企業における セキュリティ分析技術の実効性

大久保 隆夫
(株)富士通研究所

「開発者は穴（脆弱性）をすべてふさがないとならないが，攻撃者は穴を１つ見つけるだけでよい」と言われる．このような開発者に不利な状況の中で，セキュアなソフトウェアを構築するためには，従来の機能中心の分析，設計手法では十分ではない．本稿では，企業に身をおく筆者のシステム・インテグレーションやソフトウェア製品開発のセキュア化支援活動の経験に基づき，企業のソフトウェア開発におけるセキュリティ要求分析，設計の現状と課題について論じる．特に，実用的なセキュリティ分析手法として提案されている脅威モデリングを例として，既存の手法，ツールの有効性と問題点について明らかにする．

セキュリティには攻撃者の観点が必要

ソフトウェア開発においてセキュリティの問題である脆弱性の排除は，重要な問題のはずである．しかし，今日も依然として情報処理推進機構（IPA）への脆弱性届出，セキュリティ事故が後をたたず，問題は根本的に解決されているとは言いがたい．セキュリティの脆弱性が従来のデバッグやテストでも排除できないのはなぜだろうか．

セキュリティは何らかの脅威，特に人為的な攻撃などの脅威に対抗するものである．このようなセキュリティを考慮してソフトウェアの分析，設計をする場合，分析者，設計者は従来のような「通常の使い方」のシナリオだけでなく，攻撃する側の観点に立って，「通常でない使われ方」を想定しつつ，必要なセキュリティの要求や設計仕様を決定する必要がある．この「攻撃者の観点」は従来のソフトウェア開発における要求分析や，設計手法にはない考え方だったため，この観点からの検討不足が，不十分なセキュリティ要求や脆弱性を許す設計の一因となっている．

企業におけるセキュリティ分析技術の実効性

セキュリティにおいては，実装面における対策のみが注目されがちだが，他のソフトウェアの構成要素同様，開発のできるだけ早期の段階，可能であれば要求分析段階から認識され，対応されることが必要である．にもかかわらず，ソフトウェア開発，特にシステム・インテグレーション（SI）の現場では，要求分析からセキュリティが考慮されている例は，コモンクライテリア（CC, ISO/IEC 15408）認証を要求されている場合を除き，まだ数が多いとは言えない．SI 現場のセキュリティ支援にかかわってきた筆者の経験に基づき，セキュリティ要求分析に必要だが，現状の企業では不足していると考えられるものを次に挙げる．

(1) セキュリティ意識

セキュリティ意識の問題は，実装担当者だけの問題ではない．前述のように，開発の早期段階でこそ，セキュリティを意識する必要性は高い．しかし，セキュリティ機能そのものの実現を目的とする製品以外のソフトウェアでは，ステークホルダ（特に企画側，権利者，顧客）のセキュリティ意識は決して高いとはいえない．むしろ，このような現場では主要機能に関する要求が最優先され，セキュリティの優先度は概して低いか，まったく無視されることが多い．だが，脆弱性はセキュリティ機能だけでなく，あらゆる機能で発生し得るものである．事実，最近多発している SQL インジェクション攻撃の対象になり，Web 改竄や個人情報漏洩事故を起こした Web サイトのほとんどは，セキュリティ機能の提供を主としない一般の Web サイトである．

(2) セキュリティ知識（人材）

セキュリティ意識欠如の大きな要因は，セキュリティ知識の不足にあると言える．プログラマにセキュアプログラミングのノウハウがなければ，自然にソフトウェアが安全になることはあり得ないのと同様，ソフトウェアの扱う資産に対してどのようなセキュリティリスクが存在するのか，正確に理解していないゆえに，リスクへの対策が行われず，結果として脆弱なソフトウェアができて

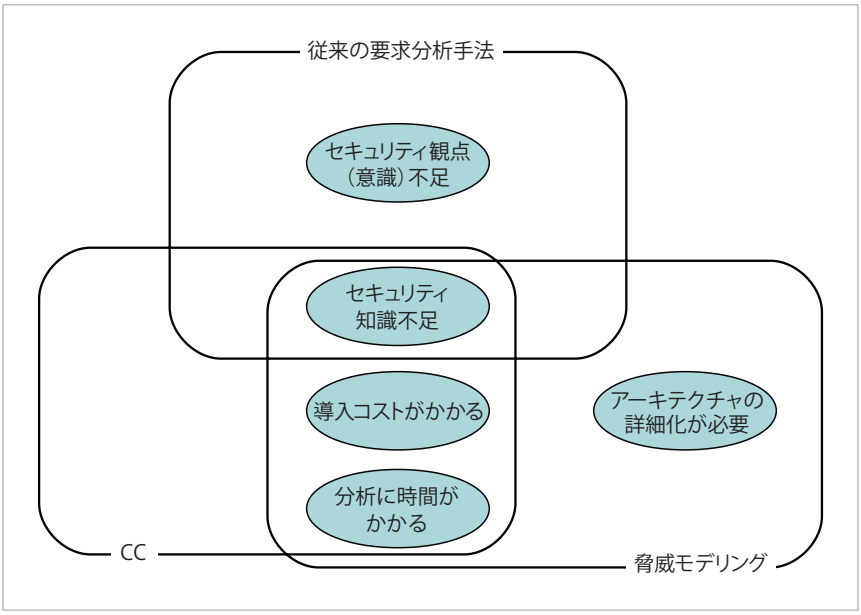

図 -1　各分析手法のセキュリティ要求分析における問題点

しまうことになる．現状においては，セキュリティ知識，ノウハウを持つ人材は，開発者においても，他のステークホルダにおいても一部にとどまっており，社会全体としては十分な数とは言えない．

(3) 時間，コスト

たとえセキュリティ知識を持つ人材が揃っていたとしても，セキュリティ要求分析に必要な保護資産の識別，脅威の識別，リスク分析を行うには相応の時間と，人的資源（= コスト）を要する．しかし，特に SI の現場では，この「カネと時間」が大きな足枷になってしまう．第一に，SI では提案段階や入札時にソフトウェア全体の価格がほぼ決まってしまう．競争入札が行われた際に，1 社だけがセキュリティの分を上乗せした価格を提示したらどうなるか．セキュリティ意識の低い顧客相手では，確実に負けてしまうのではないだろうか．だとしたら，最初からセキュリティを考慮しない価格で提案した方が有利ということになり，ますますセキュリティは軽視されることになってしまう．また，SI では提案までの期間は非常に短いことが多く，その間に十分な脅威分析のための時間を確保することは困難になっている．

(4) 開発手法，ツール

これまで挙げたように観点の相違，セキュリティに特化した人員配置や時間，予算の確保等が別途必要になるなど，セキュアなソフトウェア開発を効率的に実現するためには，従来のものとは異なる開発手法やセキュリティ分析のための道具が必要になる．しかし，問題を根本的に解決する，業界標準として用いられている開発手法や道具は現在のところはまだない．ゴール指向分析や SQUARE など，いくつかの手法が提案されてはいるが，日本の企業において導入され，効果をあげているものは，現状ではほとんどないのではないだろうか．こういった新しい手法の導入には教育のための初期コストが必要になることや，日本での実績があまりないことなどが，導入の障害になっていると考えられる．CC は ISO/IEC 15408 として標準化されており，開発標準としての立場に最も近いように見える．CC の詳細については他記事にあるので参照いただきたいが，筆者は，CC は認証を目指す目的以外で用いる場合，多くの企業にとってはその導入に必要なコストが高すぎるのではないかと考える．CC は，対象ソフトウェアにあまり知識のない第三者にも評価できるように，多くの証拠資料（ドキュメント）を用意する必要があり，その大部分は従来の開発仕様書等とは別に用意する必要がある．システム全体に対して評価しようとすると，この量が膨大になってしまうため，認証を行う場合でも通常はその評価対象を TOE（Target Of Evaluation）という形で限定する．すると今度は，TOE をどのように限定するか，という問題が生じる．TOE 内は安全と評価されたとしても，TOE 以外の部分にもし脆弱性があったら，システム全体は結局安全ではなくなってしまう．

他に脅威分析の道具としてよく用いられる道具には，脅威モデリング☆1 がある．以降では，この脅威モデリングの概要を紹介した上で，脅威モデリングが企業における要求分析に使えるかどうかを検証する．

各分析手法のセキュリティ要求分析における問題点を**図 -1** に示す．

☆1 threat modeling の訳．文献 1) の翻訳である文献 2) では，邦題は「脅威モデル」となっているが，手法としては「モデリング」「モデル化」と呼ぶべきものであるため，ここでは「脅威モデリング」と呼ぶ．

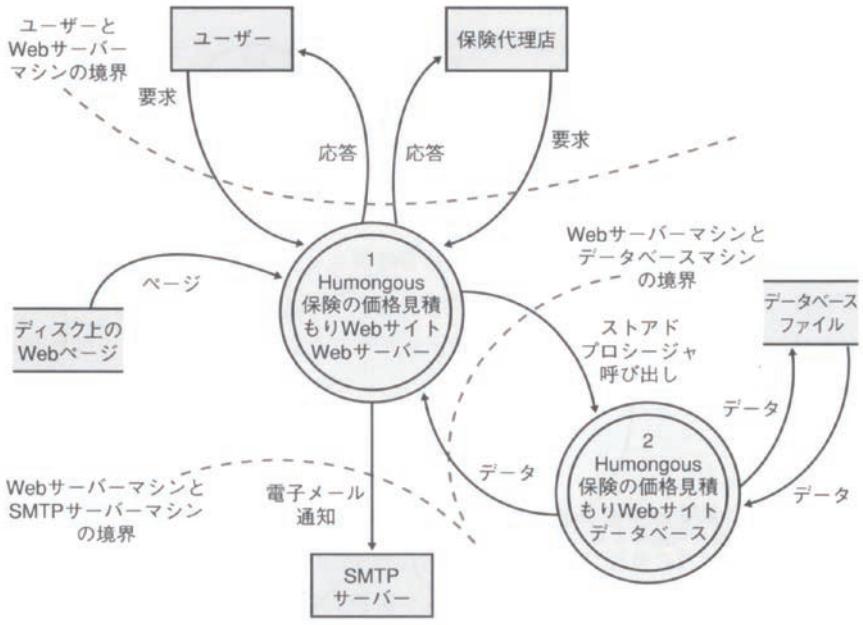

図 -2　DFD の例☆2

脅威モデリング

脅威モデリングは当初，マイクロソフトの Michael Howard，David LeBlanc らによってセキュアなアプリケーションプログラミングの手法の 1 つとして提案された[3]．現在ではマイクロソフトのセキュリティ開発ライフサイクル「Microsoft SDL」に統合されている[4]．

脅威モデリングのプロセスは次の 3 つのステップからなる．

(1) 攻撃者の視点を理解する

システムの中で，どのような個所や要素が攻撃の対象になるかを理解する．これは脅威を識別するためのノウハウといっていい．具体的には，データやコントロールの中継点（「エントリポイント」と呼ぶ）や，保護資産，信頼レベルなどが相当する．信頼レベルとは，通信などに関して相手を信頼してよいかを判断するレベルであり，通常は認証や権限のレベルと同じものを指す．たとえば，同じ機密情報を共有してよい範囲は同一の信頼レベルにあると言える．

(2) システムセキュリティの特徴を知る

システムのモデル化などを行って，脅威識別に必要な情報を収集する．

(3) 脅威を判定する

収集した情報をもとに脅威を識別し，リスクとして評価する．

☆2 図 -2，図 -3 の出典は文献 2)．

次に，上記プロセスの中で脅威モデリングの特徴となるいくつかの手法について解説する．

◆データフローダイアグラム（DFD）を用いたシステムのモデル化

セキュリティの特徴を知るためのモデル化には DFD を用いる（**図 -2** 参照）．図 -2 中の丸はソフトウェアのプロセス，水平に引かれた 2 本の線はデータストア，四角は外部エンティティを表す．外部エンティティとは，システムの外部にあって，システムに入力を行う人や他システムを指す．プロセス，データストア，外部エンティティを矢印で結んだ線がデータの流れを示している．また，点線はそれぞれの物理的，あるいは信頼の境界を表す．

抽出されたこれらの要素はすべてセキュリティや脅威に関連した特徴を持っており，脅威の識別に役立つ．たとえば，ほとんどのセキュリティ攻撃はシステムの外部からの作用である．DFD を用いてその攻撃のポイント（外部エンティティ）を明確にすることにより，具体的に脅威をイメージすることができるようになる．

DFD は通常，図 -2 に示したコンテキスト図から始まって，レベル 0，1，…のようにサブシステムごとに詳細に分解した図を記述していく．この分解作業により，より精度の高い脅威分析が可能になる．

◆脅威の分類“STRIDE”

STRIDE（ストライド）は脅威を検討する際に用いる次の 6 つの分類の頭文字を示したものである．

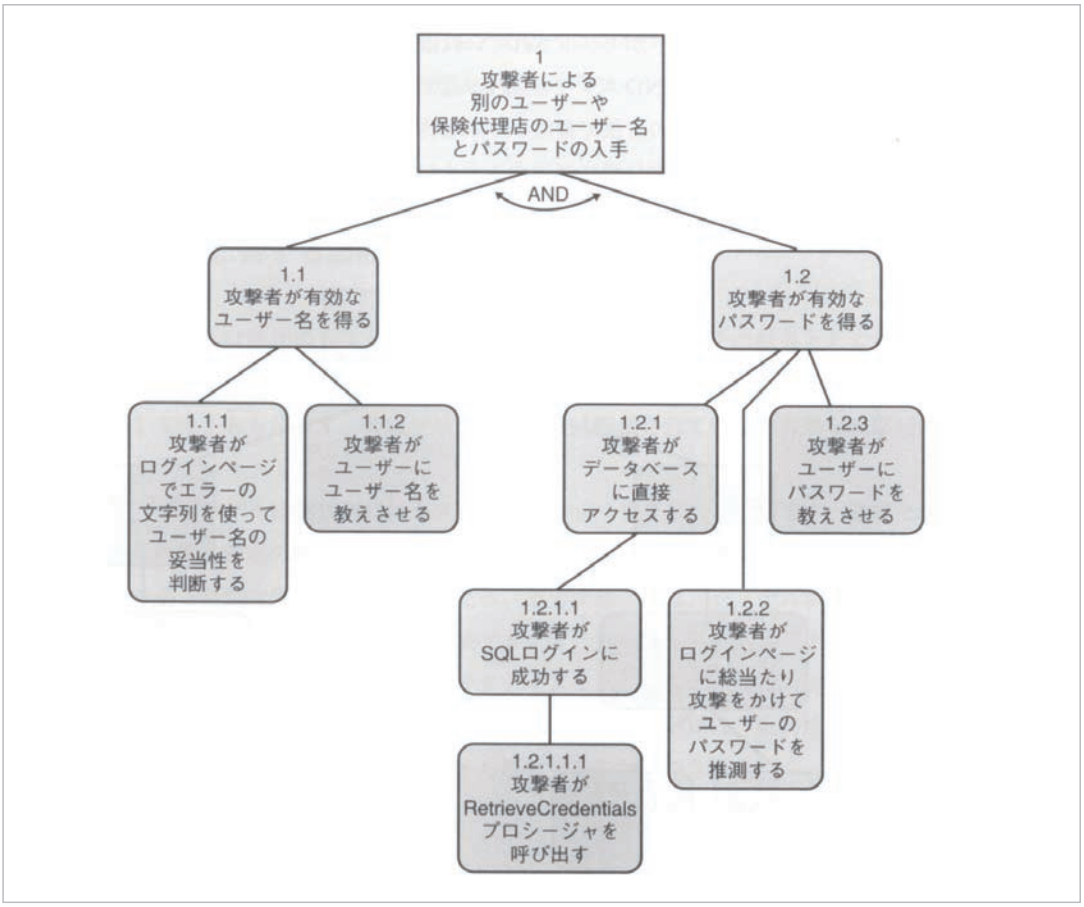

図-3　脅威ツリーの例[☆2]

- Spoofing identity（なりすまし）
- Tampering with data（データの改竄）
- Repudiatoin（否認）
- Informatoin disclosure（情報の漏洩）
- Denial of Service（DoS 攻撃）
- Elevation of privilege（権限昇格）

DFD でモデル化したシステムの各部分に対して，「なりすましは可能か」「データが改竄されると困るか」など，STRIDE を順にあてはめることで脅威の検討，識別を行うことができる．このように分類を行った後，対応するエントリポイント，関連する資産，軽減策の情報を収集する．

◆脅威の評価―脅威ツリーとDREAD

脅威ツリーは，識別された脅威に対して，その可能性を検討するために行う．脅威ツリーの例を**図-3** に示す．脅威ツリーは，識別された脅威を頂点とし，その脅威を可能にする脆弱性や攻撃をその下位に順に記述する．できあがったツリーは，「こういう状況下で，こんな攻撃をすれば脅威が実現する」という可能性を示している．

DREAD（ドレッド）は，ある脅威が資産に対してどれだけの影響を与えるかを示す指標である．STRIDE と同じく，次の 5 つの指標の頭文字からなる．

- Damage potential（潜在的な損害）
  損害の程度
- Reproductivity（再現可能性）
  攻撃の頻度
- Exploitability（攻撃利用可能性）
  攻撃に必要な労力
- Affected users（影響ユーザ）
  攻撃により影響を受けるシステムの比率
- Discoverability（発見可能性）
  脆弱性を攻撃者に発見されてしまう可能性

脅威を上記それぞれの指標で数値によるランク付けを行い，脅威のリスクを定量的に評価する．

脅威モデリングは企業のセキュリティ要求分析に使えるか？

脅威モデリングは，通常の利用という視点ではなく，脅威ツリーによる手段の分析，STRIDE の分類，DFD による攻撃対象の特定など，攻撃の観点に立った脅威分析を可能とする．事実，既存ソフトウェアの脅威分析などではよく用いられている．

一方，他の多くのセキュリティ要求分析手法においても，セキュリティ要求を決定するためには，脅威を識別，

評価する手法が必要とされている．どちらも「脅威分析」を行うのだから，脅威モデリングを脅威分析に使うというのは自然な発想ではないだろうか（筆者も最初はそのように考えた）．では，要求分析の道具としての，脅威モデリングの実効性はどうなのだろうか．

脅威モデリングはそもそも，脅威分析設計向けのプロセスとして定義されている．事実その性質（DFD，脅威ツリーにおいてアプリケーション仕様の詳細化が前提として必要）も，設計段階における分析に適していることを示している．

脅威モデリングを要求分析に用いようとした場合，次の2点がネックになると考えられる．

第1の点は，上述のようにアプリケーション仕様の詳細化が脅威分析の精度を左右するとすれば，仕様のまだ不明確な要求分析段階では脅威モデリングが十分に効果を発揮できない可能性が高い点である．逆に，設計を経て仕様が詳細化してから脅威分析を行い，そこで明確になった脅威をもとにセキュリティ要求を決定するのでは，「要求分析」としては遅すぎ，開発の手戻りを招いてしまいかねない．

第2の点は，脅威モデリングが「攻撃者の視点」寄りであるがゆえに，「開発者から見た脅威」という観点の分析には不向きである点である．脅威モデリングを用い，ある資産に対する脅威の起きる可能性を見つけることはできても，その脅威がソフトウェアにとってどれだけ重大であるかは，脅威モデル自体は判断できない．そのため，DREADなどの評価を使って人間が入力してやる必要がある．したがって，脅威モデリングにとって要求は前提条件であって，セキュリティ要求の定義そのものを脅威モデリングが担うことは困難であると言える．

その他，脅威モデリングを企業で実践するためには，次に挙げるような前提条件が必要になると考えられる．

(1) ステークホルダによるセキュリティポリシー，ないし要求が存在すること
(2) 開発担当者全員に対する，セキュリティ知識の教育およびセキュリティ知識を持つ人材の（一定数）確保が可能であること
(3) 上記を含む，通常のソフトウェア開発に対しての費用，開発期間の上積みが可能であること
(4) 手戻りを許容する開発体制が存在すること

上記の4点は，実は，前述の「企業におけるセキュリティ分析技術の実効性」の章で，現状の問題点として挙げた4項目にそれぞれ該当する．すなわち，多くの企業は脅威モデリングをセキュリティ分析で用いるための前提条件を満たせていないことになる．この手法がなぜ，マイクロソフトでは成功しているのかというと，マイクロソフトは上記4点を前提にできる，恵まれた環境にあるからである．たとえば，度重なるWindowsのセキュリティ問題は逆に，マイクロソフト製品に対して外部からのセキュリティ要求を与えることになっている．また，自社内で製品を作る場合はSIのような場合に比べ，製品にかけるコストや納期が比較的柔軟にできるという利点もある．

しかし，そうでない多くの企業にとっての選択肢は2つ．時間とコスト，労力をかけてセキュリティ教育を行い，かつ開発体制を変革する．それに加え，SI企業は自社だけでなく，顧客のセキュリティ意識，知識を高める努力もしなければならない．これらの活動はもちろん，セキュリティを向上させる上で長期的には必須の作業になるのだが，一方で，前述の4点を前提にしなくても，ある程度の効果をあげることのできるセキュリティ要求分析手法も，短期的視点では必要である．もし，少ないセキュリティ資源で，短時間にセキュリティ脅威を抽出し，リスク，コストを明確化して他のステークホルダに提示することができれば，ステークホルダのセキュリティ意識を喚起することができ，的確なセキュリティ要求の定義に向け，前進が期待できる．最近では，この点に着目した手法の研究も行われつつある☆3．SI企業としては，現状に即した即効的要求分析手法／ツールの登場を期待したい．

参考文献
1) Swiderski, F. and Snyder, W. : Threat Modeling, Microsoft Press (2004).
2) Swiderski, F. and Snyder, W. : 脅威モデル―セキュアなアプリケーション構築，日経BPソフトプレス (2005).
3) Howard, M. and LeBlanc, D. : Writing Secure Code Second Edition, Microsoft (2003).
4) Howard, M. and Lipner, S.: The Security Development Lifecycle, Microsoft (2006).
5) Braz, F., Fernandez, E. B. and VanHilst, M. : Eliciting Security Requirements Through Misuse Activities, Accepted for the 2nd Int. Workshop on Secure Systems Methodologies using Patterns (SPattern'07). In conjunction with the 4th International Conference on Trust, Privacy & Security in Digital Busines (TrustBus'07), Turin, Italy (2008).
6) Okubo, T. and Tanaka, H. : Identifying Security Aspects in Early Development Stages, Proc. International Conference on Availability, Reliability and Security (ARES'08), Barcelona, Spain, pp.1148-1155 (2008).

（平成21年2月2日受付）

☆3 文献5）や文献6）の研究では，ユースケースやアクティビティ図にセキュリティ的な拡張を行うことで，セキュリティ要求分析作業の効率化をはかっている．

大久保 隆夫（正会員）　▶okubo@jp.fujitsu.com
昭和41年生．平成3年東京工業大学大学院物理情報工学専攻修士課程修了．同年（株）富士通研究所入社．分散ソフトウェア環境，アプリケーションセキュリティ，セキュアソフトウェア工学の研究に従事．