2. ソフトウェア製品における脆弱性

（株）インターネットイニシアティブ／
有限責任中間法人 JPCERT コーディネーションセンター
歌代 和正　utashiro@iij.ad.jp

有限責任中間法人 JPCERT コーディネーションセンター
鎌田 敬介　office@jpcert.or.jp

バグと脆弱性の違い

ソフトウェアが意図通りに動作しなかったり，想定外の挙動をしてしまうことをバグと呼ぶが，その中でもセキュリティに影響のあるものをセキュリティホールまたは，脆弱性（vulnerability）などと呼んでいる．単なるバグの場合，利用者はその存在について不平は言うだろうが，少なくともそれを回避するために協力的に対応してくれるだろう．ところが，脆弱性になるとそうはいかない．第三者が，利用者自身，あるいは別の誰かに被害を与えるためにそのバグを利用しようとする．そのために，普通では考えられないような使い方をしたり，非常識なデータを与えたりする．通常のデバッグのためにも，想定外の使い方やデータへの対応は必要だが，その想定外の程度と，発見された場合の影響の仕方が，単なるバグとセキュリティにかかわる脆弱性とでは決定的に違うのである．

インターネットの普及に伴い，遠隔地からの攻撃が可能となり，その対象となる機器の数は飛躍的に増大した．メールや Web 閲覧を通じた間接的な攻撃も含めれば，ネットワークに接続するほとんどの機器が対象となる．そのため，脆弱性に対応することの重要度はますます増大している．もちろん，脆弱性を作らないことが一番であるが，それが不可能だとすれば，早期に発見し，対策を準備し，情報を流通して対策の実施を図ることが求められる．しかもハードウェアとソフトウェアの共通化は世界規模で進んでいるから，国境を越えた 24 時間体制での対応が必要なのである．本稿では，JPCERT/CC が関連組織と協力しながら進めている脆弱性への対応について解説する．

国内のソフトウェア製品における脆弱性の取り扱い

日本国内の脆弱性情報の取り扱いは，経済産業省告示「ソフトウエア等脆弱性関連情報取扱基準」に従って行われている．この中で，JPCERT/CC は「調整機関」としてソフトウェア等製品開発者（ベンダ）との調整を行う役割を担っている．調整は以下のような流れで行われる（図 -1）．

1 報告を受けた脆弱性関連情報について，連絡する必要のあるベンダを選出する．
2 選出したベンダに対して「概要情報」を送付する．
3 ベンダは該当製品の有無を調査し，該当する製品があると判断すると「詳細情報」を要求する．
4 JPCERT/CC から「詳細情報」を送付する．その際には，暗号化メールなどを用いる．
5 ベンダは自社内の製品について調査し，製品への対応方針を決定する．その後，対策方法（回避策やパッチ等）の策定スケジュールを JPCERT/CC に連絡する．
6 JPCERT/CC は，各ベンダの策定スケジュールを調整し，脆弱性関連情報の公表日を策定する．
7 各ベンダは公表日に合わせて情報を公表し，同時に JPCERT/CC は JP Vendor Status Notes（JVN）にて脆弱性情報を公表する．

JPCERT/CC は，脆弱性情報を受け付ける窓口の担当者をベンダリストとして管理し，そのリストに基づいて脆弱性情報を連絡するベンダを選出する．脆弱性情報を送付するべきベンダが登録されていない場合，リストへの登録を促すことから始まる．一筋縄ではいかないベンダも多く，登録までに数週間かかるケースも珍しくない．

各ベンダには「テクノロジーキーワード」と呼ばれる

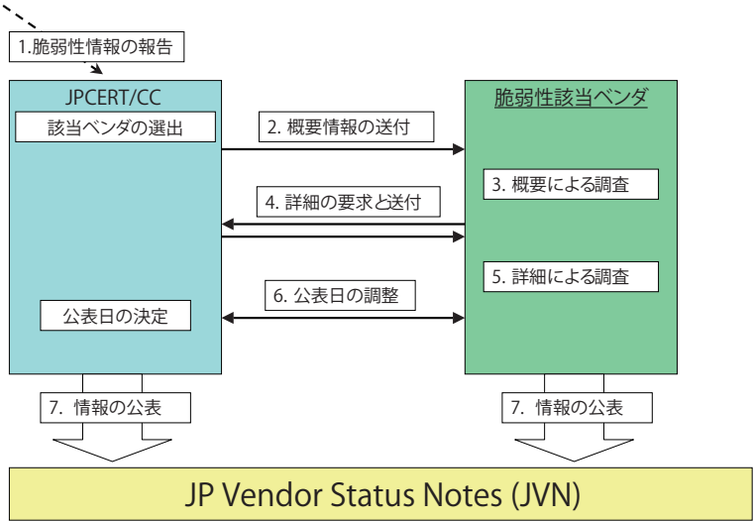

図 -1　国内のソフトウェア製品における脆弱性の取り扱い

技術用語からなるキーワードリストの登録を依頼している．JPCERT/CC がベンダを選出する際には，このキーワードリストを参考にする．

上記 2 の「概要情報」とは，脆弱性情報の中から，実際の攻撃方法につながるような情報を排除したものである．「詳細情報」には，脆弱性の再現手順などが含まれる．情報を概要と詳細に分けているのは，必要以上のベンダに対して情報を提供しないことで，情報漏洩の危険性を低減するためである．

ベンダの対応方針の決定後，脆弱性関連情報の公表日を決定する．ベンダは公表日までに，公表する情報の準備を行う．公開される情報には以下のような項目が含まれることが望ましい．

- 脆弱性に該当する製品名，バージョン
- 脆弱性の概要
- 脆弱性の対策方法（回避策やパッチ等）
- 関連情報へのリンク
- 発見者への謝辞（発見者が望んだ場合）

公開する情報には，実際の攻撃につながるような詳細な情報は記載しない．ソースコードレベルでのパッチの場合には，その情報から攻撃方法が推測できることもあるが，一般に公開される情報には，実際の攻撃手順や，攻撃コードなどを掲載すべきではない．安易にそのような情報を公開することでワームやウイルスの発生を許してしまう恐れがあるからである．

ベンダの情報の公表と同時に，JPCERT/CC と IPA が共同で運営している脆弱性対策情報ポータルサイトである JVN からも情報を公開する．ここには，詳細情報を受け取り，調査を行ったベンダの名前がその状況とともに掲載される．

意外と難しいベンダの特定

ベンダとの脆弱性情報のやりとりを一般化したかたちで紹介したが，実際には円滑に調整が進むことは珍しい．たとえば，あるベンダ A 社の特定製品に関する脆弱性情報を扱う場合，表向きには A 社が販売しているため，A 社にコンタクトをすれば製品の修正が可能であろうと推測する．

しかし，A 社にコンタクトをしてみると，実際に開発を行っているのは子会社や取引先のベンダ B 社であり，脆弱性の修正は開発をしている B 社でなければ行えない．しかし，B 社製品として公表しても一般のユーザには A 社の製品が影響を受けることは推測しにくく，公開情報としては A 社の名前で公表されるのが適切である．

このような関係は，親会社・子会社の関係や，グループ企業，取引関係など多岐に渡り，OEM やオフショア開発の普及がそれに拍車をかけている．

多様なソフトウェアの開発形態

脆弱性の対象となるソフトウェアは製品ばかりではない．オープンソースコミュニティによって開発されているものや，個人の開発者によって提供されているフリーソフトウェア，シェアウェアなども存在する．

これらのソフトウェアに関する脆弱性情報の扱いは難しく，開発者が誰であるかすら分からないこともある．開発者へのコンタクト方法としてメールアドレスのみが公開されている場合には，個人や法人を特定せずにメールを送ってみるしか方法がない．コンタクト方法がまったく分からず，開発者のコンタクト先を知っていそうな

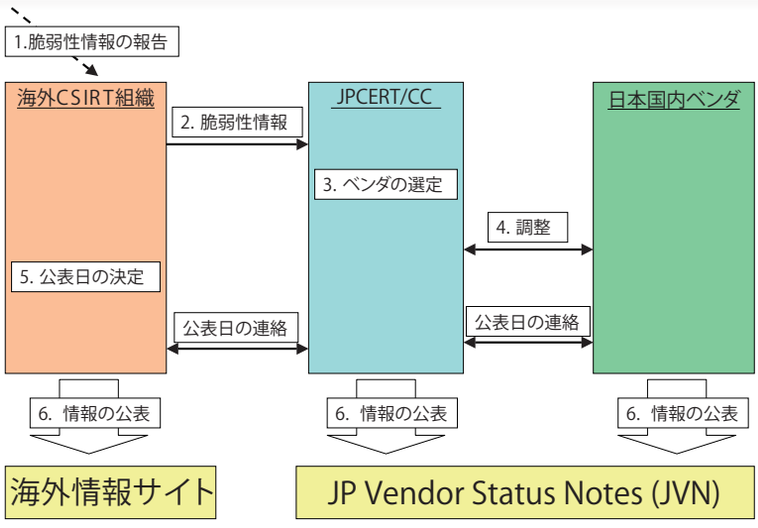

図 -2　海外組織との連携による脆弱性の取り扱い

コミュニティや会社などに相談することもある．

この種のソフトウェアの場合，サポート担当者がいることは稀であり，取り扱いの仕組みや，調整の必要性についての説明に難航することも少なくない．個人による開発の場合は，旅行や長期出張などで対応が滞ったり，病気療養中で対応できないといような状況も考えられる．

海外組織との連携による脆弱性の取り扱い

JPCERT/CC では，日本国内から報告される脆弱性情報のみでなく，海外の組織とも協力関係を構築し，受け取った脆弱性情報を日本国内のベンダに展開している．具体的には以下のような流れである（図 -2）．

1 海外組織へ脆弱性情報が報告される．
2 海外組織が日本への情報の展開が必要だと判断すると，JPCERT/CC に連絡する．
3 JPCERT/CC では既存のベンダリストを活用し，情報を提供するベンダを選定する．
4 JPCERT/CC は日本ベンダとの調整を行い，海外組織は海外ベンダとの調整を行う．
5 JPCERT/CC は日本国内のベンダの状況を伝え，海外組織が脆弱性情報の公表日を設定する．
6 公表日に合わせて，海外組織，JVN，日本ベンダから一斉に情報を公開する．

複数の CSIRT が関係して国際的に脆弱性情報を扱う場合，各組織で調整を行うベンダを明確に分けておく必要がある．JPCERT/CC であれば主に日本国内のベンダが対象となるが，CERT/CC では米国企業をはじめとした国際ベンダやオープンソースコミュニティを対象とし，英国の NISCC では特に国を指定することなく，さまざまな開発者とのやりとりを行っている．

ここで問題になるのが，国際的に製品を販売しているベンダである．たとえば，Windows という OS は米国のマイクロソフト社を本社とする会社の製品であるが，マイクロソフト社には日本法人も存在する．JPCERT/CC がマイクロソフト社にコンタクトをとりたい場合には，CERT/CC 経由で米国にコンタクトすべきか，日本法人にコンタクトすべきかが問題になる．

脆弱性情報を受け付ける窓口は開発本国に設置されている場合が多いが，ベンダによっては日本法人で受け取っても本国法人で受け取っても内部的に同一処理が行われる体制になっている場合もある．マイクロソフト社の場合には，日本法人においても脆弱性情報の報告窓口が用意されており，そのような体制が構築されている．

日本国内のみで販売されている製品なのだが，実際の開発は海外のベンダが行っているという場合もある．製品を修正することができるのが海外ベンダのみであれば，情報を開示せざるを得ないが，ベンダ登録等の問題もあり，状況に応じた臨機応変な対応が必要とされる．

脆弱性情報の海外への展開

JPCERT/CC で扱った脆弱性情報の中には，「JVN#67B82FA3 SSL-VPN 製品における Cookie の脆弱性」のように，日本で発見され，海外のベンダに展開された事例もある．

この脆弱性は，SSL-VPN 製品のうちで Web インタフェースによるログインを許可するものの中に，セッション管理で使う Cookie にセキュア属性が付与されていないため，セッションハイジャックをされてしまう可能性がある，というものである．報告に指摘されていた製品の中には，米国製品も含まれていた．

JPCERT/CC は米国ベンダへのコンタクト情報を持っ

ていないので，CERT/CC へ協力を依頼し，脆弱性の詳細な情報や，連絡を希望するベンダのリストなどを送る．CERT/CC は連絡が必要であると判断したベンダへ情報を展開し，JPCERT/CC にその旨を通知してくる．その後，JPCERT/CC は日本国内ベンダの状況などを考慮して設定した公開日を国内ベンダや CERT/CC に連絡する．米国ベンダへ公開日を伝えるのは CERT/CC である．

ベンダからの公開日変更の依頼などがなければ，当初設定された脆弱性の公表日に JVN，Vulnerability Notes，各ベンダから情報が公開される．この際，公開日の設定には時刻とタイムゾーンを指定するが，米国と日本の時差の関係上，どちらかにとって不都合な時間帯になってしまうことが多い．そのような場合には，脆弱性の影響の大きい国を優先する．

脆弱性情報の連携

米国 CERT/CC が扱った脆弱性情報は Vulnerability Notes Database として一般に公開されている．それには以下のような項目が含まれている．

- ID（識別番号）
- 名称
- 概要情報
- 内容説明
- 影響度
- 解決策
- 影響を受けるシステム
- その他，参考情報，謝辞，公開日時等

Vulnerability Notes Database の項目にあるように，公開する脆弱性情報には識別番号を割り当てるのが一般的だが，本質的に同じ内容を指す脆弱性に対して，複数のサイトが異なる識別番号を付すことがある．

さまざまなサイトや組織において同じ脆弱性に対して複数の番号を割り当てることは混乱を招く．このような問題を解決するために，各脆弱性情報に対して統一した名前（識別番号）を割り当てようとしている活動が Common Vulnerabilities and Exposures（CVE）である．

CVE では，複数個所で公開されている脆弱性情報をまとめる辞書のようなものを提供しようとしている．具体例をあげると，CVE 名 CVE-2004-0171 として公開されている脆弱性情報には，以下の複数の識別番号が割り当てられている（表 -1）．

この脆弱性は，FreeBSD において TCP セグメントの再構築を行うキューの制限を行う処理に脆弱性があり，攻撃者がシステムのメモリを使い果たし，DoS 状態を引き起こすことが可能になるというものである．これに対して，APPLE や FreeBSD をはじめとする複数の組織が文書を公開し番号を割り当てており，CVE 番号によってこれらが 1 つにまとめられている．

IDEFENSE	20040302 FreeBSD Memory Buffer Exhaustion Denial of Service Vulnerability
APPLE	APPLE-SA-2004-05-28
FREEBSD	FreeBSD-SA-04:04
CERT-VN	VU#395670
BID	9792
X-Force	freebsd-mbuf-dos（15369）
OSVDB	4124

表 -1　CVE-2004-0171 に割り当てられているさまざまな識別番号

脆弱性の分類

脆弱性情報の分類について，脆弱性情報を受け取るベンダの立場に立って，脆弱性を次のポイントで分類してみる．

- 仕様の脆弱性
- 実装の脆弱性

仕様の脆弱性とは，文字通りソフトウェアの仕様が原因で発生する脆弱性であり，実装の脆弱性とはソフトウェアの実装内容（ソースコード）によって発生する脆弱性である．以下，それぞれ詳細を述べる（図 -3）．

◆仕様の脆弱性の例

仕様の脆弱性については，2 つのパターンが考えられる．

1. 製品固有の仕様の脆弱性
2. 公開技術仕様の脆弱性

製品固有の仕様の脆弱性とは，その製品だけに影響のある脆弱性である．公開されている技術仕様の脆弱性の代表的な例は，プロトコルの脆弱性である．たとえば NISCC-236929 として公開されている TCP の脆弱性は TCP の仕様レベルでの脆弱性であるため，RFC に従って実装をしていれば脆弱性が存在してしまう．

また，仕様そのものには脆弱性が存在しなくても，共通の仕様に基づいて実装された複数のソフトウェアで確認される共通の脆弱性が存在する場合もある．NISCC-380375 の「MIME に関する複数の脆弱性」は，MIME の仕様ではなく実装上の脆弱性である．しかし，

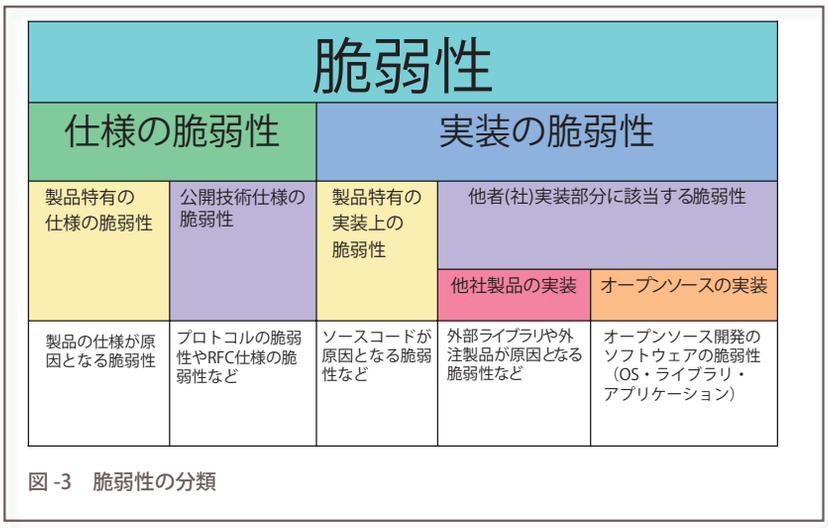

図 -3　脆弱性の分類

MIME の仕様を元に独自に実装された複数のソフトウェアに共通する脆弱性が発生しているので，仕様に由来すると考えることもできる．

◆実装の脆弱性の例

実装の脆弱性についても，2 つのパターンが考えられる．

1．製品固有の実装個所に該当する脆弱性
2．製品の開発者が実装していない個所に該当する脆弱性

製品固有の実装個所に該当する脆弱性とは，仕様の脆弱性の場合と同じように，その製品のみが持つ実装部分に存在する脆弱性である．

これに対して，製品の開発者が直接実装していない個所に脆弱性が存在することがある．この場合，以下の 2 つのケースで違いがある．

外注先やライブラリ製品など他社が開発したソースコードを利用している

製品を販売している会社が実際に脆弱性を修正できるのかどうかは当事者以外には分からない場合もあるし，実際に外部に発注している場合などもあることは先に述べた通りである．

オープンソースソフトウェアなど，公開されているソースコードを利用している

ソースが公開されている OS や，ライブラリ（libtiff, libpng など），アプリケーションに脆弱性が発見された場合は，それらのソースコードを利用している製品も影響を受ける．オープンソースのコードを流用している製品では，製品開発ベンダが直接修正する場合もあるが，開発元のコミュニティからパッチが公開されるまで修正されないケースもある．

脆弱性情報の実例

ここで，JPCERT/CC が実際に扱った脆弱性情報の中で，取り扱い中に起きた事例などをいくつか紹介しよう．

◆開発者からの情報提供

脆弱性情報「JVN#8BAAAB4E: msearch におけるディレクトリトラバーサルの脆弱性」では，報告を受けた段階では，msearch という特定されたソフトウェアの脆弱性情報であると思われたため，コンタクト先は msearch の作者のみとした．しかし，msearch 作者からの指摘により，この脆弱性に該当する可能性のある他のソフトウェアが存在することが分かったため，別の作者へもコンタクトして対応した．

当初は特定ソフトウェアのみの脆弱性として取り扱いを始めるが，開発者からの情報提供によって複数のソフトウェアに影響する脆弱性であることが分かることも多い．

◆脆弱性情報の漏洩

脆弱性情報「JVN#1BF8D7AA: LDAP サーバの更新機能におけるバッファオーバーフローの脆弱性」は，一部の LDAP サーバにバッファオーバーフローの脆弱性が存在するという内容であった．脆弱性の対象となったのは，米国ミシガン大学で公開されていた LDAP サーバの実装 UMich LDAP のある時点のコードを元に実装されたソフトウェアである．

JPCERT/CC では，国内の登録ベンダおよび CERT/CC の協力で米国のベンダへの情報展開を行った（Vulnerability Notes VU#258905）．ところが，米国のあ

るベンダが公開日を待たずに情報を公開してしまったのである．そのため，設定された公開日を早めざるを得なかった．情報の漏洩はいつどこで起こるか分からず，厳しく管理されているはずの情報が情報の管理者ではなくシステム管理者の手によって漏洩することもある．機密情報を社内に展開する際には，絶対に漏洩しないという過信は禁物である．

公開日を待たずに情報を公開してしまったベンダは，ペナルティとして脆弱性のハンドリング対象から外されることもある．

◆定期的パッチ公開の弊害

ある脆弱性情報の扱いの中で，あるベンダA社のパッチスケジュール化が問題になったことがある．A社では，毎月特定日にパッチの公開を行うことになっていたが，社内の調整ミスにより，脆弱性情報の公表日が決定されていないにもかかわらず，翌月の修正項目に含めてしまった．

しかし，同じ脆弱性情報に関係した会社の中には，A社のパッチ公開日までには対応できないところもあった．そこで，公開内容に脆弱性に関する情報は一切記載しないという条件でA社については先行してのパッチの公開を許可し，脆弱性情報の公開日はA社のパッチ公開の翌月に設定した．

このケースでは，定期的なパッチ公開体制が逆に問題となってしまった．脆弱性にかかわる複数のベンダがパッチをスケジュール化している場合，情報の公開日時はどちらか一方のパッチ公開日にしか合わせられない．今後，より深刻な問題となる可能性もある．

◆開発ベンダの認識不足

ある脆弱性情報で，詳細情報を受け取ったベンダから修正が完了したとの連絡を受け，公開に至った．しかし，脆弱性情報の公開後，発見者から脆弱性が修正されていない旨の連絡が入った．

当初発見者から報告された方法では脆弱性は再現しなかったが，少しやり方を変えるだけで同種の脆弱性が再現されてしまうのであった．製品開発ベンダの脆弱性に対する認識が浅く，本質的な解決にはなっていなかったケースである．

◆発見者と開発ベンダの意識の違い

脆弱性の発見者と該当製品の開発ベンダとの意識の違いは大きい．発見者は，事実を誇張して表現しがちである．報告した脆弱性がいかに驚異的なものであるか，甚大な被害が予想されるかを言及したがる．一方，脆弱性情報を受け取ったベンダは，大きく取り上げられることで企業イメージの低下につながることを恐れたり，製品のユーザに脆弱性の存在を報告することがデメリットになると考えることもある．概して，発見者から公開される情報は大げさに，ベンダから公開される情報は控えめな内容になりがちなのである．

ある脆弱性情報では，発見者は大きな被害になり得ると報告してきたのに対し，ベンダ側は「イントラネット内で使用するソフトウェアなので脆弱性の脅威は低い」と公表した．その公表内容を見た発見者から，ベンダの公表内容について指摘があり，ベンダ側ではそれを受けて公表情報を修正した．

現時点では，報告者は脆弱性に関する専門家であることが多い．ベンダは脆弱性の対応について経験が浅いところも多く，意識のずれによる影響はさらに拡大する．

今後の課題

もちろん脆弱性など存在しないに越したことはないし，バグや脆弱性を発生させないための技術開発も行われている．しかし，ソフトウェアが大規模化，複雑化していく状況では，完全になくすことは不可能であろう．円滑な事後対応の重要性はますます高まりつつある．

脆弱性については，開発者側に非難が集中する傾向があるが，限られた経費と期限の中で懸命に努力しているベンダもあることは評価すべきである．いたずらに高機能低価格化を求める利用者にも責任の一端はある．その一方で，脆弱性対応の重要性を理解せずに開発を行っているベンダがあることも残念ながら事実である．「嫌なら使うな」という論理が通用するはずはない．

開発者，利用者，報告者のそれぞれが互いの立場を理解し尊重して，協力しあうことが重要であろう．そのために必要な調整作業を充実していくことが，我々の目標であり責任でもある．本稿でもいくつか挙げたように，脆弱性取り扱いの仕組みには，まだ問題点や課題も数多く存在する．しかし，完璧な仕組みではないからとそのままにしておいては，問題は拡大するばかりである．現時点でできることを実行しながら，着実に状況を改善する努力を怠ってはならない．

参考情報

1）JPCERT コーディネーションセンター：http://www.jpcert.or.jp/
2）JP Vendor Status Notes JVN：http://jvn.jp/
3）経済産業省－脆弱性関連情報取扱体制：
http://www.meti.go.jp/policy/netsecurity/vulhandlingG.html
4）US-CERT Vulnerability Notes Database：
http://www.kb.cert.org/vuls/
5）Open Source Vulnerability Database：http://www.osvdb.org/
6）X-Force Database：http://xforce.iss.net/xforce/search.php
7）SecurityFocus HOME Vulns Archive：
http://www.securityfocus.com/bid/

（平成17年4月18日受付）