なシャットダウンまたはリリース

貴重なシステムリソースのライフサイクルが終わりに近付いてくると—。

詳細: http://cwe.mitre.org/top25/#CWE-404

CWE-665: 不適切な初期化

一日の初めに健康的な朝食を摂ることが大切なように、適切な初期化は—。

詳細: http://cwe.mitre.org/top25/#CWE-665

CWE-682: 計算の誤り

攻撃者が数値計算に使用する入力についてある程度制御できるようになると、この弱点が脆弱性に発展する可能性がある。これに伴い、セキュリティに関する誤った判断を下してしまうことがある。

詳細: http://cwe.mitre.org/top25/#CWE-682

カテゴリ: Porous Defenses(不完全な防御策)

CWE-285: 不適切なアクセス制御 (承認)

ソフトウェアのユーザが、許可された作業だけを行っていることを保証できなければ、攻撃者が不当な承認を利用して—。

詳細: http://cwe.mitre.org/top25/#CWE-285

CWE-327: 不完全、またはリスキーなアルゴリズムの使用

攻撃者が解読しにくくなるだろうと期待して、ユーザ独自の暗号化スキームを開発する場合がある。この種の自作の暗号化は、攻撃者にとって歓迎すべき—。

詳細: http://cwe.mitre.org/top25/#CWE-327

CWE-259: パスワードのハードコーディング

機密アカウントやパスワードをソフトウェアの認証モジュールにハードコーディングすると—。

詳細: http://cwe.mitre.org/top25/#CWE-259

CWE-732: 重要なリソースに対する安全性の低いアクセス権の割り当て

重要なプログラム、データストア、構成ファイルに、リソースへのアクセスが許可されるパーミッションが含まれている場合、攻撃者にとっては格好の—。

詳細: http://cwe.mitre.org/top25/#CWE-732

CWE-330: 不十分なランダム値の使用

セキュリティ機能で必要なランダム性が十分に確保されていないと、攻撃者には好都合な—。

詳細: http://cwe.mitre.org/top25/#CWE-330

CWE-250: 不要な特権による実行

有名な漫画のヒーロー「スパイダーマン」には「大いなる力には、大いなる責任が伴う」というモットーがある。ソフトウェアでは特定の機能の実行に特定の特権が必要になるが、必要以上の権限の行使はきわめてリスキーな—。

詳細: http://cwe.mitre.org/top25/#CWE-250

CWE-602: サーバサイドのセキュリティをクライアントサイドで適用

嗜好を凝らしたGUIの下にあるのはまさにコードの羅列である。攻撃者はクライアントのリバースエンジニアリングを行って、独自のクライアントを作り上げ、煩わしいセキュリティコントロールなどの使いにくい機能はすべて取り除き—。

詳細: http://cwe.mitre.org/top25/#CWE-602

TOP 25エラーの排除に向けて利用できるリソース

TOP 25エラーリストは定期的に更新され、SANSとMITREの各サイトに掲載される。

www.sans.org/top25

cwe.mitre.org/top25

MITREでは、米国Department of Homeland SecurityのNational Cyber Security Divisionの支援を受けてCWE (Common Weakness Enumeration) Webサイトが運営されている。このサイトでは、プログラミングエラー25項目の詳細な説明とともに、問題の緩和や回避に関する正式な指針を提供している。また、700種類を超えるプログラミングエラーに関するデータのほか、悪用可能な脆弱性に結び付く可能性のある設計上およびアーキテクチャ上のエラーも紹介している。

cwe.mitre.org/

SANSでは、3言語によるセキュアコーディングスキルの評価基準と各種認定試験が提供されている。この基準を元に、プログラマがセキュアコーディングの知識に関するギャップを認識したり、購入者がアウトソーシングしたプログラマが十分なプログラミングスキルを備えているかどうかを判断したりできる。プログラマの数が500人を超える組織では、最大100名のプログラマを対象としてセキュアコーディングスキルを経費負担なしで査定できる。

詳細の問い合わせ先: spa@sans.org

GSSPブループリントの詳細: www.sans-ssi.org/certification/

SAFECode - EMC、Juniper、Microsoft、Nokia、SAP、Symantecの各社が参加するSoftware Assurance Forum for Excellence in Codeでは、ソフトウェアの保証に関連した業界のベストプラクティスに関する記述と、セキュアソフトウェアの開発に応用できる実証済み手法の実装に向けた実践的なアドバイスを記載した次の優れた資料を公開している。

http://www.safecode.org/publications/SAFECode_BestPractices0208.pdf

http://www.safecode.org/publications/SAFECode_Dev_Practices1108.pdf

10社を超えるソフトウェア会社から、こうしたエラーを対象としてプログラムをテストする自動ツールが提供されている。SANSでは、これらのツールをはじめ、その他のセキュリティツールに関するユーザエクスペリエンスを取り上げたケーススタディを発表している。

www.sans.org/whatworks

ニューヨーク州では、企業がセキュリティの裏付けがあるソフトウェアを購入する際に役立つように調達基準の草案を策定した。

ニューヨーク州の調達基準草案は、www.sans.org/appseccontractで参照できる。

詳細情報の問い合わせ先:

SANS: Mason Brown,mbrown@sans.org

MITRE: Bob Martin,ramartin@mitre.org

MITRE: Steve Christey,coley@mitre.org

SANS Software Security Institute (SSI) コース

- Web Application Pen Testing Hands-On Immersion : Developer 538
- Web App Penetration Testing and Ethical Hacking : Security 542
- Intro to Web Application Security : Developer 319
- Web Application Security Essentials : Developer 422
- Secure Coding in Java/JEE: Developing Defensible Applications : Developer 541
- Secure Coding for PCI Compliance : Developer 536
- Defensible .NET : Developer 616
- Exploiting Regular Expressions to Process Text : Security 651
- AJAX and Web Services Security Overview : Security 426
- Java Quality Assurance, Security Testing and Auditing : Audit 428
- Secure Web Services for Managers : Management 431
- Security Policy & Awareness : Management 524
- Software Security Awareness : Developer 304

Japanese Translated Edition by NRI SecureTechnologies,Ltd.