

# 遊び方 1

## Elevation of Privilege の遊び方

カードを配る前に、脅威モデリングをしたいシステムの図を描きます。

カードを3～6人のプレイヤーに配ります。「Tamperingの3」のカードを持つプレイヤーは、第1ラウンドのスタートプレイヤーとしてこのカードを場に出します。ゲームは時計回りに進みます。各プレイヤーは、スタートプレイヤーが場に出したカードと同じ記号のカードを持っている場合は、そのカードを場に出します。同じ記号のカードを持っていない場合は、別の記号のカードを場に出します。各プレイヤーが1枚ずつカードを場に出したらラウンド終了です。

カードを場に出す際、プレイヤーはカードを読み上げ、対応する脅威を発表し、これを記録します。対応する脅威が思いつかなかった場合はその旨を発表します。いずれの場合も次のプレイヤーに番が移ります。

各ラウンドは、Elevation of Privilege (EoP) カードを誰も出さなかった場合、最初に場に出されたカードと同じ記号を持つカードのうち、最も大きいカードを場に出したプレイヤーの勝利となります。EoPカードが出された場合は、最も大きいEoPカードを場に出したプレイヤーが勝利します。なお、カードの大きさは、A > K > Q > J > 10…の順となり、2が最も小さいカードです。

ラウンドの勝者は、次のラウンドのスタートプレイヤーとして好きなカードを場に出します。新たなラウンドを始める前に、脅威について検討する時間を数分取りましょう。

### 得点：

場に出すカードに対応する脅威を発表できたら1点、ラウンドに勝利したら+1点。

## Elevation of Privilege の遊び方

読み上げる脅威は明確に表現され、テスト可能であり、対処可能でなければなりません。脅威の内容で揉めた場合は次の質問を自問してみてください「我々はこの脅威を、対応可能なバグや機能要求、設計変更とみなすだろうか？」。もし答えが「はい」であれば、それは本当の脅威です。

(答えが「いいえ」であった場合、これが本当の脅威ではないという意味ではありません。この質問は単に、対応可能な脅威に議論を集中させるための方法でしかありません)

「〇〇すれば攻撃できる」という質問は「〇〇すれば攻撃できる。これに対応するには△△と読み替えましょう。「あなたのコードにおいて〇〇」という質問は「我々が共同で作成しているコードにおいて〇〇。これに対応するには△△と読み替えましょう。

デッキには「EoP」と「オープン脅威」という特別なカードが入っています。EoPカードは、たとえ小さいカードであっても、場に最初に出されたカードと同じ記号を持つカードより強いカードと見なされます。各記号のエースはオープン脅威カードです。カードを場に出す際、プレイヤーは他のカードに記載されていない新たな脅威を発表しなければなりません。

すべてのカードが場に出されたらゲーム終了です。最も得点が高いプレイヤーがゲームの勝者となります。

楽しくプレイすることを忘れずに！

# 遊び方 2

## Elevation of Privilege 別の遊び方

### ルールのバリエーション：

- ・プレイヤーが対応する脅威を思いつけなかったカードに対して、他のプレイヤーが代わりに脅威を発表できるようにする。
- ・得点を2倍にした上で、代わりに脅威を発表することができたプレイヤーに対して1ポイントを与える。
- ・さらに、他のプレイヤーがその脅威に対して「リフ（重ねあわせ）」できるようにする。新たな脅威の重ね合わせに成功する度に、成功したプレイヤーに対して1ポイントを与える。
- ・「リフ」ができる時間は、最長でも60秒以内とする。
- ・システム図中の、脅威が発生した箇所に印をつける。

STRIDE脅威参照カードには、他のカードでカバーされている脅威一覧がまとめられています。オープン脅威カード（エース）を使う際の参考情報としてご利用ください。

## 内容：

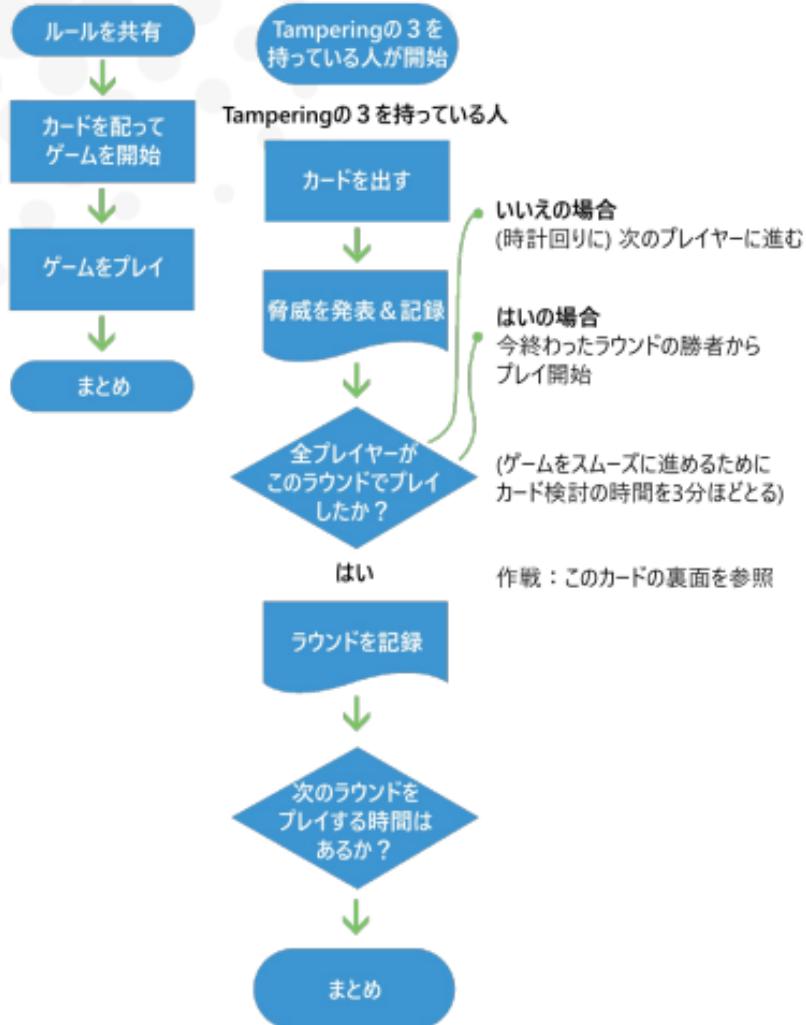
- ・「遊び方」カード 2枚
- ・「攻略図」カード 1枚
- ・STRIDE脅威カード 記号6種
  - 1. Spoofing (なりすまし): 2-K, Ace
  - 2. Tampering (改ざん): 3-K, Ace
  - 3. Repudiation (否認): 2-K, Ace
  - 4. Information Disclosure (情報漏えい): 2-K, Ace
  - 5. Denial of Service (サービス拒否): 2-K, Ace
  - 6. Elevation of Privilege (権限昇格): 5-K, Ace
- ・STRIDE脅威参照カード 6枚
- ・「脅威モデルについて」カード 1枚

Portions © 2010 Microsoft Corporation. This game was invented by Adam Shostack at Microsoft. The original EoP deck was produced by Microsoft.

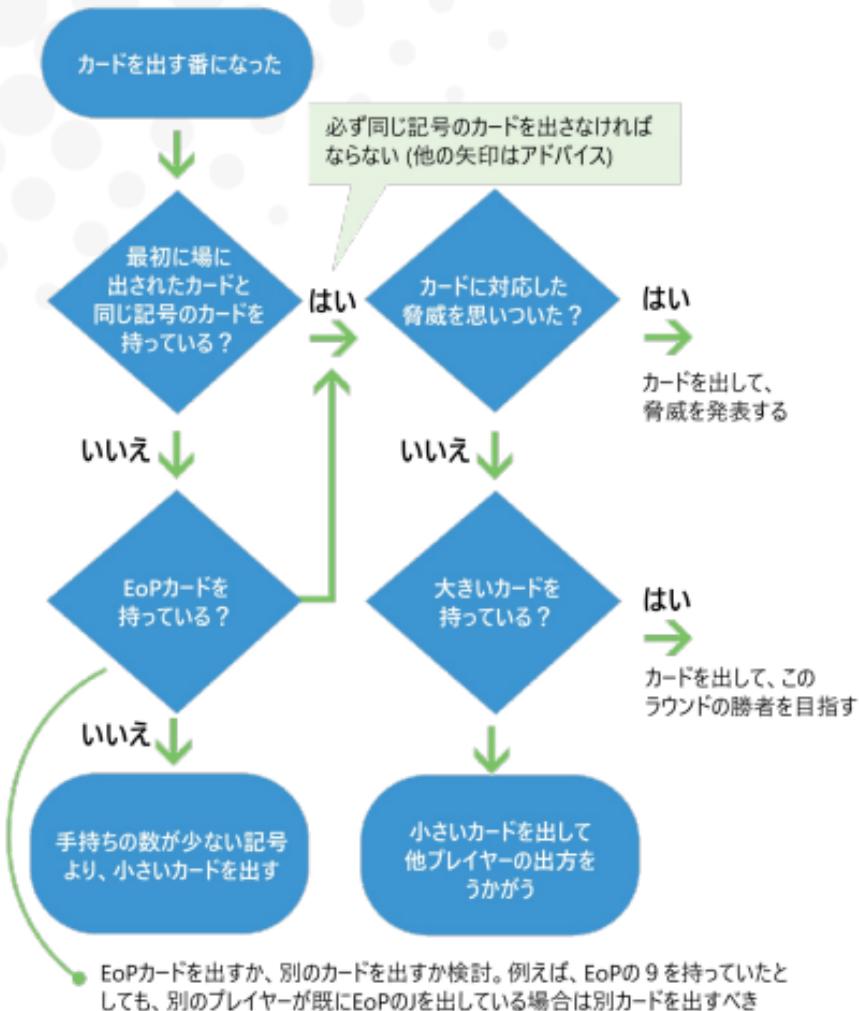
The original version of this edition © 2018 Agile Stationery Limited.  
<https://github.com/adamshostack/eop>

この日本語化版は、Kii株式会社の有志により、無料配布を前提に作成されたものです。  
この版では「いらすとや」のフリー素材を使用しています。使用されている素材の利用規約については <https://www.irasutoya.com/p/terms.html> をご確認ください。

# 大まかな流れ 遊び方



## 作戦の一例



# elevation of privilege

## 日本語バージョン



# 2

# Spoofing

攻撃者は、サーバが通常使用するランダムポートやソケットをスクワッティング（占拠）できる



# elevation of privilege

## 日本語バージョン



# 3

# Spoofing

攻撃者は、認証情報に対する  
総当たり攻撃をオンライン/オフラインで  
仕掛けることができる。この攻撃を  
遅らせるための仕組みは存在しない



# elevation of privilege

## 日本語バージョン



# 4

# Spoofing

上位レベルで認証が完了していると誤った期待をしているため、攻撃者が匿名で接続可能な状態になっている

上で認証しているはずだからヨシ！



# elevation of privilege

## 日本語バージョン



## 5

# Spoofing

サーバ識別方法が多すぎるため、  
攻撃者がクライアントを混乱させ  
ることができる



# elevation of privilege

## 日本語バージョン



## 6

# Spoofing

クライアントに識別子が保存されておらず、再接続時に識別子の一貫性（鍵の永続性）がチェックされていないため、攻撃者がサーバになりすますことができる



# elevation of privilege

## 日本語バージョン



# 7

# Spoofing

攻撃者は、認証されていない/  
暗号化されていないリンクを介して  
サーバまたはピアに接続できる



# elevation of privilege

## 日本語バージョン



## 8

# Spoofing

攻撃者は、サーバに保存されている認証情報を盗み、これを再利用できる（例：鍵が誰でも読めるファイルとして保存されている）



# elevation of privilege

## 日本語バージョン



9

# Spoofing

パスワードを入手した攻撃者が、これを再利用できる（より強固な認証システムの導入を検討すべきである）

パスワードではあんかった…



# elevation of privilege

## 日本語バージョン



10

# Spoofing

攻撃者は、より弱い認証方式を選択できる。もしくは完全に認証を回避できる。



10

# elevation of privilege

## 日本語バージョン



J

# Spoofing

攻撃者は、クライアントに保存されている認証情報を盗み、再利用できる

ヘー、パスワードは  
「！？ # A」なんだ！



# elevation of privilege

## 日本語バージョン



# Q

# Spoofing

攻撃者は、認証情報の更新・  
復元フローを悪用できる  
(例：古いパスワードの提示なく  
アカウントが復元できてしまう)



# elevation of privilege

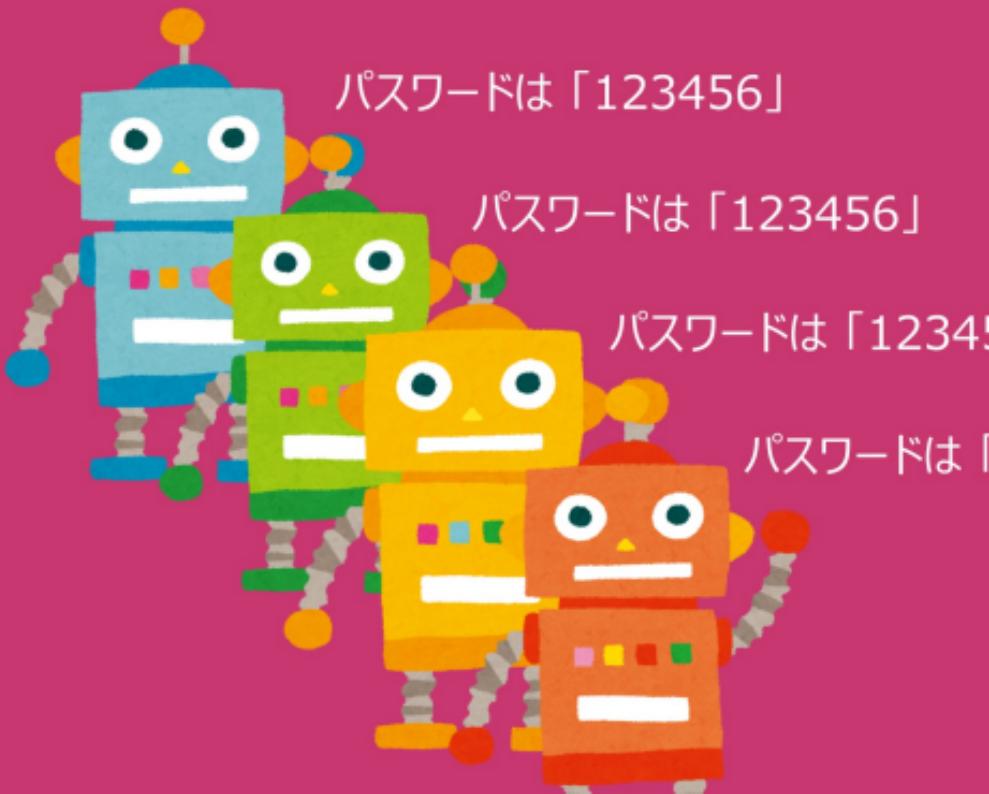
## 日本語バージョン



# K

# Spoofing

デフォルト管理パスワードが設定された  
状態でシステムをリリースしており、  
このパスワードを強制変更させていない



パスワードは「123456」

パスワードは「123456」

パスワードは「123456」

パスワードは「123456」

# elevation of privilege

## 日本語バージョン



A

# Spoofing

新たな「なりすまし」攻撃を考案した



A

# elevation of privilege

## 日本語バージョン



3

# Tampering

攻撃者は、標準的な暗号の代わりに採用されたオレオレ鍵交換プロトコルや完全性チェック機構の弱点を突くことができる

オレのシステムは世界一！



# elevation of privilege

## 日本語バージョン



# 4

# Tampering

コードにおけるアクセス制御の決定が、  
セキュリティカーネルではなくあらゆる  
場所で行われている

ワシが白と言うたら白じゃ



# elevation of privilege

## 日本語バージョン



## 5

# Tampering

タイムスタンプやシーケンス番号が存在しないため、攻撃者は検出されることなくリプレイ攻撃を仕掛けられる



# elevation of privilege

## 日本語バージョン



# 6

# Tampering

攻撃者は、コードが依存する  
データストアに書き込む



# elevation of privilege

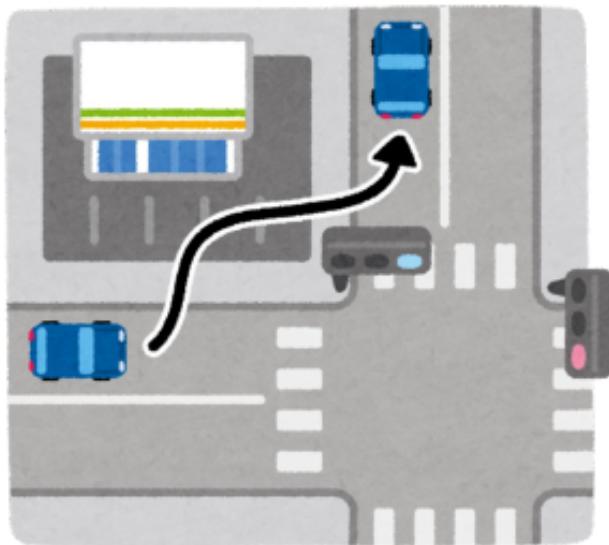
## 日本語バージョン



## 7

# Tampering

アクセスパーミッション確認前に  
名前を正規化していないため、  
攻撃者はパーミッションを迂回できる



# elevation of privilege

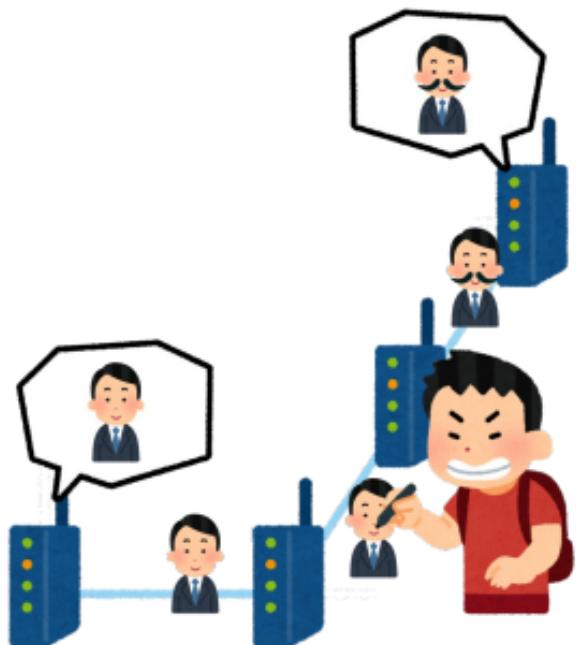
## 日本語バージョン



## 8

# Tampering

ネットワーク上を流れるデータの完全性が担保されていないため、攻撃者はデータを改ざんできる



# elevation of privilege

## 日本語バージョン



9

# Tampering

攻撃者は、ステート情報を提供  
または制御できる



# elevation of privilege

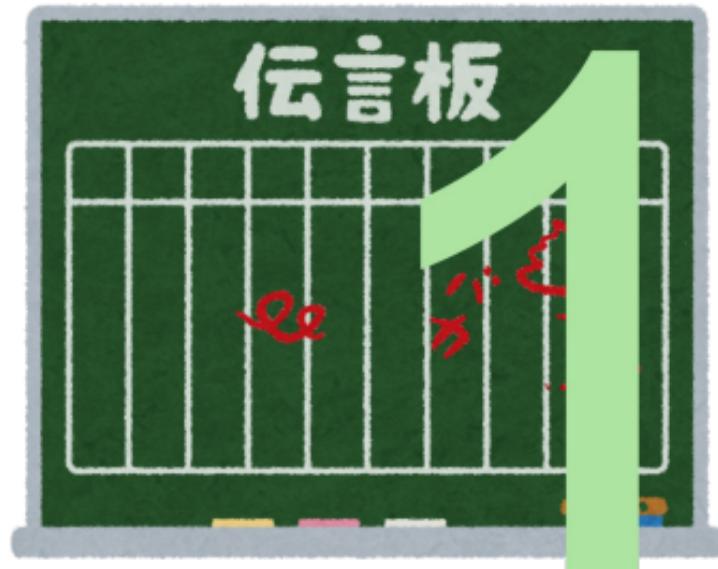
## 日本語バージョン



10

# Tampering

データストアのパーティションが脆弱/オープン/オープンに等しい状態（例：Facebookアカウントを持つ全員）であるため、攻撃者はデータストア内の情報を変更できる



# elevation of privilege

## 日本語バージョン



J

# Tampering

パーミッションが全ユーザーに付与されているか、そもそもACLが存在しないため、攻撃者は一部リソースに書き込みできる



# elevation of privilege

## 日本語バージョン



# Q

# Tampering

攻撃者は、バリデーション後のパラメータを、信頼境界を越えて変更できる（例えば、HTMLの非表示フィールド内の重要なパラメータや、重要なメモリーへのポインタ渡しなど）



# elevation of privilege

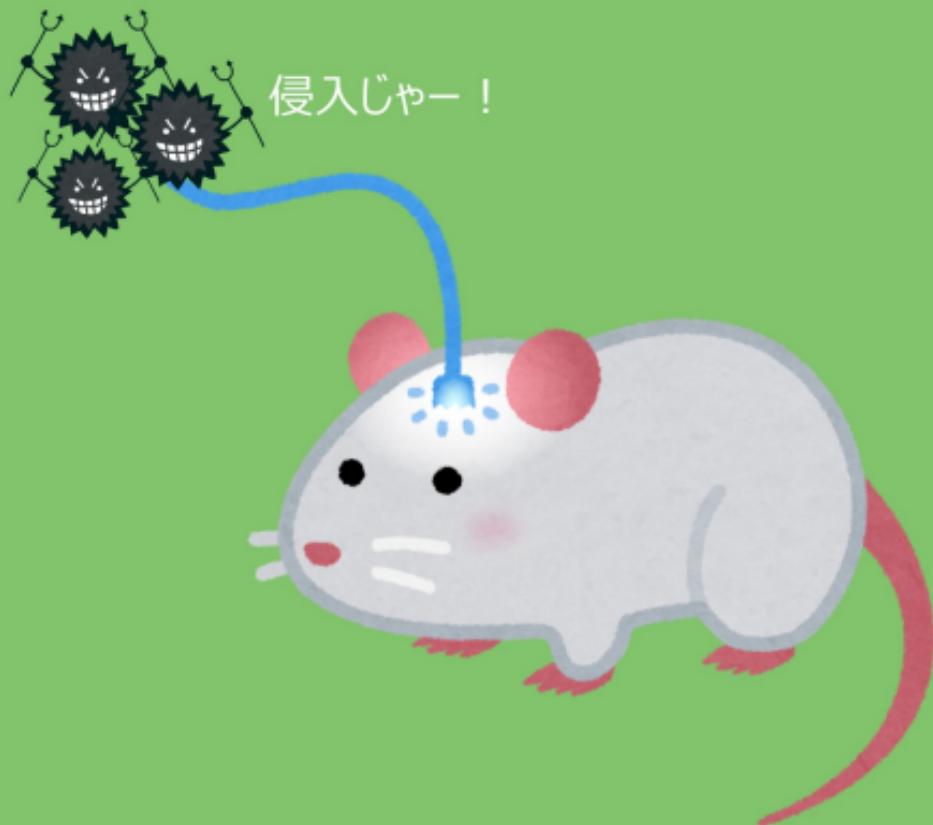
## 日本語バージョン



# K

# Tampering

攻撃者は、拡張ポイントを介して  
プロセス内に任意のコードをロード  
できる



# elevation of privilege

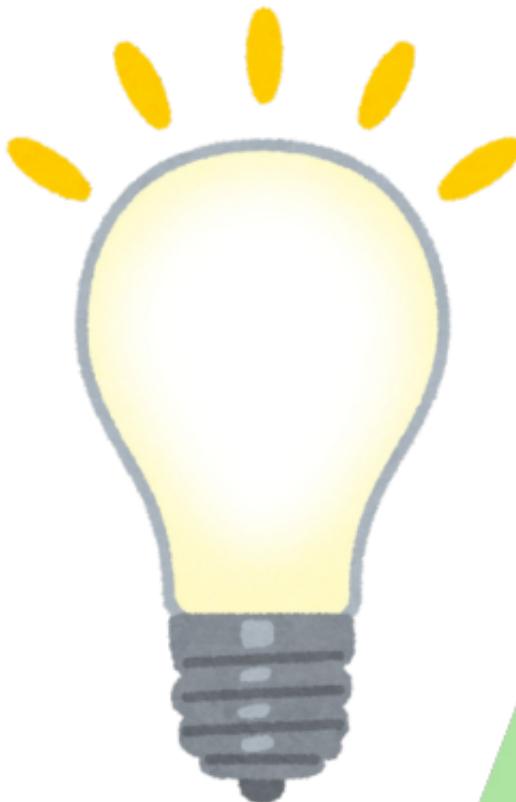
## 日本語バージョン



A

# Tampering

新たな「改ざん」攻撃を考案した



A

# elevation of privilege

## 日本語バージョン



2

# Repudiation

攻撃者はログに任意のデータを渡してログの読み手を搅乱できる。ログ書き込み時に実施されるバリデーションについて書かれた文書は存在しない。



# elevation of privilege

## 日本語バージョン



3

# Repudiation

低い権限しかもたない攻撃者が、  
ログ内の興味深いセキュリティ情報を  
読める



# elevation of privilege

## 日本語バージョン



## 4

# Repudiation

実装されたデジタル署名システムが脆弱であったり、署名を使用すべきところでMACを使用しているなどが原因で、攻撃者はファイルやメッセージを改ざんできる。



# elevation of privilege

## 日本語バージョン

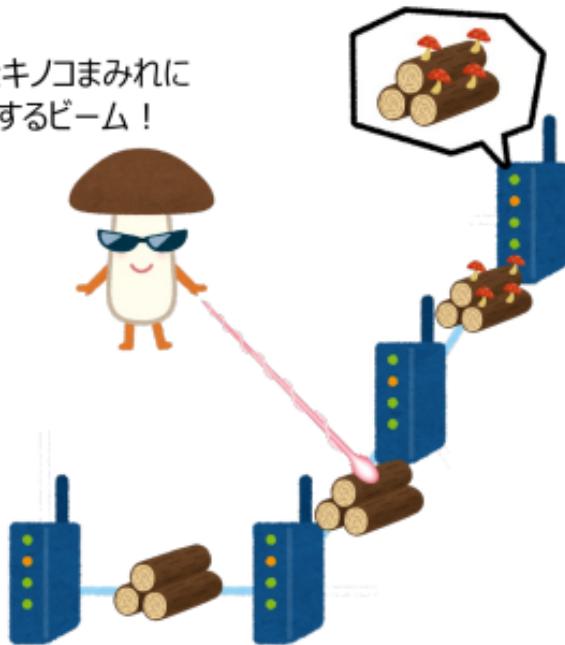


## 5

# Repudiation

完全性担保の仕組みが弱いため、  
攻撃者はネットワークを流れるログ  
メッセージを改ざんできる

ログをキノコまみれに  
改竄するビーム！



# elevation of privilege

## 日本語バージョン



# 6

# Repudiation

攻撃者は、タイムスタンプのないログエントリを作成できる（もしくは、ログエントリにタイムスタンプがない）



タイムスタンプが…  
ない……



# 6

# elevation of privilege

## 日本語バージョン



## 7

# Repudiation

攻撃者は、ログをローテートすることによりログを消滅させることができる



# elevation of privilege

## 日本語バージョン



8

# Repudiation

攻撃者はログを消滅させたり、  
セキュリティ情報を混乱させる  
ことができる



# elevation of privilege

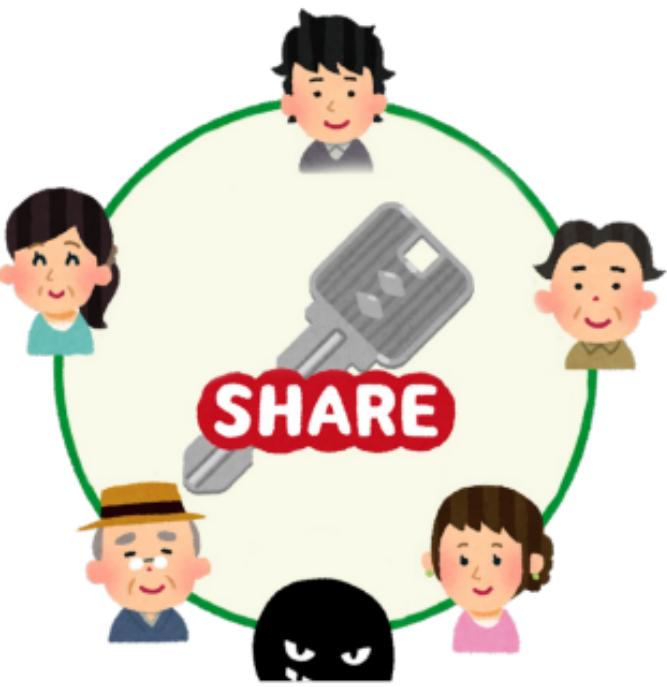
## 日本語バージョン



9

# Repudiation

複数のユーザ(principal)が鍵を共有しているため、攻撃者はこの共有鍵を使って別のprincipalのフリをすることでログの情報を混乱させることができる。



# elevation of privilege

## 日本語バージョン



10

# Repudiation

攻撃者は、弱い認証しか通っていない（または全く認証されてない）外部者として、検証なしに任意のデータをログに注入できる



10

# elevation of privilege

## 日本語バージョン



J

# Repudiation

攻撃者はログを編集することができ、  
編集された場合それを見分ける  
手段がない（おそらくログシステムに  
ハートビートオプションがないため）



# elevation of privilege

## 日本語バージョン



Q

# Repudiation

攻撃者に「私はやってない」と  
言わわれると、この主張が嘘で  
あることを証明できない

やってねーよ



# elevation of privilege

## 日本語バージョン



K

# Repudiation

このシステムにはそもそもログがない

てへぺろ (・ω<)



# elevation of privilege

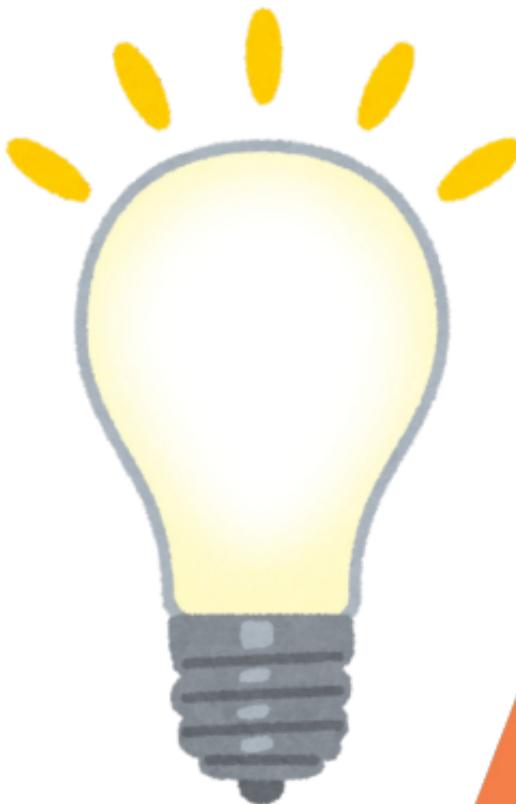
## 日本語バージョン



A

# Repudiation

新たな「否認」攻撃を考案した



A

# elevation of privilege

## 日本語バージョン



2

# Information Disclosure

パスワードストレッチングなどの対策が取られていないため、攻撃者が現実的な時間で総当たり攻撃を実行できる



# elevation of privilege

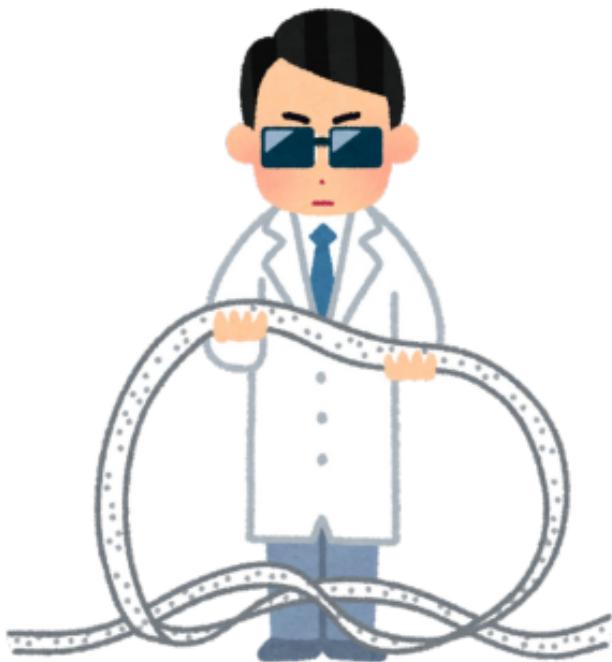
## 日本語バージョン



3

# Information Disclosure

攻撃者がエラーメッセージからセキュリティに関連した情報を得ることができる



# elevation of privilege

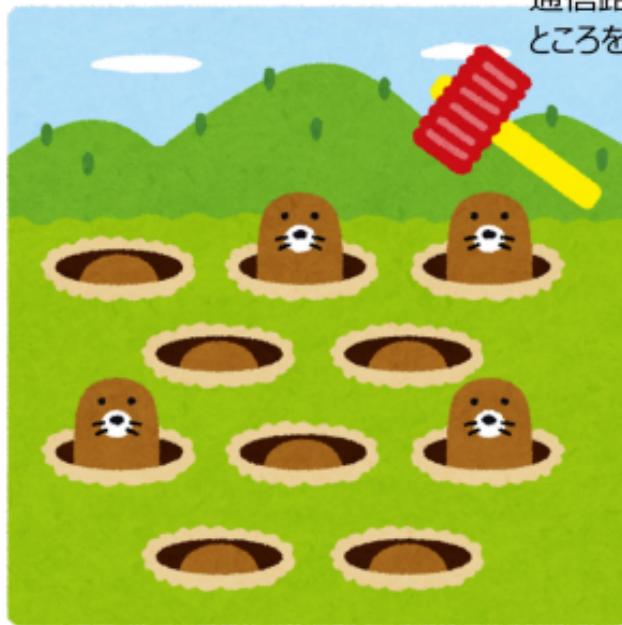
## 日本語バージョン



# 4

# Information Disclosure

暗号化された通信を使っているが、メッセージ（例：emailやHTTP cookie）が暗号化されてないため攻撃者が内容を見ることができる



# elevation of privilege

## 日本語バージョン



## 5

# Information Disclosure

標準的ではないアルゴリズムで暗号化  
されているため、攻撃者がデータや  
ドキュメントを読むことができてしまう



# elevation of privilege

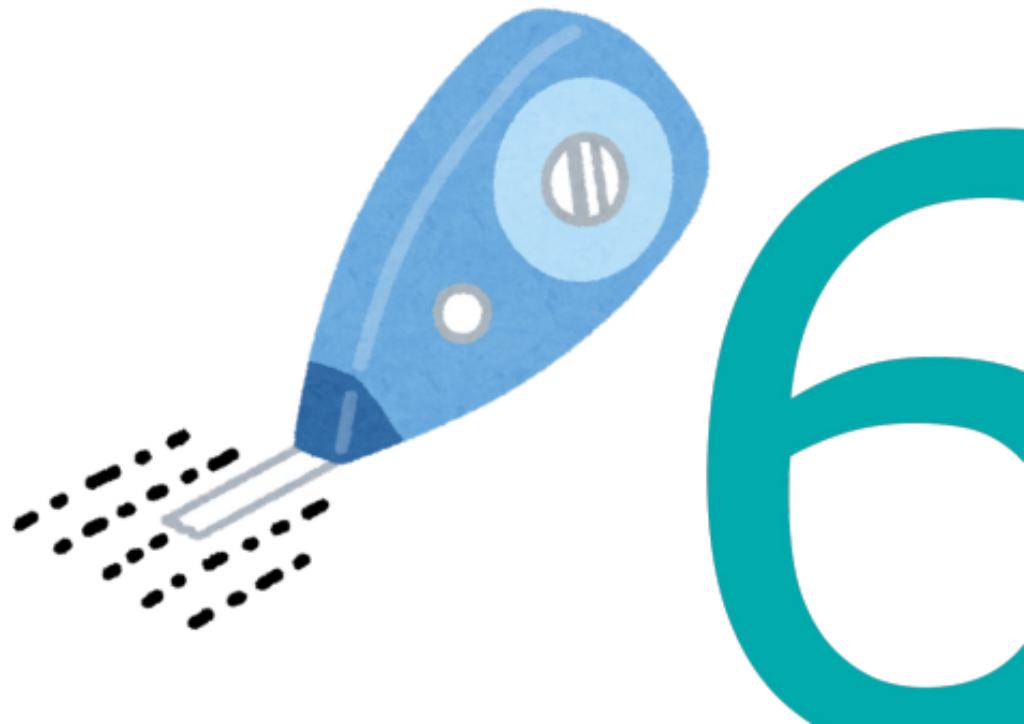
## 日本語バージョン



## 6

# Information Disclosure

攻撃者は、undo機能や編集履歴などにユーザがうっかり残してしまったデータを読むことができる



# elevation of privilege

## 日本語バージョン



# 7

# Information Disclosure

ネットワーク接続におけるエンドポイントを認証していないため、攻撃者が中間者攻撃を仕掛けられる



# elevation of privilege

## 日本語バージョン



# 8

# Information Disclosure

攻撃者は検索機能やログ機能などを悪用することにより情報にアクセスできる



# elevation of privilege

## 日本語バージョン



9

# Information Disclosure

攻撃者は、ACL設定に不備があるファイルの機密情報を読める



# elevation of privilege

## 日本語バージョン



10

# Information Disclosure

攻撃者はACLが設定されていない  
ファイルより情報を読める



1

0

# elevation of privilege

## 日本語バージョン



J

# Information Disclosure

攻撃者は暗号に使われている  
固定の鍵を見つけることができる



# elevation of privilege

## 日本語バージョン



Q

# Information Disclosure

通信経路が暗号化されていないため、  
攻撃者は通信経路全体を読むことができる（例：HTTPやSMTP）



# elevation of privilege

## 日本語バージョン



K

# Information Disclosure

暗号が一切使われていないため、  
攻撃者はネットワーク上のデータを  
読むことができる



# elevation of privilege

## 日本語バージョン



A

# Information Disclosure

新たな「情報漏えい」攻撃を考案した



# elevation of privilege

## 日本語バージョン



2

# Denial of Service

攻撃者は、認証システムを  
不安定または使用不能にできる



# elevation of privilege

## 日本語バージョン



3

# Denial of Service

攻撃者は、容易に交換可能な  
バッテリーを消耗させることができる  
(バッテリー、一時的)



# elevation of privilege

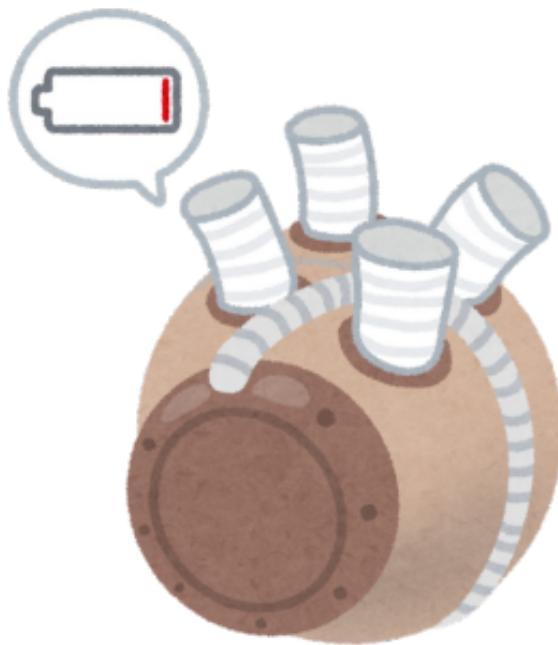
## 日本語バージョン



# 4

# Denial of Service

攻撃者は、携帯電話や移植された医療機器、到達困難な場所に封入されているなどの理由で交換が困難なバッテリーを消耗させることができる  
(バッテリー、永続的)



# 4

# elevation of privilege

## 日本語バージョン



# 5

# Denial of Service

攻撃者は、我々のクラウド予算を使い切ることができる  
(予算、永続的)



# elevation of privilege

## 日本語バージョン



## 6

# Denial of Service

攻撃者は、認証を突破することなく  
サーバを不安定または使用不能に  
できる。ただし攻撃者が攻撃を  
やめると問題は解決する  
(サーバ、匿名、一時的)



攻撃者は認証突破が不要

# elevation of privilege

## 日本語バージョン



## 7

# Denial of Service

攻撃者は、クライアントを不安定または使用不能にできる。攻撃者が攻撃をやめても問題は持続する  
(クライアント、認証済、永続的)



攻撃者は認証突破が必要

# elevation of privilege

## 日本語バージョン



# 8

# Denial of Service

攻撃者は、サーバを不安定または  
使用不能にできる。攻撃者が攻撃を  
やめても問題は継続する  
(サーバ、認証済、永続的)



攻撃者は認証突破が必要

# elevation of privilege

## 日本語バージョン



9

# Denial of Service

攻撃者は、認証を突破することなく  
クライアントを不安定または使用  
不能にできる。攻撃者が攻撃を  
やめても問題は継続する  
(クライアント、匿名、永続的)



攻撃者は認証突破が不要

# elevation of privilege

## 日本語バージョン



10

# Denial of Service

攻撃者は、認証を突破することなく  
サーバを不安定または使用不能に  
できる。攻撃者が攻撃をやめても  
問題は継続する  
(サーバ、匿名、永続的)



攻撃者は認証突破が不要

# elevation of privilege

## 日本語バージョン



J

# Denial of Service

攻撃者は、ログインサブシステムの動作を停止できる



# elevation of privilege

## 日本語バージョン



# Q

# Denial of Service

攻撃者は、このコンポーネントを悪用することにより、10:1のオーダーでDoS攻撃を增幅できる



# elevation of privilege

## 日本語バージョン



# K

# Denial of Service

攻撃者は、このコンポーネントを悪用することにより100:1のオーダーでDoS攻撃を增幅できる



# elevation of privilege

## 日本語バージョン



A

# Denial of Service

新たな「サービス拒否」攻撃を考案した



# elevation of privilege

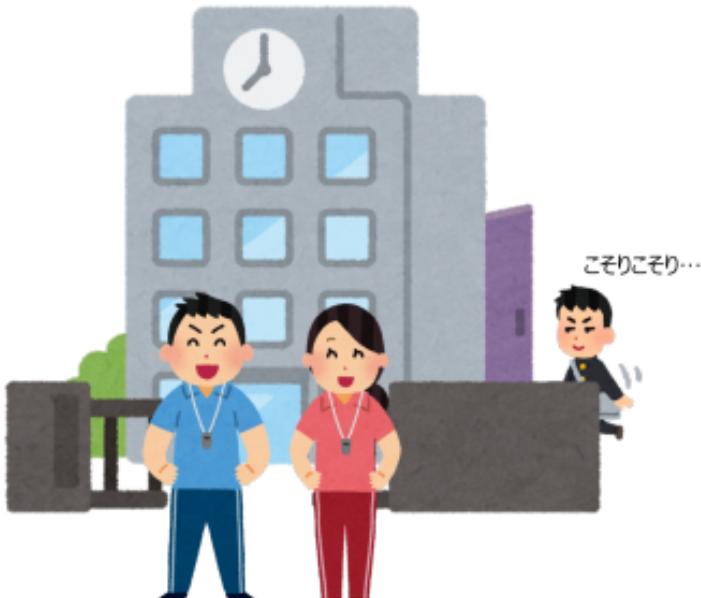
## 日本語バージョン



# 5

# Elevation of Privilege

攻撃者は、異なる結果を与える  
バリデーションを通すことによって、  
データを強制的に注入できる



持ち物チェック！！

# elevation of privilege

## 日本語バージョン



## 6

# Elevation of Privilege

実際には利用していない (.NET等の)  
フレームワークのパーミッションが許可  
されており、攻撃者はこれを悪用できる



# elevation of privilege

## 日本語バージョン



## 7

# Elevation of Privilege

攻撃者は、バリデーション可能なデータではなくポインターを信頼  
境界をまたいで渡すことができる



# elevation of privilege

## 日本語バージョン



## 8

# Elevation of Privilege

攻撃者は、バリデーション対象であるデータのコントロールを握っているため、信頼境界の向こう側に任意のデータを通すことができる



# elevation of privilege

## 日本語バージョン



9

# Elevation of Privilege

APIが返すデータに対してあなたのシステムが実施したバリデーションの内容を、API呼び出し元は知ることができない

これ、本当に検査済？



# elevation of privilege

## 日本語バージョン



10

# Elevation of Privilege

APIが返すデータに対してあなたのシステムが受け手に実施を期待するバリデーションの内容を、API呼び出し元は知ることが出来ない

い、一体何をしろと！？



APIレスポンス

# elevation of privilege

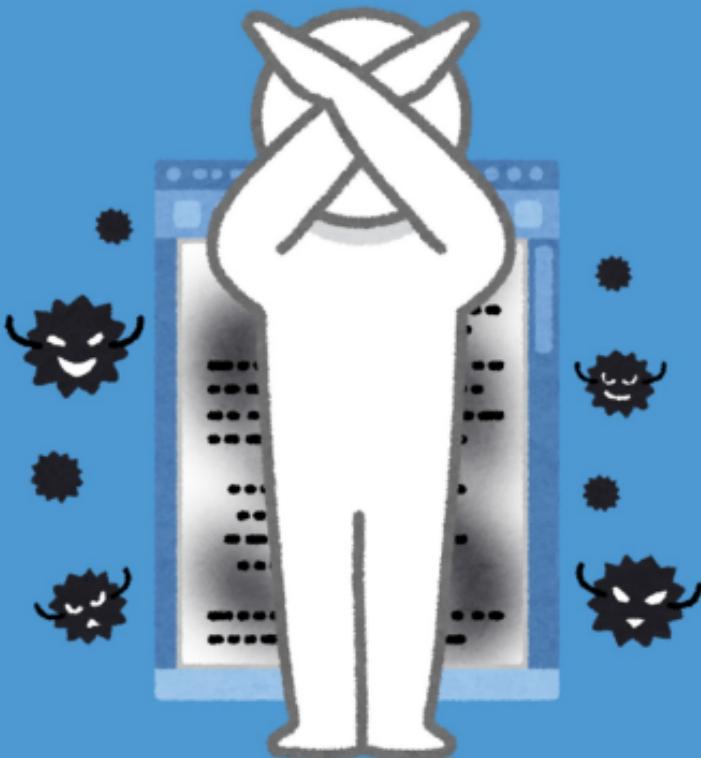
## 日本語バージョン



J

# Elevation of Privilege

攻撃者は、クロスサイトスクリプティングのように入力内容を他のユーザに反映できる



# elevation of privilege

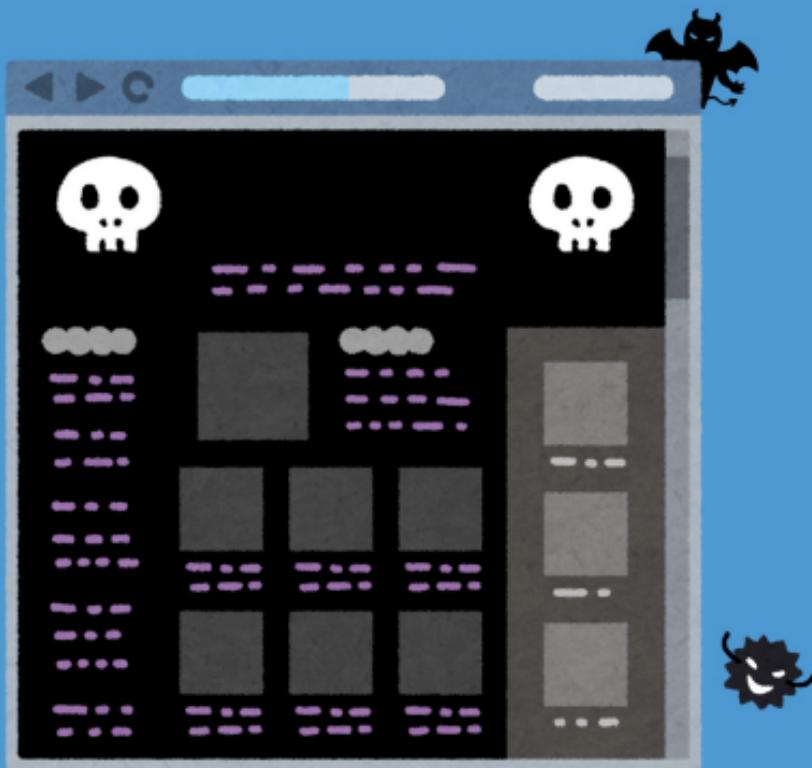
## 日本語バージョン





# Elevation of Privilege

ランダムURLのコンテンツが含まれて  
いる可能性のあるユーザー生成  
コンテンツをWebページに表示している



# elevation of privilege

## 日本語バージョン



# K

# Elevation of Privilege

攻撃者は、システムがより高い特権レベルで実行するコマンドを注入できる



# elevation of privilege

## 日本語バージョン



A

# Elevation of Privilege

新たな「権限昇格」攻撃を考案した



# Spoofing

2. 攻撃者は、サーバが通常使用するランダムポートやソケットをスクワッティング（占拠）できる
3. 攻撃者は、認証情報に対する総当たり攻撃をオンライン/オフラインで仕掛けることができる。この攻撃を遅らせる仕組みが存在しない
4. 上位レベルで認証が完了していると誤った期待をしているため、攻撃者が匿名で接続可能な状態になっている
5. サーバを識別する方法が多すぎるため、攻撃者がクライアントを混乱させることができる
6. クライアントに識別子が保存されておらず、再接続時に識別子の一貫性（鍵の永続性）がチェックされていないため、攻撃者がサーバになりますことができる
7. 攻撃者は、認証されていない/暗号化されていないリンクを介してサーバまたはピアに接続できる
8. 攻撃者は、サーバに保存されている認証情報を盗み、再利用できる（たとえば、鍵が誰でも読めるファイルに格納されている）

# Spoofing

9. パスワードを入手した攻撃者が、これを再利用できる  
(より強固な認証システムの導入を検討すべきである)
10. 攻撃者は、より弱い認証方式を選択できる。もしくは完全に認証を回避できる。
- J. 攻撃者は、クライアントに保存されている認証情報を盗み、再利用できる
- Q. 攻撃者は、認証情報の更新・復元フローを悪用できる  
(例：古いパスワードの提示なくアカウントが復元できてしまう)
- K. デフォルト管理パスワードが設定された状態でシステムをリリースしており、このパスワードを強制変更させていない
- A. 新たな「なりすまし」攻撃を考案した

# Tampering

3. 攻撃者は、標準的な暗号の代わりに採用されたオレオレ鍵交換プロトコルや完全性チェック機構の弱点を突くことができる
4. コードにおけるアクセス制御の決定が、セキュリティカーネルではなくあらゆる場所で行われている
5. タイムスタンプやシーケンス番号が存在しないため、攻撃者は検出されることなくリプレイ攻撃を仕掛けられる
6. 攻撃者は、コードが依存するデータストアに書きに入る
7. アクセスパーミッション確認前に名前を正規化していないため、攻撃者はパーミッションを迂回できる
8. ネットワーク上を流れるデータの完全性が担保されていないため、攻撃者はデータを改ざんできる
9. 攻撃者は、ステート情報を提供または制御できる
10. データストアのパーミッションが脆弱/オープン/オープンに等しい状態（例：Facebookアカウントを持つ全員）にあるため、攻撃者はデータストア内の情報を変更できる

# Tampering

- J. パーミッションが全ユーザーに付与されているか、そもそもACLが存在しないため、攻撃者は一部リソースに書き込みできる
- Q. 攻撃者は、バリデーション後のパラメータを、信頼境界を越えて変更できる（例えば、HTMLの非表示フィールド内の重要なパラメータや、重要なメモリーへのポインタ渡しなど）
- K. 攻撃者は、拡張ポイントを介してプロセス内に任意のコードをロードできる
- A. 新たな「改ざん」攻撃を考案した

# Repudiation

2. 攻撃者はログに任意のデータを渡してログの読み手を攪乱できる。  
ログ書き込み時に実施されるバリデーションに関する文書は存在しない
3. 低い権限しかもたない攻撃者が、ログ内の興味深いセキュリティ情報を読み取ることができる
4. 実装されたデジタル署名システムが脆弱であったり、署名を使用すべきところでMACを使用しているなどが原因で、攻撃者はデジタル署名を改ざんできる
5. 完全性担保の仕組みが弱いため、攻撃者はネットワーク上のログメッセージを改ざんできる
6. 攻撃者は、タイムスタンプのないログエントリを作成できる（もしくは、ログエントリにタイムスタンプがない）
7. 攻撃者はログをローテートさせることによりログを消滅させることができる
8. 攻撃者はログを消滅させたり、セキュリティ情報を混乱させることができる

# Repudiation

- 9. 複数のユーザ(principal)が鍵を共有しているため、攻撃者はこの共有鍵を使って別のprincipleのフリをすることでログの情報を混乱させることができる。
- 10. 攻撃者は、弱い認証しか通っていない（または全く認証されていない）外部者として、検証なしに任意のデータをログに注入できる
- J. 攻撃者はログを編集することができ、編集された場合それを見分ける手段がない（おそらくログシステムにハートビートオプションがないため）
- Q. 攻撃者に「私はやってない」と言われると、この主張が嘘であることを証明できない
- K. このシステムにはそもそもログがない
- A. 新たな「否認」攻撃を考案した

# Information Disclosure

2. パスワードストレッ칭などの対策が取られていないため、攻撃者が現実的な時間で総当たり攻撃を実行できる
3. 攻撃者がエラーメッセージからセキュリティに関連した情報を得ることができる
4. 暗号化された通信を使っているが、メッセージ（例：emailやHTTP cookie）が暗号化されてないため攻撃者が内容を見ることができる
5. 標準的ではないアルゴリズムで暗号化されているために攻撃者がデータやドキュメントを読むことができてしまう
6. 攻撃者は、undo機能や編集履歴などにユーザがうっかり残してしまったデータを読むことができる
7. ネットワーク接続におけるエンドポイントを認証していないため、攻撃者が中間者攻撃を仕掛けられる
8. 攻撃者は検索機能やログ機能などを悪用することにより情報にアクセスできる

# Information Disclosure

- 9. 攻撃者は、ACL設定に不備があるファイルの機密情報を読める
- 10. 攻撃者はACLが設定されていないファイルより情報を読める
- J. 攻撃者は暗号に使われている固定の鍵を見つけることができる
- Q. 通信経路が暗号化されていないため、攻撃者は通信経路全体を読むことができる（例：HTTPやSMTP）
- K. 暗号が使われていないため、攻撃者はネットワーク上のデータを読むことができる
- A. 新たな「情報漏洩」攻撃を考案した

# Denial of Service

2. 攻撃者は、認証システムを不安定または使用不能にできる
3. 攻撃者は、容易に交換可能なバッテリーを消耗させることができる（バッテリー、一時的）
4. 攻撃者は、携帯電話や移植された医療機器、到達困難な場所に封入されているなどの理由で交換が困難なバッテリーを消耗させることができる（バッテリー、永続的）
5. 攻撃者は、我々のクラウド予算を使い切ることができる（予算、永続的）
6. 攻撃者は、認証を突破することなくサーバを不安定または使用不能にできる。ただし攻撃者が攻撃をやめると問題は解決する（サーバ、匿名、一時的）
7. 攻撃者は、クライアントを不安定または使用不能にできる。攻撃者が攻撃をやめても問題は継続する（クライアント、認証済、永続的）

# Denial of Service

8. 攻撃者は、サーバを不安定または使用不能にできる。  
攻撃者が攻撃をやめても問題は継続する  
**(サーバ、認証済、永続的)**
9. 攻撃者は、認証を突破することなくクライアントを不安定または使用不能にできる。攻撃者が攻撃をやめても問題は継続する  
**(クライアント、匿名、永続的)**
10. 攻撃者は、認証を突破することなくサーバを不安定または使用不能にできる。攻撃者が攻撃をやめても問題は継続する  
**(サーバ、匿名、永続的)**
- J. 攻撃者は、ログインサブシステムの動作を停止できる
- Q. 攻撃者は、このコンポーネントを悪用することにより10:1のオーダーでDoS攻撃を增幅できる
- K. 攻撃者は、このコンポーネントを悪用することにより100:1のオーダーでDoS攻撃を增幅できる
- A. 新たな「サービス拒否」攻撃を考案した

# Elevation of Privilege (EoP)

5. 攻撃者は、異なる結果を与えるバリデーションを通すことによって、データを強制的に注入できる
6. 実際には利用していない (.NET 等の) フレームワークのパーミッションが許可されており、攻撃者はこれを悪用できる
7. 攻撃者は、バリデーション可能なデータではなくポインターを信頼境界をまたいで渡すことができる
8. 攻撃者は、バリデーション対象であるデータのコントロールを握っているため、信頼境界の向こう側に任意のデータを通すことができる
9. API が返すデータに対してあなたのシステムが実施したバリデーションの内容を、API呼び出し元は知ることができない
10. API が返すデータに対してあなたのシステムが受け手に実施を期待するバリデーションの内容を、API呼び出し元は知ることが出来ない

# Elevation of Privilege (EoP)

- J. 攻撃者は、クロスサイトスクリプティングのように、入力内容を他のユーザに反映できる
- Q. ランダムURLのコンテンツが含まれている可能性のあるユーザー生成コンテンツをWebページに表示している
- K. 攻撃者は、システムがより高い特権レベルで実行するコマンドを注入できる
- A. 新たな「権限昇格」攻撃を考案した

# About

## 脅威モデリング

Elevation of Privilegeゲームは、セキュリティの観点からあなたのシステムの設計を検証するための最も簡単なツールとして設計されています。このゲームは脅威モデリングを行う1つの方法であり、どのような開発グループであっても利用可能なものとなっています。このゲームはSTRIDEの脅威モデルをベースに設計されており、脅威分析のフレームワークと想定される脅威の具体例を提供します。

STRIDEは、以下の想定される脅威分類の頭文字をとったものです。

**Spoofing (なりすまし)**: 他のものや他の誰かになります

**Tampering (改ざん)**: データやコードを変更する

**Repudiation (否認)**: アクションを実行していないと主張する

**Information Disclosure (情報漏えい)**: 権限のない人に情報を公開する

**Denial of Service (サービス拒否)**: ユーザへのサービスを拒否/機能低下させる

**Elevation of Privilege (権限昇格)**: 正当な許可なく権限を取得する

EoPカードゲームのオリジナル版やスコアシートなどは

<https://www.microsoft.com/en-us/download/details.aspx?id=20303> より入手できます。



# SDL

The Elevation of Privilege game is a fun and easy way to get started understanding the security of your systems by threat modeling. As you discover and correct design-level security problems, it's worth thinking about the other ways security issues can creep into your code. Microsoft has a large collection of free resources available to help you get started with the Security Development Lifecycle (SDL).

To learn more about threat modeling and the Microsoft Security Development Lifecycle, visit our website at [microsoft.com/sdl/](http://microsoft.com/sdl/)