Vol.2021-IS-158 No.6
2021/12/4

DFD を利用した簡易的な脅威分析手法に関する研究

奥山順子[1]　藤本正代[2]

概要：小規模なアプリケーション（ソフトウェア）開発の現場においては，セキュリティを意識した設計・開発の優先度は，機能実装や早期のリリースと比べて優先度が低くなりがちである．セキュリティ人材や予算の不足などの理由から対象システムに十分なセキュリティ対策を実装することは難しい．これに対して，脅威分析の知識や人材が十分でない場合でも実施可能な，簡易的であってもかつ実施されて当然と考えられる脅威への対策がわかるような脅威分析手法がないかを考えた．その方法として，A. Shostack が提唱した脅威分析手法をベースに OWASP Top10 で示されている脅威をすべて含む形で，簡素化した脅威分析手法を作成した．
簡素化した脅威分析手法の有効性を，仮想的に作成した学務支援アプリケーションサービスに適用し検証した．
その結果，提案する簡易的な脅威分析手法を適用した場合，専門知識がなくても，OWASP Top 10 に含まれるような脅威がどこに存在するのかを図示することができることが示せた．一方，取るべき対策がアプリケーション上の設計だけではなく，キャッシュやシステム固有の設定ファイルや設定値に対して言及する場合は，DFD で描いた上位レベルのダイアグラムでは把握しづらいことなどがわかった．
今回は，仮想的に作成した学務支援アプリケーションサービスを対象として有効性の検証を行ったが，今後はより小規模な開発で手がけることが多いアプリケーションサービスについての有効性を検証する必要がある．

キーワード：小規模開発，DFD，脅威分析，STRIDE

1. はじめに

アジャイル開発全盛の現在では，現在のアプリケーション開発の現場において重視されるのがより短期間の開発，リリースである．一方でそのリリースしたシステムで情報漏洩が起きないか，不正アクセスによりシステムに侵入をされないかを十分に検討した上で設計されたのか，特に IT 人材が充実していない中小規模の開発ベンダーではそれが十分に対応できていない実態があるように感じられる．

Web アプリケーションなどに関するセキュリティ対策では，調査対象の 37%が「セキュリティ設計をしていない」と回答，またセキュリティ管理ができていない理由の上位に，「専門知識の不足」，「組織内の優先順位の低さ」，「十分な予算が確保できない」といった理由が上がっている．[1]

小規模 Web サイト運営者へのアンケートでは，70%が脆弱性やセキュリティ対策の必要性を認識しながらも，60%はそのための情報がどこにあるかをわかっていないとしている．[2]

直近のニュースを見ても，もちろん有名企業で発生した情報漏えい事故は大きく報道されるが，セキュリティニュースに特化したニュースサイトでも中小企業で発生した自社サイトからの個人情報漏えい，不正アクセスといったキーワードでの報道がある．再発防止策を講じてサービスを再開するところもあるが，サービス提供の断念も少なからず見られる．

本研究においては，必ずしもセキュリティの専門家がいない中小規模の開発現場，組織で，特に情報漏えい事件が発生した場合の影響が大きいと考えられる Web アプリケーション開発を対象としたセキュリティ対策の検討に役立つ手法がないかを研究した．

2. 先行研究

アプリケーションがセキュアであるためには，脆弱性のないアプリケーション開発を行わなければならない．

大久保は，セキュリティ要求工学[4]の中でソフトウェア開発での十分なセキュリティ要求の獲得のためには“ソフトウェアやデータに対し，どのようなセキュリティ脅威がありうるかを洗い出し，そのリスクの大きさを見積るという 作業が必要になる.この作業を脅威分析と呼ぶ”とし，この技法としてゴール指向分析，エージェント指向分析をあげ，attack tree 解析，ミスユースケースなどを手法の例として紹介している．また，セーフティとセキュリティの相違を分析する中で “情報セキュリティの機密性，完全性，可用性の性質や，脅威分析の際に用いられる STRIDE などの脅威分類”が情報セキュリティの観点と指摘している．

中野らは，システムに対する脅威分析におけるコスト及び属人性低減に向けた手法の提案 [5]において，“脅威分析とは，システムやデバイスにおいてセキュリティ事故や故障の原因となる脅威の有無を分析し，必要な対策を明確にする作業である.”とし，“従来手法と比較してコストの少ない脅威分析手法を開発，提案”している．“既存の手法で用いられることの多い DFD, STRIDE, Attack Tree を挙げ,”手法を展開し，脅威分析ツールを使用して重要資産を絞り込み重要な箇所を絞り込むことで，専門の分析者の手法と比較した検証を行なっている．

1 情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY
2 情報セキュリティ大学院大学 教授

中野らが提示している脅威分析の従来手法として，以下の 3 点が挙げられている．

2.1.1 DFD を用いたシステム構成要素の洗い出し

2.1.2 STRIDE による脅威抽出と分類

2.1.3 Attack Tree による脅威の分析

検討の対象となるシステムの構成要素にどのような脅威があるかを把握するには，まず DFD を用いてシステム構成要素を洗い出す．システムの構成要素は，システム外要素（外部要素などともいう），プロセス，データフロー，データストアの 4 種類の要素で記述する．

次に，脅威を STRIDE という Microsoft 社[a]Adam Shostack 氏が開発した，脅威の分類を行う手法[6]がある．

STRIDE は，6 種類の脅威を示す英単語の頭文字で，Spoofing（なりすまし），Tempering （改ざん），Repudiation（否認），Information Disclosure（情報漏えい），Denial of Service（サービス妨害），Elevation of Privilege（権限昇格）を指す．

DFD の要素と STRIDE の脅威との間には，表 1 の関係があり，4 つのシステム構成要素に対して，該当する脅威の種類を決める．この表では，“「システム外要素」については，改ざんと情報漏えい，サービス妨害及び権限昇格は本質的に生じない．対してユーザーやクライアントへのなりすまし，通信の否認などは生じうるという事を意味する．”としている．表内の記号は，〇：対応がある　△：場合によっては対応がある　-：対応しないことを示す

表 1 DFD 要素と STRIDE(脅威)の着目箇所

	S	T	R	I	D	E
プロセス	〇	〇	〇	〇	〇	〇
データストア	-	〇	△	〇	〇	-
データフロー	-	〇	-	〇	〇	-
外部要素	〇	-	〇	-	-	-

Attack Tree は，“脅威が生じうる原因を分析する手法”で，“木構造で表現される図”で，Attack Tree による脅威の分析は，表 1 で示した，着目箇所において“抽出した脅威について，それぞれが生じる原因を調査し，詳細に解析を行う.”ことである．この Attack Tree は，類似した内容であっても原因の発見もれを防ぐために脅威の数だけ作成する必要があるとしている．

3. 本研究で提案する簡易的な脅威分析手法

本研究では，アプリケーションの対象を，『Web アプリケーション』とし，事前に脅威を特定したものを用いて，簡易的に実施できる脅威分析手法を提案する．

3.1 簡易的な脅威分析手法の考え方

先行研究で提示された手法は，脆弱性のないアプリケーション開発にとっては有効であることがわかったが，実際の分析には高度な知識が必要で，それによって専門能力のある技術者と分析にかかる工数が必要であることも同時にわかった．しかし，1.はじめにでも言及したとおり中小企業で見られる小規模な開発組織では，脅威分析の専門性の高い人材を参画させることも，脅威分析のために工数を組み込むことも容易ではない実態がある．

本研究では，アプリケーションの対象を，『Web アプリケーション』とした場合に，事前に脅威を特定したものを用いることで，特に作業負荷の脅威分析の作業を簡素化できるのではないかと考えた．簡素化の弊害としては，重要な脅威の特定漏れが想定されるため，以下の３つの観点を網羅することで見落としてはならない脅威を特定することとした．

1. OWASP Top 10 [7]
2. STRIDE Reference Sheets [8]
3. 攻撃事例[3],[9]

また，特定した脅威に対する対策については，以下を参照した．

1. OWASP Top 10
2. 安全なウェブサイトの作り方[11]

参照した資料の概要は次の通りである．

(1) OWASP Top 10

OWASP Top 10 は，「Open Web Application Security Project（国際ウェブセキュリティ標準機構）」という，非営利組織が作成した，Web アプリケーションのセキュリティに関する脅威や危険性のトレンドを危険度の高い 10 種類の脆弱性を整理して，攻撃シナリオ，対策等の観点でまとめた資料である．研究を開始した時点では，2017 年 11 月 20 日に発行された内容に基づいていたが，2021 年 9 月 24 日に最新の OWASP Top 10[12]が公開された．内容の差分については現在検討中である．

(2) STRIDE Reference Sheet

OWASP 内で紹介されている DFD と STRIDE の関係及び該当する脅威を例示した資料．

(3) 安全なウェブサイトの作り方

“IPA が届出を受けた脆弱性関連情報を基に，届出件数の多かった脆弱性や攻撃による影響度が大きい脆弱性を取り上げ，ウェブサイト開発者や運営者が適切なセキュリティを考慮したウェブサイトを作成するための資料”[12]で，11 の脆弱性を取り上げ，発生しうる脅威と対策について解説をしている．

(4) 攻撃事例

日本で Web アプリケーションに対して発生した攻撃事

a Microsoft，またはその他のマイクロソフト製品の名称および製品名は，米国 Microsoft Corporation の，米国およびその他の国における商標または登録商標です.

例として, IPA に届出された Web アプリに対する攻撃[11], アプリケーションの脆弱性に起因して発生した情報漏えいのニュース掲載[3]の内容を分類した結果を加味して内容を整理した．それぞれの事象について，該当する脅威が STRIDE のどれに相当するのかを分類してカウントした結果が表 2 となる．

表 2 公開された Web アプリケーションへの脅威の分類

	S	T	R	I	D	E
IPA 届出	5	12	-	11	-	6
ニュース掲載	1	9	1	4	1	-

IPA 届出[9] : 2021 年 1 月 - 6 月までに発生した Web アプリに対する攻撃 25 例について分類した結果（事例数には重複あり）

ニュース掲載[3] : 2021/7/1-9/9 までに Security NEXT に掲載された Web アプリに対する攻撃のニュースから 22 例について分類した結果（事例数には重複あり）.

Web アプリケーションへの攻撃として注目すべき脅威としては，改ざんと情報漏えいの内容が数多く出現しているため，これらの脅威への対策は必須である．

Tampering (改ざん)のカテゴリーに含まれる脅威として数多く登場したのが，ログイン ID を改ざん，Web サイト，アプリ，ファイルの改ざん，不正なプログラム，ファイルの設置，外部サイトへの誘導

Information Disclosure (情報漏えい) のカテゴリーに含まれる脅威として数多く登場したのが，個人情報（メールアドレス，パスワード）漏えい，クレジットカード情報漏えい

本研究では表 1 の枠組み（縦軸と横軸の項目）をベースに，STRIDE Reference Sheet 内で提示された脅威の例を大まかに当てはめ，それぞれに相当する具体的な脅威を，OWASP Top 10，安全なウェブサイトの作り方（発生しうる脅威）及び攻撃事例で使用される脅威として記述された表現に当てはめ，また対策については，OWASP Top 10，安全なウェブサイトの作り方でカテゴライズされた番号で一覧にしたものを,「脅威と対策の標準パターン」として提案する．

3.2 提案する脅威と対策の標準パターン

3.1 で提示した資料から Web アプリケーションに起きうるかつ見落としてはいけない脅威を確認し，STRIDE で分類される脅威について，その発生しうる箇所を DFD の各要素に当てはめ，DFD と STRIDE の交差する箇所に 3 から 5 程度の主要な脅威例が当てはまるように作成した「脅威と対策の標準パターン」の一部が，
表 3 となる．

実際には，取り上げた脅威の数が目標とした数量に足りていない箇所がある．

表 3 Web アプリケーション開発における標準パターン(一部)

DFD要素	Spoofing	Tampering	Repudiation	Information Disclosure	DoS	Elevation privilege
脅威と対策	なりすまし	改ざん	否認	情報漏えい	サービス妨害	権限昇格
プロセス						
脅威の例	1) 不正なプロセスが正規のプロセスになりすます 2) システムが乗っ取られる	1) Webサイトの改ざん 2) Webアプリの改ざん 3) 不正なプログラム、ファイルの設置 4) キャッシュサーバーの汚染で、偽のWebサイトを閲覧	1) 適切なログを生成しない	1) エラーメッセージから秘匿情報の読み取り 2) 起きたエラーの内容からマシンに関する情報などの読み取り 3) 暗号化キーの読み取り	1) CPUやメモリーの処理能力を超えた要求 2) プログラムの異常な終了	1) 適切に処理できない入力の送信 2) 権限が適切にチェックないログイン 3) 認証回避による不正ログイン
安全なウェブサイトの作り方	1.1, 1.2, 1.8, 1.10, 1.11	1.1, 1.2, 1.3, 1.5, 1.7,		1.2, 1.3,	1.2, 1.10	1.1, 1.11
OWASP Top 10 (2017)	A1, A6, A9	A6, A7, A8,A9	A10	A4, A5, A6		A1, A2, A5, A6, A8
データストア						
脅威の例		1) ファイルの改ざん 2) 正規の利用者本人のみ利用できる情報の改ざん	1) ログファイルが消去される	1) 個人情報の漏えい 2) クレジットカード情報の漏えい 3) 正規の利用者本人のみ利用できる情報の閲覧	1) データストアの容量をいっぱいにされる	
安全なウェブサイトの作り方		1.1, 1.4, 1.6, 1.8		1.1, 1.4		
OWASP Top 10 (2017)		A9	A10	A1, A3, A6, A9	A4	
データフロー						
脅威の例		1) ネットワーク上で、外部サイトへ誘導 2) ネットワーク上のデータを改ざん		1) 通過するファイルを読み取られる 2) フロー先をリダイレクトされる 3) トラフィックを解析される	1) ネットワークリソースを消費される	

例えば，プロセスで生じる可能性のあるなりすましの脅威として，1) 不正なプロセスを，正規のプロセスになりすます，2) システムが乗っ取られる，が考えられることを示す．

とるべき対策には，安全なウェブサイトの作り方[11], OWASP Top 10[7]で参照するべき項目の番号を記載した. プロセスにおけるなりすましの対策は，安全なウェブサイトの作り方の 1.1（SQL インジェクション），1.2（OS コマンドインジェクション）， 1.3（パス名パラメータの未チェック／ディレクトリ・トラバーサル），1.5（クロスサイト・スクリプティング），1.7（HTTP ヘッダ・インジェクション）の章に記載された対策（根本的解決，保険的対策）を参照，OWASP Top 10 からは，A1（インジェクション），A6（不適切なセキュリティ設定），A9 (既知の脆弱性のあるコンポーネントの使用）の項に記載された対策（防止方法）を参照して検討することを示している．

3.3 提案する簡易的な脅威分析手法

本研究では，アプリケーションの対象を,『Web アプリケーション』とし，Attack Tree を作成する工数を削減し，事前に特定した脅威を用いて分析する手順を作成した．その脅威分析の手順は次のとおりである．

(1) 脅威と対策の標準パターンの作成（3.2 の手順で作成した表 3）
(2) 対象システムの DFD を作成し，重要な情報への入力ルートを特定
ここでは脅威分析の対象となるシステムの上位レベル（主要な機能，情報の単位でプロセス，データストアを配置）DFD を作成し，その中でも重要な情報の保管（データストア）に着目する．
(3) (1)で作成した表 3 より，該当する脅威をピックアップ

4. 提案手法の検証

提案する「脅威と対策の標準パターン」の効果については，架空の Web アプリケーションとして「学務支援システム」をケーススタディとして作成し，同じ DFD に対して理想的な脅威分析と，簡易的な脅威分析でどのような差があるかを比較することで提案手法が有効であることを証明したいと考えた．

提案する簡易手法では，理想的な脅威分析手法に対して専門家等による脅威の洗い出しの作業が簡素化（事前に設定された表を使う方法）されている．この 2 つの手法について，分析結果にどのような違いが生じるかを検証した．

4.1 検証方法

4.1.1 検証対象の Web アプリケーションシステム

Web アプリケーションの事例として，大学の学生，教員，事務管理者が利用する学務支援システムをモデルとしたシステムを題材とした．

このシステムは，大学管理者が，個人情報を含む学籍情報と講義の時間割情報を登録する，教員がシラバス情報，学生の成績評価情報を登録する，学生が受講する講義情報を登録するといった機能を持つものと想定したものである．

図 1 は，筆者が脅威分析機能を持たない作図ツールを使用して作成した比較検証の基準となる DFD である．現実で使用された設計書などに基づいてはいないので，あくまでも研究のために想定した機能に限定している．

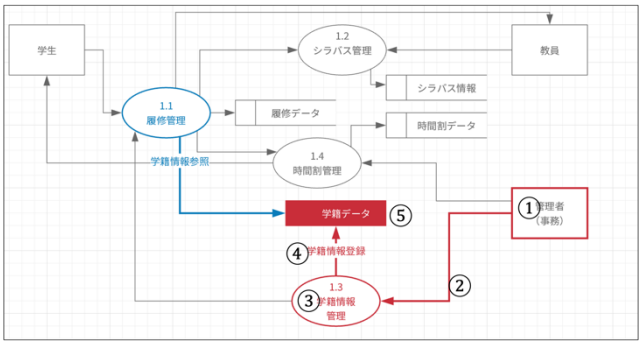

図 1 比較検証するシステムの DFD

4.1.2 理想的な脅威分析手法

本研究の脅威を洗い出す理想的な比較検証の手法を実践するために利用可能なツールとして，Microsoft Threat Modeling Tool（以下 TMT）[13]を使用した脅威分析の結果を利用した．

TMT は，中野らの研究の検証でも使用されたツールだが，“すべての開発者が簡単に脅威をモデリングできます．”と紹介[13]されている，理想的な脅威分析の実施に近く，PC で利用可能な無料のツールである．

TMT の画面上に DFD を作図し，各要素に命名，定義を与えたのち，TMT が持つ分析機能を実行して自動的に脅威分析の結果レポートを出力する．STRIDE で分類された脅威の出力結果を見て，開発者が設計の欠陥の可能性，攻撃の可能性を検討し，対策を実施するものである．

脅威と対策のリストとしては以下の項目を参照した．

- 名称（脅威を示す）
- カテゴリー（”STRIDE”に相当）
- 説明（脅威の詳細）
- とりうる対策（脅威に対して考えられる対策）

TMT は DFD に要素の属性情報を入力することでより正確な分析ができるとしているが，今回の検証ではプロセスに Web アプリケーションという属性だけ設定し，他は定義可能な属性は設定しなかった．

4.2 検証の結果

まず，一般的な脅威分析手法における脅威の洗い出しとして，提案する手法の適用対象が，小規模な開発規模を想定しているので，その環境で使用が想定される別の対比する手法として，無料で提供されており，脅威分析の初心者・初級者でも使えることを表明している TMT を使った脅威の洗い出しの作業を実施した．

4.2.1 提案する手法による脅威分析

図 1 に対して，表 3 の標準パターンを用いて，個人情報を含む「学籍データ」への入力ルート上に出現する脅威を取り出した．

図 1 で，①外部要素の管理者（事務），②データフロー（命名なし，DF01 とする），③プロセスの学籍情報管理，④データフローの学籍情報登録，⑤データストアの学籍データの順に入力ルートがあると考える．これに，表 3 で提示した標準パターンにある提案手法で該当する脅威をピックアップした結果が表 4 になる．

表 4 選択した脅威

DFD上の要素	選択した脅威
①管理者（事務）	1) ユーザーになりすます 2) 役割になりすます 3) ブラウザーに保存されたCookieを取得される 4) 任意のCookieをブラウザーに保存させられる 5) 操作を否認する 6) 他人のアカウントや決済手段で処理する
②DF01	1) ネットワーク上で、外部サイトへ誘導 2) ネットワーク上のデータを改ざん 3) 通過するファイルを読み取られる 4) フロー先をリダイレクトされる 5) トラフィックを解析される 6) ネットワークリソースを消費される
③学籍情報管理	1) 不正なプロセスが正規のプロセスになりすます 2) システムが乗っ取られる 3) Webサイトの改ざん 4) Webアプリの改ざん 5) 不正なプログラム、ファイルの設置 6) キャッシュサーバーの汚染で、偽のWebサイトを閲覧 7) 適切なログが生成されない 8) エラーメッセージから秘匿情報の読み取り 9) 起きたエラーの内容からマシンに関する情報などの読み取り 10) 暗号化キーの読み取り 11) CPUやメモリーの処理能力を超えた要求 12) プログラムの異常な終了 13) 適切に処理できない入力が送信される 14) 権限が適切にチェックされずログインできる 15) 認証回避による不正ログイン
④学籍情報登録	②DF01と同じ
⑤学籍データ	1) ファイルの改ざん 2) 正規の利用者本人のみ利用できる情報の改ざん 3) ログファイルが消去される 4) 個人情報の漏えい 5) 正規の利用者本人のみ利用できる情報の閲覧 6) データストアの容量をいっぱいにされる

4.2.2 TMT を利用した脅威分析

筆者が TMT を使用して図 1 と同じ内容の DFD を作図したものが図 2 となる．

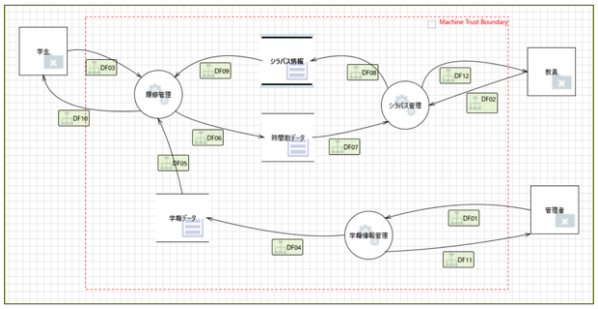

図 2　TMT で作図した DFD

その後 TMT の脅威分析レポート機能を実行してレポートを出力した一部が図 3 になる．

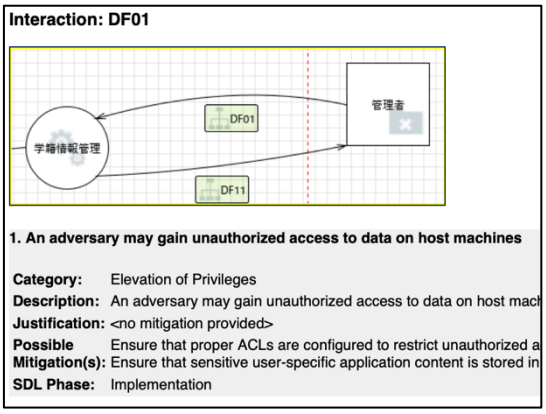

図 3 TMT による脅威分析レポート出力例（DF01 部分）

図 3 に出力された脅威別ごとに STRIDE に対応したカテゴリーなどが表示されるので，これを見て管理者から学籍データへの入力ルート上にある要素に対して表示された，脅威分析結果（脅威の検出数，その内容）を確認した．

4.2.3 検証の結果

提案手法と TMT による 2 つの手法で，守るべき個人情報であると見立てた「学籍データ」への入力ルート（図 1 の①から⑤）上に出現した脅威の数を単純に比較した結果が表 5 になる．

表 5 分析した脅威数の比較

	S	T	R	I	D	E
提案手法	6	8	3	8	4	3
TMT	7	9	1	9	1	5

標準パターンから転記した脅威と，TMT を利用した脅威の単純な検出数比較では大きな差はなく，簡易的であるとした提案手法でも十分な脅威を検出できる可能性を確認した．

一方で，OWASP Top 10 や安全なウェブサイトの作り方での脅威が発生するメカニズムや対策の詳細に関する解説では，マシンのリソース，OS やミドルウェアへの言及，設定ファイルやキャッシュに対する言及があり今回作成したような上位レベル（主要な機能の単位で要素を作図）の DFD で表現することが難しいこともわかった．

5. まとめ

今回の研究では，小規模開発で実践する機会が多いと思われる Web アプリケーションを想定した標準パターンを作成して，簡易的な手法として適用した．アプリケーションを特定して標準パターンを作成すれば簡易的な手法であっても有効な脅威分析ができるのではないかと考える．

今後さらに，個々の脅威に対応する対策をあらかじめ決めておくことにより，Attack Tree を作成するといった一定のスキルを有した専門家がいないような状況においても，ある程度の対応策の実装を開発プロセスに組み込むことができると考える．

セキュリティバイデザイン，シフトレフトとして提唱されているように[14]，セキュリティの脅威分析は，より早期の段階で検討することで，後続の工程で発覚した対策のコストより少なく済むと言われている．

今回適用しようとする小規模な開発を考えた場合，早期の検討段階では機能や仕様，構築するインフラが確定しておらず本格的な脅威分析を早期に実施すること自体考えにくい．

そもそも DFD を作図して設計を行うことがあまり見られなくなった今，「まず，DFD を描きましょう」では設計者，開発エンジニアも躊躇するであろう．このため，本論文で提案する脅威分析の標準パターンのほかにも，DFD のパターンを用意しなければ現場への展開が望めないと考えている．

小規模な開発現場での実態をさらに確認してセキュリティ脅威分析の取り組みやすい第一歩になる実績を作っていきたい．

参考文献

[1] 金根學，原田要之助．WEB アプリケーションのセキュリティマネジメント (情報処理学会研究報告　Vol.2017-EIP-78 No.21) .
[2]「小規模ウェフブサイト運営者の脆弱性対策に関する調査報告書」. (情報処理推進機構，2021 年 3 月) .
[3] Security NEXT, https://www.security-next.com, (参照 2021-10-30)
[4] 大久保 隆夫．安全工学.セキュリティ要求工学.2015 年 54 巻 6 号 p. 460-463.
[5] システムに対する脅威分析におけるコスト及び属人性低減に向けた手法の提案. https://www.ffri.jp/assets/files/research/research_papers/Threat_Analysis_Method_Reducing_costs_and_variability_in_results.pdf,

(参照 2021-10-30) .
[6] A. Shostack, Threat Modeling: Designing for Security, Wiley, Feb 2014.
[7] OWSP Top 10 － 2017 日本語版(OWASP 2017 年 12 月) .https://github.com/owasp-ja/Top10/blob/master/2017/ja/OWASP%20Top%2010-2017(ja).pdf, (参照 2021-10-30) .
[8] STRIDE Reference Sheets, https://owasp.org/www-pdf-archive/STRIDE_Reference_Sheets.pdf,(参照 2021-10-30) .
[9] 独立行政法人情報処理推進機構.コンピュータウイルス・ 不正アクセスの届出事例［2021 年上半期（1 月～6 月)］.https://www.ipa.go.jp/files/000093083.pdf,(参照 2021-10-30) .
[10] OWASP Top 10 2021 の紹介.https://owasp.org/Top10/ja/ , (参照 2021-10-30) .
[11] 独立行政法人情報処理推進機構.安全なウェブサイトの作り方.https://www.ipa.go.jp/files/000017316.pdf , (参照 2021-10-30) .
[12] 独立行政法人情報処理推進機構.安全なウェブサイトの作り方.https://www.ipa.go.jp/security/vuln/websecurity.html (参照 2021-10-30) .
[13] Microsoft Threat Modeling Tool.https://docs.microsoft.com/ja-jp/azure/security/develop/threat-modeling-tool , (参照 2021-10-30) .
[14] 独立行政法人情報処理推進機構.セキュリティ・バイ・デザイン入門,https://www.ipa.go.jp/files/000055823.pdf,(参照 2021-10-30).