Confidential

文書番号 XXX-SC-YYMMDD-XX

株式会社 Example 御中

ウェブアプリケーション脆弱性診断報告書

201X 年 MM 月 DD 日

本報告書は、2010 年 MM 月 DD 日から DD 日にかけて、株式会社 Example 様「example サイト」ウェブアプリケーションの脆弱性を診断した結果をご報告するものです。報告書の内容には、脆弱性に関する情報が含まれますので、報告書の取り扱いにはご注意ください。

合同会社シマンテック・ウェブサイトセキュリティ

目次

1 はじめに 1
1.1. 診断の目的 1
1.2. ウェブアプリケーション脆弱性診断の位置づけ 1
1.3. 本報告書の取り扱いについて 1
1.4. セキュリティ対策の運用について 2

2 診断内容 3
2.1. 診断環境 3
2.2. 診断項目 4
2.3. 診断対象 5

3 診断結果概要 7
3.1. 総合評価 7
3.2. 検出された脆弱性一覧 7
3.3. 総評 7
3.4. 想定被害 8

4 診断結果詳細 9
4.1. クライアント側での攻撃 9
4.2. コマンドの実行 13
4.3. 情報公開 17

5 注意事項 20
5.1. HTTP での重要情報送信について 20

6 お問合せ 21
6.1. お問合せ先 21
6.2. 再診断について 21
6.3. その他のサービスについて 21

付録 A 危険度の判定基準 22

付録 B JNSA 想定損害賠償金額算定式 23

付録 C 参考文献 25

1 はじめに

1.1. 診断の目的

インターネットを介した不正アクセスの多くが、ウェブサイトを狙った攻撃となった今日では、不正アクセス等の攻撃に耐えられる堅牢なウェブシステムが求められています。

堅牢なウェブシステムを構築するためには、システム構成要素の脆弱性を正しく認識し対処する必要があります。OS や、ウェブサーバおよびデータベースなどのミドルウェアの脆弱性情報は信頼のおける公的な機関から情報提供されますが、各々が構築するウェブアプリケーションの脆弱性はウェブシステムを運用している企業が自ら調べるほか脆弱性を正しく認識することはできません。

弊社のウェブアプリケーション脆弱性診断は、堅牢なウェブシステムを構築するためにウェブアプリケーションに潜在する脆弱性を弊社技術者が診断・発見し、脆弱性の種類、発見箇所および対策指針をご報告いたします。弊社のウェブアプリケーション脆弱性診断により、堅牢なウェブシステムの構築・運用に役立てていただくことを目的としています。

1.2. ウェブアプリケーション脆弱性診断の位置づけ

堅牢なウェブシステムを構築・運用するためには、OS やウェブサーバおよびデータベースなどのミドルウェアの脆弱性に対しても適切に対処する必要があります。また、Firewall に代表されるネットワークのセキュリティ対策も不可欠です。したがって、ウェブアプリケーションの脆弱性診断と診断結果による対策も非常に重要であるとともに、ウェブサイトの各構成要素に対しても脆弱性診断とその結果による対策を施す必要があります。

本報告書が対象としているのは、ウェブアプリケーションの脆弱性診断のみとなっており、OS やミドルウェアなどの脆弱性に関する情報は含まれていません。別途、これらについても専門機関を利用するなどして十分な対策を行なわれることを推奨します。

1.3. 本報告書の取り扱いについて

本報告書には、お客様のウェブアプリケーションに関するセキュリティ上の問題点が記載されています。この情報がひとたび悪意ある第三者に渡ってしまうと、セキュリティ上の問題点を狙った不正アクセス攻撃を受け情報漏えい等の事故が発生する可能性があります。したがって、本報告書のお取り扱いには十分に注意して頂けますようお願いいたします。

1.4. セキュリティ対策の運用について

本報告書は診断時点でのお客様のウェブアプリケーションのセキュリティ上の問題点を診断した結果が記載されています。ウェブアプリケーションへの攻撃は日々研究され進化し続けているため、時間経過とともにセキュリティ上の問題が増加することが考えられます。

堅牢なウェブシステムを継続的に運用するためには、定期的にウェブアプリケーションに潜んでいるセキュリティ上の問題点を正しく認識し対策を施すことをお勧めいたします。

2 診断内容

2.1. 診断環境

2.1.1. 診断日時

- 2010 年 MM 月 DD 日　10:00 ～ 18:00
- 2010 年 MM 月 DD 日　10:00 ～ 18:00
- 2010 年 MM 月 DD 日　10:00 ～ 18:00

2.1.2. 診断元 IP アドレス

- XXX.XXX.XX.XXX

2.2. 診断項目

主な診断項目を以下に列挙します。

区分	名称
認証	パスワードポリシー
	不適切な認証
	脆弱なパスワードリマインダ
承認	セッションの推測
	不適切な承認
	セッションの固定
クライアント側での攻撃	クロスサイトスクリプティング
	コンテンツの詐称
コマンドの実行	バッファオーバーフロー
	書式文字列攻撃
	LDAP インジェクション
	OS コマンドインジェクション
	SQL インジェクション
	SSI インジェクション
	XPath インジェクション
情報公開	ディレクトリインデクシング
	ソース記載による情報漏えい
	パストラバーサル
	推測可能なリソース位置
ロジックを狙った攻撃	機能の悪用
	リダイレクタ
	不適切なプロセスの検証

2.3. 診断対象

2.3.1. 対象サイト

- ※対象サイトのサービス名などが記載されます。

2.3.2. 対象 URL

- http://www.example.com/index.pl
- http://www.example.com/confirm.php
- http://www.example.com/thanks.asp
- http://www.example.com/entry.jsp

※診断対象の URL が記載されます

2.3.3. 対象 IP アドレス

- ※お客様の IP アドレスが記載されます。

2.3.4. ウェブサーバの情報

URL	http://www.example.com/
IP アドレス	xxx.xxx.xx.xxx
OS バージョン	Windows
ウェブサーバのバージョン	Apache
アプリケーションサーバのバージョン	ASP
データベース	Oracle
開発言語	Java

※お客様のウェブサーバの情報を記載します。

3 診断結果概要

3.1. 総合評価

D. 一部危険な個所があります

3.1.1. 総合評価について

総合評価では、診断結果詳細にて記載されている各脆弱性の危険度（Low、Medium、High）のレベルと危険度の個数を元に下記の表のような評価をしております。なお、危険度の詳細については、付録 A を参照してください。

評価	基準
A	脆弱性が検出されなかった
B	危険度 Low の脆弱性のみ検出
C	危険度 Medium の脆弱性を検出
D	危険度 High の脆弱性を検出
E	危険度 High の脆弱性を複数検出

3.2. 検出された脆弱性一覧

脆弱性区分	脆弱性名	危険度	個数
クライアント側での攻撃	クロスサイトスクリプティング	Medium	2
コマンドの実行	SQL インジェクション	High	1
情報公開	情報漏えい (エラーメッセージ)	Low	2

3.3. 総評

今回の診断の結果、お客様のサイトではユーザから送信されてくる各種の値に対する基本的なチェックが行われている様子が確認されましたが、いくつかのパラメータにおいてチェック漏れがあるため、

SQL インジェクションなどの非常に危険度の高い脆弱性が存在することが確認されました。これらの脆弱性を悪用された場合、データベースに格納されている個人情報の漏えいやデータの改ざんなどの被害が発生する可能性があります。

また、サイトの一部においてエラー発生時にその詳細な情報を表示してしまっている箇所があります。このようなところからシステムの動作に関する情報が漏えいしており、その他の脆弱性と組み合わせて悪用されることで、より大きな被害に繋がる可能性があります。

その他にも、やや危険度は低くなりますが、クロスサイトスクリプティングなどの問題も検出されています。これらについても本報告書の内容を参考に十分な対策を行っていただければと思います。

3.4. 想定被害

今回発見されたウェブアプリケーションの脆弱性を悪用された場合、以下のような個人情報が漏えいする可能性があります。

- 氏名
- 住所
- 電話番号
- 購入履歴など

このような個人情報が漏えいする最悪の事態が発生した場合の、1人あたりの想定賠償金額を算定すると 1 人あたりの損害賠償額は￥XX,000-となります。

最悪の事態が発生した場合、被害者への損害賠償金以外にも、直接被害として機会損失や逸失利益などが発生するため、想定被害額は想定損害賠償額よりも大きな金額となります。

なお、損害賠償額の計算は、NPO 日本ネットワークセキュリティ協会が公開している「情報セキュリティインシデントに関する調査報告」に記載されている個人情報漏えいにおける想定損害賠償額の算定式を利用して算定しています。詳細については、付録 B を参照してください。

※対象のサイトで取り扱っている情報の種類を記載して、被害総額を計算します。

4 診断結果詳細

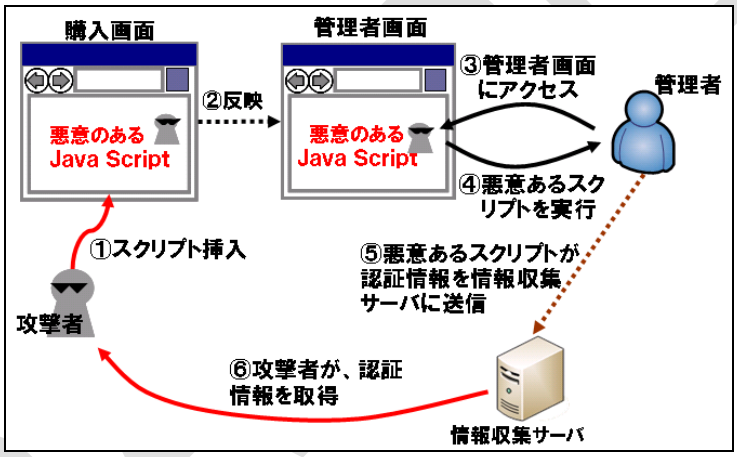
4.1. クライアント側での攻撃

4.1.1. クロスサイトスクリプティング

(1) 危険度

Medium

(2) 脆弱性概要



悪意あるスクリプトがユーザのブラウザ上で実行されてしまうことで、認証情報が悪意ある第三者に取得される可能性があります。また、ウェブページの内容を改ざんし、フィッシング詐欺に悪用される可能性があります。

(3) 発生箇所

No	URL	パラメータ名
1	http://www.example.com/login.php	param
2	http://www.example.com/confirm.php	param

(4) 脆弱性発生状況

発生箇所 No.1 について解説します。パラメータ param に対して以下のように不正な値を設定して送信します。

```
param=param1<script>alert(document.cookie)</script>
```

その結果、図 4-1 のようにスクリプトが実行されることが確認されました。

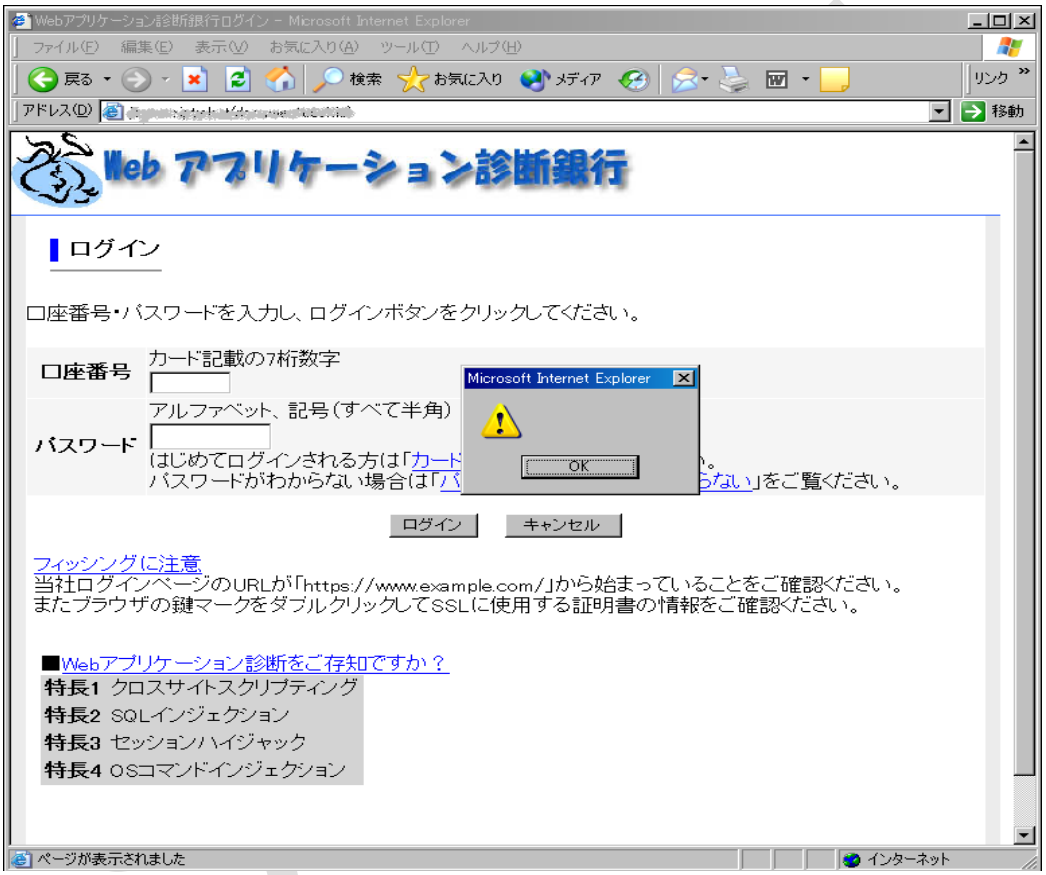


図 4-1

このようなクロスサイトスクリプティングの脆弱性を悪用することにより、Cookie の値を攻撃者のサーバに送信することでその値を盗みとることなどが可能となります。また、スクリプトにより画面の表示内容を別のものに見せかけることで、本来とは異なる別のサーバにデータを送信させられてしまう場合やフィッシングサイトとして悪用される場合があります。

(5) 想定される脅威

ユーザの Cookie の情報が攻撃者に取得され、他のユーザになりすましが行なわれるほか、サイトの内容を書き換えられ、さらにその結果としてフィッシングサイトとして悪用される可能性があります。

(6) 想定される被害

① 脆弱性悪用の実現度

クロスサイトスクリプティングは貴社のサイトに誘導するステップが必要となるため、直ちにこの脆弱性を悪用される可能性はそれほど高くはありませんが、攻撃者はスクリプトを埋め込んだサイトへ顧客を誘導する際に、知名度や社会的信頼度の高いサイトを悪用します。

したがって、貴社のように知名度や社会的信頼度の高いサイトは攻撃者に狙われ易いと考えられます。また、クロスサイトスクリプティングという名称が世間的に広く知られている脆弱性であるため脆弱性の存在が報道された場合の風評被害が大きいという特徴があります。

② 脆弱性悪用により想定される被害

- なりすましによる他のユーザ権限の不正利用
- なりすましによる個人情報の漏えい
- 情報漏えい等の事故を原因とするサイト停止による機会損失
- 情報漏えい等に対する損害賠償や見舞金
- 情報漏えい等の事故による企業の信用失墜

(7) 対策

ユーザから入力された値を画面上に出力するような処理の場合、出力する値の中にスクリプトなどの不正な値が含まれていないかを確認します。不正な値が含まれていた場合には、無害な文字に置き換える処理（サニタイジング）を行なってください。

また、より有効な対策としては、ユーザから値を渡された時点で値をチェックすることで、入力可能となる文字を制限する（郵便番号であれば数字とハイフンのみを受け付け、それ以外の値が混入したらエラー処理を行なう）などの処理を行なうことがあげられます。このような手法によって不正な文字列の混入を防止することで、より安全なサイトが構築可能となります。

(8) 参考情報

① 無害化処理について

一般的には HTML タグとして認識される文字を置換することになります。HTML タグとして認識される文字とは以下のようなものがあります。

- & ⇒ &
- < ⇒ <

- > ⇒ >
- " ⇒ "
- ' ⇒ &#39

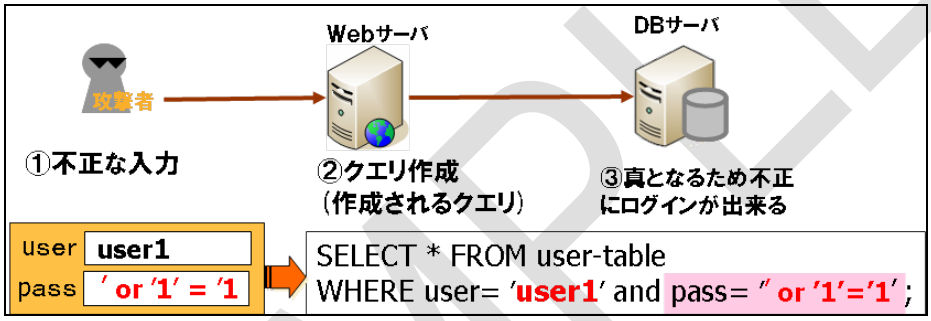
4.2. コマンドの実行

4.2.1. SQL インジェクション

(1) 危険度

High

(2) 脆弱性概要



データベースの SQL クエリのパラメータとなる入力に、不正な文字列を挿入（インジェクション）して、不正な SQL クエリを実行させる攻撃です。上図は不正な文字列により認証を回避する際の SQL インジェクション攻撃の例です。

(3) 発生箇所

No	URL	パラメータ名
1	http://www.example.com/confirm.php	param

(4) 脆弱性発生状況

発生箇所 No.1 を例に解説します。パラメータ param に対し以下のような不正な値を設定して送信します。

```
param=param1¥
```


その結果、図 4-2 のような画面となりました。

図 4-2

上記画面に表示されているメッセージより、SQL 文を実行しようとしている際にエラーが発生している様子が確認されます。

そこでさらに、パラメータ param に対して、以下のような不正な値を設定して送信します。

param =param1'or'1'='1'--

その結果、図 4-3 のような画面となりました。

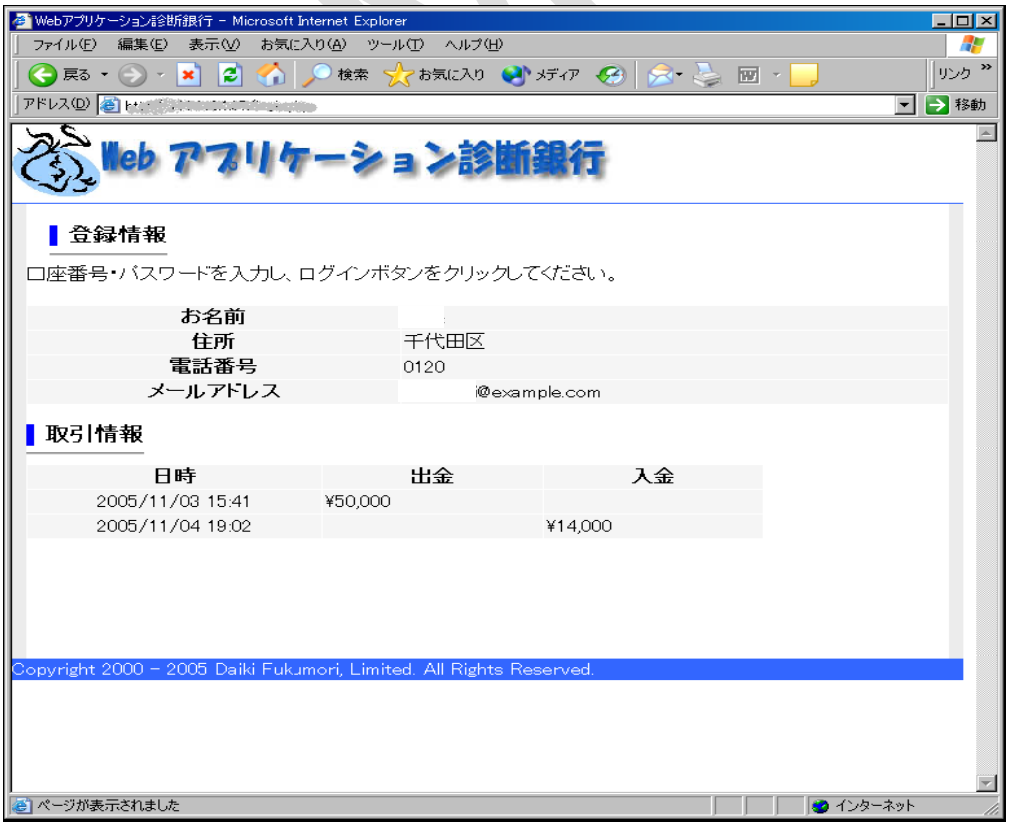


図 4-3

これより、パラメータ param に設定した SQL の条件が常に真となり得る値の入力により、ログイン処理を迂回しログイン可能なことが確認されます。

本脆弱性の存在により、攻撃者によってさらに多くの SQL 文を実行されることで重要な情報の漏えいなどが発生する可能性があります。

※実際の診断においては、お客様のデータベースの内容を大幅に変更してしまうような SQL 文を送信しないように十分注意した方法で診断を行います。

(5) 想定される脅威

データベースに保存されている情報の漏えい、改ざん、破壊等の恐れがあります。

(6) 想定される被害

① 脆弱性悪用の実現度

攻撃方法にある程度の知識を必要とする点、攻撃成功のためには多数のリクエストが必要となる点を考慮すると、実現度はやや低くなります。また、攻撃者にとってより有効な情報を得るためには、非常に多くの SQL 文を実行することが必要となってきます。このため、侵入検知システムなどを導入している場合は容易に検知可能であると考えられます。

② 脆弱性悪用により想定される被害

- データベースに保存されているユーザ情報等の漏えい
- データベースに保存されているユーザ情報等の改ざん、破壊
- 情報漏えい等の事故を原因とするサイト停止による機会損失
- 情報漏えい等に対する損害賠償や見舞金
- 情報漏えい等の事故による企業の信用失墜

(7) 対策

SQL 文を文字列組み立てにより生成するのではなく、プレースホルダ、バインドメカニズムを用いてあらかじめ実行する SQL 文を決定しておくことを推奨します。

そのような対策が取れず、ユーザから入力された値を元にしてデータベースへの問い合わせを行なう場合、実際にデータベースへの問い合わせを行なう段階で、入力された値の中に不正な文字列が含

まれていないかをチェックし、その結果としてエラー処理や無害な文字列に変換するなどの処理を行なってください。

また、郵便番号などのようにパラメータに与える文字種が限定されるような場合では、それ以外の文字種を受け付けないようにしておくことでより安全なものとなります。

(8) 参考情報

① プレースホルダ、バインドメカニズム

発行する SQL 文のうちで値を入力できる部分をあらかじめ決めておくことで、意図しない SQL 文が発行されることを防ぐことが出来ます。

PHP の場合は以下のようになります。

```
// $con という名前の DB オブジェクトを取得している場合
$sql = ”select username from usertable where userid = ? and password = ?”;
$parameter = array(”xxx”,”secret”);
$result = $con->getAll($sql, $parameter);
```

② ホワイトリスト方式

不正な値を排除する方法として「不正な値がふくまれていないか」を確認するよりも「正当な値か」を確認したほうがチェック漏れを減らすことが出来ます。例えば電話番号入力欄には入力された値が数字かを確認し、性別選択欄では男女のどちらかが入力されているかを確認します。「不正な値がふくまれていないか」を確認する方法はブラックリスト方式と呼ばれており、「正当な値か」を確認する方法はホワイトリスト方式と呼ばれています。

入力値のチェックを行なう場合には、「ホワイトリスト方式」を使用するように心がけてください。

③ 必要最小限の権限付与

万一攻撃に成功されてしまうことも想定してデータベースの権限分離を行なっておくことで被害を最小限に食い止めることができます。例えば、データベースのシステムテーブル（ユーザ ID やパスワード等を管理しているテーブル）には、一部のユーザだけがアクセス権を有していれば十分です。各データベース、テーブルに対して必要最小限のアクセス権が与えられていることを確認してください。

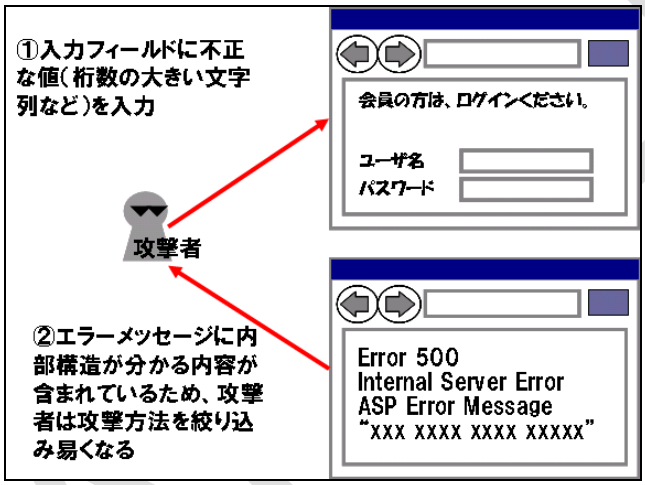
4.3. 情報公開

4.3.1. 情報漏えい（エラーメッセージ）

(1) 危険度

Low

(2) 脆弱性概要



不正な値(記号や桁数の大きい文字列など)を入力した際に、サーバ内部のエラー情報が表示されています。プログラム内部エラーが発生しているので、プログラムの異常終了やデータの不整合等の予期しない動作が発生している可能性があります。また、エラーメッセージによってはサーバ内部の情報を漏えいしてしまう恐れがあります。

(3) 発生箇所

No	URL	パラメータ名
1	http://www.example.com/login.php	param
2	http://www.example.com/confirm.php	param

(4) 脆弱性発生状況

発生箇所にて指摘している各パラメータに対して、桁数の大きな文字列や想定外と考えられる文字列(数字の入力が想定されているところに文字列を入力するなど)を入力することで、エラーが発生しています。

発生箇所 No.2 を例に解説します。パラメータ param に対して、以下のように不正な値を設定して送信します。

```
param=param1'
```

その結果、図 4-4 のようにエラー画面が表示されます。

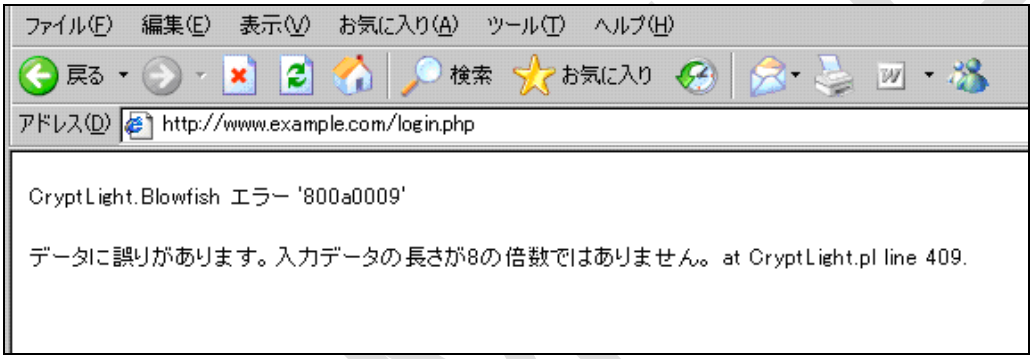


図 4-4

このようなエラーメッセージより、暗号ライブラリとして CryptLight を使用していることや、暗号アルゴリズムとして Blowfish を使用している様子がわかります。

これらの情報をもとにして暗号解読など、さらなる攻撃に発展する可能性があります。

(5) 想定される脅威

サーバの情報が漏えいする可能性があります。

(6) 想定される被害

① 脆弱性悪用の実現度

脆弱性発生箇所のパラメータに桁数の非常に大きな文字列を入力したり、数字が入力されることを仮定しているパラメータに対して、数字以外の値を入力したりするなど、本来パラメータで入力されると考えられている値以外の文字を入力することなどで、容易に実現可能です。

本問題の存在によってただちに何か問題が発生するというわけではありませんが、このようなプログラムのエラーが発生することは、攻撃者にとっては攻撃の動機をあげることに繋がり、また本サイトを利用しているユーザにとっては、信頼度を下げる結果となる可能性があります。

② 脆弱性悪用により想定される被害

- プログラムの異常終了、データ不整合、サーバ内部の情報漏えい
- 上記内容を原因とするサイト停止による機会損失

(7) 対策

ユーザから値を渡された時点で、入力文字中に不正な値が混入していないかをチェックし、混入していた場合は専用のエラー処理を行うことで、内部サーバエラーが発生しないようにしてください。例として、入力可能となる文字を制限する(郵便番号であれば数字とハイフンのみを受け付け、それ以外の値が混入したらエラー処理を行う)などの処理を行うことがあげられます。

また、万一内部サーバエラーが発生してしまった場合にも、不必要な情報を攻撃者に与えないことが重要です。そのためには詳しいエラー内容が特定できるようなエラーメッセージの表示を抑制し、代わりのエラーページを作成して表示するか、何も表示しないようにします。

(8) 参考情報

PHP では php.ini を以下のように編集することで、たとえ実行時にエラーが発生してしまった場合でも、そのエラーメッセージが画面へ表示されることを抑制することが可能です。

```
display_errors = Off
```


5 注意事項

注意事項では、脆弱性ではありませんがお客様のサイトのセキュリティをなおいっそう高いものとなりますよう推奨項目として記載しております。

5.1. HTTP での重要情報送信について

アカウント情報や個人情報などの重要情報をサーバに送信する際に HTTP 通信を利用すると、通信データは暗号化されず送信されます。このとき、攻撃者に通信路が盗聴されてしまうと重要情報が攻撃者に盗み取られてしまう可能性があります。

したがって、重要情報を通信する際には通信路の盗聴による情報漏えいを防ぐために HTTPS 通信を利用することを推奨します。

また、重要情報送信箇所に HTTPS を利用する際には以下の点に注意をして実装してください。

① Cookie の Secure 属性

HTTPS のページでセッション管理を行なう場合には、Cookie の Secure 属性を有効にしてください。Cookie の Secure 属性を有効にすることで、ブラウザは暗号化されていない通信路には Secure 属性が有効となっている Cookie を送信しないようになります。

これにより重要情報であるセッション ID の通信路盗聴による漏えいを防ぐことが可能となります。

② セッション ID

HTTP のウェブページと、HTTPS のウェブページにまたがって Cookie によるセッション管理を行なう場合、前者には「Secure 属性なし」の Cookie を使用し、後者には「Secure 属性あり」の Cookie を使用するようにして下さい。

③ HTTP 通信の制限

HTTPS を利用してアクセスする箇所では HTTP では通信できないように制限を設けて下さい。HTTP でも通信が可能であると、攻撃者により HTTP で通信するように誘導された場合には暗号化されていないデータを盗聴される可能性があります。

6 お問合せ

6.1. お問合せ先

本報告書の内容に関するお問合せは、下記メールアドレスまでご連絡ください。お問合せの対応は、報告書提出から 1 ヶ月以内に限らせていただきます。なお、電話や FAX でのお問合せは受付けておりませんので、ご了承ください。

xxx@symantec.com

6.2. 再診断について

弊社が指摘した脆弱性発生箇所をお客様にて修正された後に無料で再診断をご利用可能です。なお、再診断実施に当たっては以下の条件がございますので、ご注意ください。

- 再診断実施回数　1 回
- 期間　報告書提出から 1 ヶ月以内
- 診断箇所　弊社より指摘した脆弱性発生箇所(危険度 Medium 以上のもののみ)
- 成果物　報告書（報告会は実施しません）

6.3. その他のサービスについて

弊社では、ウェブアプリケーション脆弱性診断をはじめとする、「診断・防御・教育」の 3 つのアプローチでサービスを提供しております。

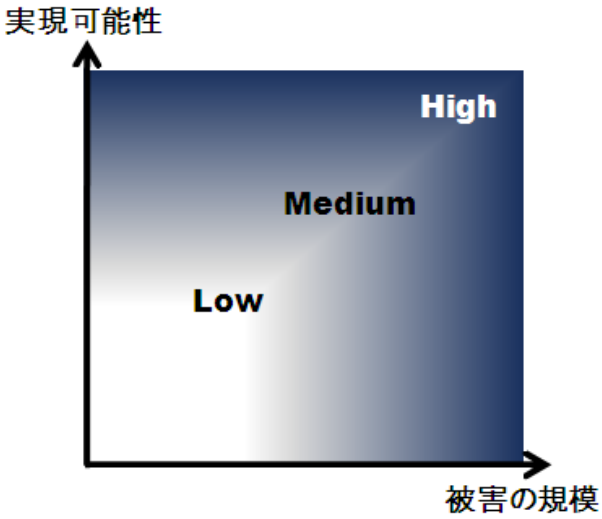
- プラットフォーム診断
  お客様のサーバやネットワーク機器に潜むセキュリティ上の問題点を診断します。

- ウェブアプリケーションファイアウォールサービス
  お客様サイトを攻撃する通信を弊社のウェブアプリケーションファイアウォールが防御します。

- セキュリティ教育
  サイト管理者、開発者の視点に立ったセキュアなウェブサイト設計のための教育サービスを提供します。

付録 A 危険度の判定基準

脆弱性の危険度は、以下の図のように攻撃の実現可能性と被害の規模を軸として High, Medium, Low の 3 段階の値に分けています。



・攻撃の実現可能性（縦軸）　：環境要因によるものや人的要因を考慮します。
・被害の規模（横軸）　：被害者数や被害の種類などを考慮します。

- High　攻撃の実現可能性が高く、被害の規模も大きい
- Medium　攻撃の実現可能性が低いかまたは被害の規模が小さい
- Low　攻撃の実現可能性が低く被害の規模も小さい

なお、実現度が低くても被害の規模が過度に大きいようであれば High と判定し、また実現度が高くても、被害の規模が小さいようであれば Low と判定します。

付録 B JNSA 想定損害賠償金額算定式

⑴ 算定式

想定損害賠償額 = 漏えい個人情報価値

× 情報漏えい元組織の社会的責任度

× 事後対応評価

⑵ 漏えい個人情報の価値

漏えい情報価値 ＝ 基礎情報価値 × 機微情報度 × 本人特定容易度

(ア) 基礎情報価値

基礎情報価値は一律、「500」ポイントで設定します。

(イ) 機微情報度

機微情報度は、精神的苦痛レベルと経済的損害レベルを考慮して算出します。算出の基礎となる漏えいした情報の精神的苦痛レベルと経済的損害レベルは下表の上部(x、y)から、下記の式に代入して求めます。レベルの異なる複数の漏えい情報がある場合は、全情報のうちでもっとも大きなxの値と、yの値を採用します。

算出式:機微情報度 ＝ $(10^{x-1} + 5^{y-1})$

経済的損失レベル	精神的苦痛レベル 1	精神的苦痛レベル 2	精神的苦痛レベル 3
3	口座番号/暗証番号, クレジットカード番号, カード有効期限, 銀行アカウント/パスワード	遺言書	前科前歴, 犯罪歴, 与信ブラックリスト
2	パスポート情報, 購入記録, ISPのアカウント/パスワード	年収, 年収区分, 資産, 建物, 土地, 残高, 借金, 所得, 借入れ記録	
1	氏名, 住所, 生年月日, 性別, 金融機関名, 住民票コード, メールアドレス, 健康保険証番号, 年金証書番号, 免許証番号, 社員番号, 会員番号, 電話番号, ハンドル名, 健康保険証情報, 年金証書情報, 介護保険証情報, 会社名, 学校名, 役職, 職業, 職種, 身長, 体重, 血液型, 身体特性, 写真(肖像), 音声, 声紋, 体力診断	健康診断, 心理テスト, 性別判断, 妊娠経験, 手術歴, 看護記録, 検査記録, 身体障害者手帳, DNA, 病歴, 治療法, 指紋, レセプト, スリーサイズ, 人種, 地方なまり, 国籍, 趣味, 特技, 嗜好, 民族, 日記, 賞罰, 職歴, 学歴, 成績, 試験得点, メール内容, 位置情報	加盟政党, 政治的見解, 加盟労働組合, 信条, 思想, 宗教, 信仰, 本籍, 病状, カルテ, 痴呆症, 身体障害, 知的障害, 精神的障害, 保有感染症, 性癖, 性生活

（ウ） 本人特定容易度

本人特定容易度は、漏えいした個人情報から被害者本人の特定しやすさをあらわします。

判定基準	本人特定容易度
個人を簡単に特定可能。 「氏名」「住所」が含まれること。	6
コストをかければ個人が特定できる。 「氏名」または「住所＋電話番号」が含まれること。	3
特定困難。 上記以外。	1

⑶ 情報漏えい元組織の社会的責任度

社会的責任度は、「一般より高い」と「一般的」の 2 つから選択します。

判定基準		社会的責任度
一般より高い	個人情報の適正な取り扱いを確保するべき分野の業種（医療、金融、情報通信など）および公的機関、知名度の高い大企業	2
一般的	その他一般的な企業および団体、組織	1

⑷ 事後対応評価度

事後の対応の評価値を求めます。

判定基準	事後対応評価度
適切な対応	1
不適切な対応	2
不明、その他	1

付録 C 参考文献

1.「安心・安全な情報経済社会の実現のための行動計画」の発表について

http://www.meti.go.jp/press/20060302002/20060302002.html

2.セキュアプログラミング講座

http://www.ipa.go.jp/security/awareness/vendor/programming/

3.新版 セキュアプログラミング講座

http://www.ipa.go.jp/security/awareness/vendor/programmingv2/

4.情報処理推進機構 安全なウェブサイトの作り方

http://www.ipa.go.jp/security/vuln/documents/ウェブ site_security.pdf

5.情報処理推進機構 安全な SQL の呼び出し方

http://www.ipa.go.jp/security/vuln/documents/ウェブ site_security_sql.pdf

6.NPO 日本ネットワークセキュリティ協会「2009 年 情報セキュリティインシデントに関する調査報告書」

http://www.jnsa.org/result/incident/2009.html

