

NRI OSSソリューションマガジン

Annual Summary

2012-2013

～OSSの動向がわかる羅針盤～

当資料は、毎月1回配信されているメールマガジン「NRI OpenStandiaニュース」のうち、2012年3月から2013年2月の間に配信されたものに掲載されているコンテンツを抜粋し、サマリーとして編集したものです。

1 10年後(2023年)の情報サービス業界

2 イベント/セミナーサマリー

3 OSSコンプライアンス紹介

4 バグ&セキュリティ情報

5 オープンソースまるごとコンサルティング

6 編集後記サマリー

1. 10年後(2023年)の情報サービス業界

1. 10年後(2023年)の情報サービス業界

10年後(2023年)の情報サービス業界

企業の事業環境や情報サービス業界が、大きな転換点にさしかかっているということは、皆様それぞれに感じていらっしゃるのだと思います。

もちろん、未来を正確に予測するのは不可能ですが、自分なりの考えを持つことは重要だと思いますので、10年後(2023年)の情報サービス業界の姿について、思いをめぐらせてみたいと思います。

なお、この「予測」は私個人の意見であり、私が所属する会社、組織、団体のものではないことをご了承頂きたく、お願い致します。

■ ユーザ企業の変化

・経営にとって、ITはより重要に

全世界で50億人がネットにつながる時代において、ITは経営にとって、さらに重要な要素になる。

グローバル化が進むが、画一的な市場にはならない。様々な国、文化、世代、価値観、所得層が混在する中で、消費者のニーズは今以上に多様化する。事業環境の変化は今以上に激しく、これらに柔軟に対応する必要がある。企業にとってITは、このような多様化するニーズや、激しく変化する事業環境に対 するためのツールとして重要であると同時に、ITそのものにも高度な柔軟性が求められる。

・プロジェクトは小規模化

多様化するニーズに対応するため、IT関連のプロジェクトはより現場に密着した組織が主導する。また、その規模は今よりも(桁が異なるくらい)小規模になる。しかし、プロジェクトの数は多くなる。クラウドサービスの普及により、インフラへの大規模投資が不要となったことも、プロジェクトの小規模化を加速させる。

・継続的な改善

事業環境の変化は早く、ITに対する改善も今よりも早いサイクル(月単位、週単位)で行われる。当初から完全に要求を満たすシステムを構築するのではなく、いわゆる「スモールスタート」型のシステムが主流となる。システムの寿命は(改善を繰り返すことによって)今よりも長くなる可能性もあるが、「スクラップ&ビルド」的なシステムが増加する可能性もある。

■ エンジニアの変化

・多能工化するエンジニア

特定分野(例えば、コーディング、テスト、ネットワーク、サーバ基盤、等といった分野の専門家)ではなく、一人で全ての役割をこなす「多能工」が主流になる。クラウドサービスの普及により主にインフラ構築が容易になったこと、インターネットの普及により知識の習得が容易になったこと、オープンソースの普及により、様々なツールが無償で入手可能になったことなどにより、エンジニア個人の生産性が飛躍的に向上し、多能工化が実現される。一方、「特定の業務(の実装)に詳しい」、「特定のオープンソースに詳しい(カスタマイズできる)」、「特定のデバイスに詳しい」など、+αの専門性も求められる。

・不足する「新たなITアーキテクト」

プロジェクトの小規模化によって、従来の「顧客対応、プロジェクトマネジメント、ITアーキテクト」の役割を一人でこなす、「新たなITアーキテクト」が必要になる。「新たなITアーキテクト」は数名のエンジニアと共にチームを組織し、小規模なプロジェクトを遂行する。「新たなITアーキテクト」は、単なる手配師ではない。自らもコーディングができ、自分一人でもチームを組織できる。プロジェクトの数が増えるため「新たなITアーキテクト」の要員が不足する。「新たなITアーキテクト」は、ユーザ企業に雇用されているか、ユーザ企業から委託されるがユーザ企業の社員に近い立場で業務を行う。

・グローバルな競争

インターネットの普及や、クラウドサービスの普及により、ユーザ企業は世界中からエンジニアを調達できるようになる。エンジニアはグローバルな競争に巻き込まれる。エンジニアは「新たなITアーキテクト」としてユーザ企業に近い立場で活躍するか、「特定の業務に詳しい」、「特定のオープンソースに詳しい」、「特定のデバイスに詳しい」などにより自分自身を差別化する必要がある。

・ITで社会に貢献

エンジニアの仕事に対するモチベーションとして、(従来言われていた)報酬や新技術の習得よりも、仕事を通じて社会に貢献したいという欲求が強くなる。このような欲求が満たされるかどうかで、仕事を選ぶようになる。業務時間以外で自らの専門性を活かしたボランティア活動(プロボノ)や、オープンソース・コミュニティの活動を行うエンジニアも増える。また、このような活動を行いやすい雇用形態(例えばフリー)を選択することが増える。

1. 10年後(2023年)の情報サービス業界

■ ベンダー/システムインテグレーターの変化

- ・ベンダー/システムインテグレーターの役割は、以下の2つのタイプに分かれる。
 - 1) 超大規模プロジェクトを遂行する組織。
 - 2) 前述の小規模プロジェクトを遂行するチームの集合体。
- ・この2つのタイプは1つの会社内に共存することは難しい。
- ・また、ベンダー/システムインテグレーターに所属しない、フリーのエンジニアが増加する。

■ オープンソースの役割

・ベース+カスタマイズ/アドオン開発

ますます多様化するニーズに対応するため、既存のソフトウェア製品をそのまま利用するのではなく、カスタマイズやアドオン開発を行う形態が増える。よって、オープンソースをベースとして、これにカスタマイズやアドオン開発を行うタイプのシステム開発が増加する。

・標準ツールとしてのオープンソース

継続的な改善の中で、当初開発したシステムは複数のエンジニアに引き継がれながら開発されていく。このようなケースにおいて、特定のベンダーに依存しておらず、世界中で利用されているオープンソースをベースにするメリットは大きい。グローバルなエンジニアの調達があたりまえになる状況においては、なおさらである。

・公共財としてのオープンソース

エンジニアにとって、オープンソースは今以上に必要不可欠なツールとなる。エンジニアは、自分自身の生産性を高めるためにツールの改善を行ったり、オープンソース・コミュニティへの貢献として改善や普及活動を行ったりすることに、さらに時間を割くようになる。

以上、稚拙な内容で申し訳ありませんが、皆様が今後業務を行う上で、少しでも気づきになれば幸いです。

好ましい未来、厳しい未来、ともにありますが、今以上にエンジニアが生き活きと活躍し、今以上にITが企業に貢献できる「明るい未来」を、我々は皆様と一緒に作る事ができると確信しています。また、そのためにオープンソースが重要な役割を果たすことは間違いありません。

情報サービス業界の「明るい未来」とオープンソースの普及、発展のため、2013年もOpenStandiaは活動してまいります。



2. イベント/セミナーサマリー

2. イベント/セミナーサマリー

【直近のセミナー講演資料】

<シングルサインオン、統合認証関連、SSO&IDM>

「統合認証基盤、貴社の要件に最適な導入パターンは？」

http://openstandia.jp/pdf/130124j4_oss_strategy_sso.pdf

<MongoDB>

「MongoDB 2.4 リリース間近！最新情報と事例紹介」

http://openstandia.jp/pdf/130213f3_introseminar_mongodb.pdf

<オープンソースERP Adempiere>

「オープンソースERP(Adempiere/iDempiere)活用セミナー」

http://openstandia.jp/pdf/prop_openstandia_erp.pdf

| 開催日 | 実施イベント/セミナー |
|------------------|---|
| 2012/03/08 | OSS BI(Jaspersoft)のご紹介とJasperReportの導入 & 実践解説セミナー |
| 2012/03/16,17 | オープンソースカンファレンス2012 Tokyo/Spring |
| 2012/04/05 | 第2回 OpenSSO & OpenAMコンソーシアムセミナー |
| 2012/04/19 | オープンソース最新動向セミナー |
| 2012/05/09,10,11 | リード エグジビション ジャパン主催 「第3回 クラウド コンピューティングEXPO 春」 |
| 2012/05/12 | オープンソースカンファレンス2012 Nagoya |
| 2012/06/07 | OSS BI(JASPERSOFT)のご紹介と JasperReportの導入 & 実践解説セミナー |
| 2012/06/14 | オープンソースムーブメントが創りだす新たなビジネスモデルセミナー |
| 2012/06/28 | 【大阪】DELL・Intel・OSSTech・NRI共催「オープンソース技術解説セミナー」 |
| 2012/07/12 | クラウド時代のOSS戦略セミナー ～SSO、ID管理、OSS利用など最新事例紹介～ |
| 2012/07/19 | 【東京】DELL・Intel・OSSTech・NRI共催「オープンソース技術解説セミナー」 |

2. イベント/セミナーサマリー

| 開催日 | 実施イベント/セミナー |
|------------------|---|
| 2012/07/19 | クラウド時代のOSS戦略セミナー(MongoDB、OpenAM) |
| 2012/07/24 | オープンソースERP ADempiere紹介セミナー |
| 2012/07/25 | オープンソースカンファレンス2012 .DB |
| 2012/07/26 | OSS BI(JASPERSOFT)、DB(MongoDB)のご紹介セミナー |
| 2012/08/03,04 | オープンソースカンファレンス2012 Kansai@Kyoto |
| 2012/08/09 | クラウド時代のOSS戦略セミナー ～シングルサインオン(OpenAM)、統合ID管理(LISM)など最新事例紹介 |
| 2012/08/23 | OSS BI(JASPERSOFT)、DB(MongoDB)のご紹介セミナー |
| 2012/08/28 | オープンソースERP ADempiere紹介セミナー |
| 2012/09/06 | クラウド時代のOSS戦略セミナー ～シングルサインオン(OpenAM)、統合ID管理(LISM)など最新事例紹介 |
| 2012/09/07,08 | オープンソースカンファレンス2012 Tokyo/Fall |
| 2012/09/13 | クラウド時代のOSS戦略セミナー(MongoDB、OpenAM) |
| 2012/09/20 | OSS BI(JASPERSOFT)、DB(MongoDB)のご紹介セミナー |
| 2012/09/25 | オープンソースERP ADempiere紹介セミナー |
| 2012/10/04 | クラウド時代を見据えた次世代統合認証基盤セミナー (OpenAM、LISM) |
| 2012/10/10,11,12 | 日経BP社主催 ITpro EXPO 2012にて「NRI Insights 2012」開催 |
| 2012/10/10,11,12 | 日経BP社主催「ITpro EXPO 2012」出展 |

2. イベント/セミナーサマリー

| 開催日 | 実施イベント/セミナー |
|---------------|---|
| 2012/10/18 | OSSでここまでできるビジネスインテリジェンスセミナー(Jaspersoft) |
| 2012/10/18 | MongoDB 2.2 の新機能ご紹介セミナー |
| 2012/10/19 | 第3回 OpenAMコンソーシアムセミナー『OpenAM Summit』 |
| 2012/11/07 | 「クラウド」と「モバイル」はエンタープライズ ID 管理をどう変える？ ～新たな ID/API 連携仕様が目指す、SSO の先の「サービス連携」のかたち |
| 2012/11/29 | MongoDB 2.2 の新機能ご紹介セミナー |
| 2012/12/04 | 「クラウド」と「モバイル」はエンタープライズ ID 管理をどう変える？ ～新たな ID/API 連携仕様が目指す、SSO の先の「サービス連携」のかたち |
| 2012/12/16 | オープンソースカンファレンス2012 .Cloud |
| 2012/12/20 | 統合認証基盤、貴社の要件に最適な導入パターンは？ ～事例から考察する、要件とシステム構成～(OpenAM、LISM) |
| 2012/12/20 | MongoDBを導入する前に検討した7つのこと |
| 2013/01/17 | MongoDBを導入する前に検討した7つのこと |
| 2013/01/24 | 統合認証基盤、貴社の要件に最適な導入パターンは？ ～事例から考察する、要件とシステム構成～(OpenAM、LISM) |
| 2013/01/29 | 第52回 ビジネスICTセミナー 「クラウドやオープンソースのパワーを活かし企業ITを徹底強化！」 |
| 2013/02/13 | MongoDB 2.4 リリース間近！最新情報と事例紹介 |
| 2013/02/15 | PostgreSQLを中心としたオープンソースの業務システムへの活用セミナー |
| 2013/02/21 | 統合認証基盤、貴社の要件に最適な導入パターンは？ ～事例から考察する、要件とシステム構成～(OpenAM、LISM) |
| 2013/02/22,23 | オープンソースカンファレンス2013 Tokyo/Spring |
| 2013/02/28 | オープンソースERP(ADempiere/iDempiere)活用セミナー |

3. OSSコンプライアンス紹介

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.1

オープンソースソフトウェア(OSS)とは、どういうソフトウェアでしょうか？

The Open Source Initiative(OSI)という団体が1998年に発表した「The Open Source Definition」(OSD)では自由に再配布ができる、ソースコードが入手できる、派生物が生成できる等々、10種の定義がなされています。

では、再配布やソースコードの入手、あるいは派生物の生成等がなぜできるのでしょうか？

その答えを示すために少し著作権について触れておきましょう。

著作権とは、自ら思想・感情を創作的に言語、音楽、絵画、建築、図形、映画、写真、コンピュータプログラムなどの表現形式によって表現した創作物(著作物)に対して、その利用を著作物の創作者(著作者)が占有することを法律(著作権法)で定めた権利をいいます。

従って、あなたが著作権者でないのならばその著作物を利用できないということになります。

では、著作権者ではないあなたが、その著作物を利用したい場合、どのような方法があるのでしょうか？

一つに著作権法は著作権者にその権利を他人に譲渡することを認めています。しかし、著作権者はあなたに権利を譲渡することでその権利を失ってしまいますから簡単には権利を譲ってくれないでしょう。また、著作権者が死亡することで権利を相続することもできますが、これはあなたが著作権者の法定相続人である、または著作権者が遺言書に自ら亡くなった後に権利を贈与する者として指定する必要がある、これも一般的ではありません。

他方、著作権法は、著作権者に他人に対して著作物の利用を許諾できることを認めています。さらに許諾を受けた他人がその著作物を利用する際は、著作権者の許諾に係る利用方法及び条件の範囲内においてこれを利用することができますと定めています。この方法をとれば著作権者は自らの権利を失うことなく、いつでも、他人であるあなたに対して著作物を利用させることができるだけでなく、その利用についての方法や条件を定めることができます。これがライセンスといわれているものです。

また、OSSの著作権者は、そのOSSを利用に対して自らライセンスを策定したり、既存のOSSライセンスを利用したりすることができるだけでなく同じコード内容のOSSに対して異なるライセンスを併存させることもできます。ライセンスは、OSSを利用する上で欠かせない要件です。あなたが、OSSを利用するのであれば、まずOSSのライセンスを入手し、あなたの利用したい条件と矛盾しない内容であることを確認することが大切です。

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.2

前回、OSSを利用する際には、OSSライセンスの利用条件を確認することが大切と述べましたが、このOSSの利用とは、著作権を行使することをいいます。

ところでこの著作権ですが、その内容は支分権と呼ばれるいくつかの権利から構成されている集合体を称したもので、単一の権利ではありません。また、著作権を構成する支分権は「著作財産権」と「著作者人格権」に分類することができます。

我が国の著作権法では、「著作財産権」としては複製権、上演権・演奏権、上映権、公衆送信権(送信可能化権、公に伝達する権利を含む)、口述権、展示権、頒布権、譲渡権、貸与権、翻訳・翻案権、二次的著作物の利用権。そして「著作者人格権」としては公表権、氏名表示権、同一性保持権が規定されています。

「著作財産権」と「著作者人格権」の違いですが、著作物の権利を保護する「著作財産権」は文字通り財産的な性格を有しており、その権利の一部または全部を譲渡や相続等により移転することができます。

OSSライセンスの中に「Author: 著作者」と「Copyright Holder: 著作権者」という文言を見かけることがありますが、著作物を創作する著作者はもちろん、著作権者になり得るのですが、権利の移転により、直接、創作に携わらなくても著作権者になり得る場合もあるため、条項の内容によって使い分けしているケースもあります。

著作財産権をもって狭義の「著作権」と称されることもあります。一方、「著作者人格権」は、著作者であることを保護する権利であり、移転の対象にあたらぬこともあり一身専属権とも称されています。「著作者人格権」は著作者の死亡によってその権利は消滅しますが、著作者の死後でも一定の範囲において保護されることになっています。

あなたが、他人の著作物を利用する場合、譲渡や相続によらなければ、著作者から利用の許諾を得ることになりますが、著作権者はこの許諾について支分権ごとにこれを行なうことができるだけでなく、利用に伴う期間や地域の限定等種々の条件を自由に規定することができます。

あなたが、あるOSSについて、自由に複製ができるとするならば、それは、そのOSSの著作権者が専有している複製権についてあなたの行使を許したことにはなりますが、必ずしも無条件で権利を行使できるということではありません。従って、OSSライセンスを確認する際は、利用を許諾された支分権は何か？利用にあたっての条件は何か？にまず着目することが望ましいでしょう。

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.3

OSSを利用する際に、許諾の対象となる主だった支分権について個々に触れていきましょう。

まずは、「複製」とはどういうことでしょうか？

あなたが、OSSを入手しようとしてWebサイトからダウンロードしたり、それを他人に配布するためにCDやDVDに書き込んだりする行為は、「複製」にあたります。「複製」とは、著作物を新たな有体物、つまり一般的にメディアといわれるHD、CD、DVD、ROMやマイクロチップ等に固定することをいいます。

次に、あなたが入手したOSSを実行するために、ソースコードをコンパイル・リンクして実行イメージを生成したとします。この場合、生成されたイメージはコンパイラやリンカーによってソースコードとは異なる形態になりますが、法的には同一のものと解されており、かつ生成されたイメージもメディアに固定されていますので、複製物にあたるとされています。

では、固定されるとはどういうことでしょうか？たとえば、メモリー上にロードされるOSSの実行イメージあるいはOSSをクラウド環境のWebアプリケーションとして利用する場合などは、厳密言えばメディアに固定されているとはいえません。

こうした利用は「一時的蓄積」と定義づけられていますが、欧米先進国では、一時的に蓄積されることも「複製」に該当するとした上で、複製権を制限すべきという解釈がなされています。

複製権の制限とは、著作権者の許諾を得ずとも複製行為ができるということで、EUでは、権利制限規定として、契約において特段の定めがない、プログラムの利用に必要であること、プログラムの適法な取得者によること、プログラムの意図された使用目的(プログラムの利用許諾契約において規定されることが考えられる種々の条件)に合致していることが要件として明文化されています。

一方、我が国では一次的蓄積に関する明文規定がないこと、「複製」の概念とするには、蓄積行為が発生する都度、著作権者の許諾を要する可能性もあることから解釈が分かれ、欧米と同様の見解に立つことに慎重でしたが、平成21年の著作権法の改正で、電子計算機における著作物の利用に伴う複製(法47条の8)として、例えばOSSの実行(メモリー上の一次的蓄積)やクラウド環境におけるOSSのブラウザ上での利用(ユーザー端末上の一次的蓄積)行為そのものは、その行為の対象となるOSSの利用について著作権を侵害していなければ、著作権侵害とならないことが明文化されました。

従って今後は、「複製」の範囲や権利制限のおよぶ範囲についても欧米と差異のない扱いがなされるものと考えられます。

【ご参考URL】

<http://moe3.co.jp/2012/05/ebata/>

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.4

OSSは、その言葉どおりソースコードが開示されているソフトウェアですが、では、ソースコードがなぜ開示されているのでしょうか？

多くのOSSに採用されているライセンス「GPL」の第3版では、「ソースコードは、その著作物に改変を加えるに当たって好ましいと考えられる形式のことであり(1. ソースコード)」、Androidが採用している「Apacheライセンス」の第2版では、「ソース形式とは、ソフトウェアのソースコード、ドキュメントソース、設定ファイルといった改変を加えるのに都合の良い形式(1. 定義)」と規定されています。

つまり、ソースコードの開示はOSSの改変を可能にするためであることが推測できます。

ではこの「改変」について触れてみましょう。

我が国の著作権法には「改変権」という権利はありませんが、支分権として「翻訳権・翻案権」という権利があります。

翻訳権は、別の言語に翻訳する権利ですが、OSSですと、他のプログラム言語に書き換える行為が該当するでしょう。

他方、翻案権は、元の著作物の特徴を維持しつつ、具体的な表現形式を変更、あるいは追加することで新たな著作物を創作する権利ですが、OSSでは、機能の変更や追加がなされたバージョンアップ等が該当するでしょう。

我が国では「翻訳権・翻案権」の他に、著作者の人格的利益を保護するために、「同一性保持権」として、著作者の意に反して変更、切除その他の改変を禁止できる権利を規定しています。

しかし、多くのOSSライセンスを生んだ米国の著作権法においては「同一性保持権」の適用は視覚芸術著作物に限定され、プログラムには適用されていないため、当初、ライセンスの適法性について問題視する意見も聞かれましたが、今日では概ね以下のような見解がとられています。

つまり、OSSにおいて改変が可能であるということは、そもそも著作権者が「翻訳権、翻案権」を許諾しているという前提の上で、同時に著作者として、その翻案に必然に伴う改変の限度で同一性保持権の不行使について黙示の同意を与えているという考え方、あるいはその翻案に必要な限度での改変は同一性保持権の適用除外規定(特定のコンピュータで利用できるようにしたり、より効果的に利用し得るようになるために必要な改変)に該当するとして、実務上よく行われている他のチップやOS上への移植、プログラムの不具合修正、プログラムの最適化などは同一性保持権を主張し得ないであろうという考え方などが挙げられています。

【ご参考URL】

<http://moe3.co.jp/2012/05/ebata/>

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.5

OSSは特徴として、そのソフトウェアの「配布」を行うことができるとされています。

ソフトウェアの「配布」とは、ソフトウェアを固定した媒体を他人に供給すること(この場合、我が国の著作権法で言えば「譲渡権」又は「貸与権」若しくは「公衆送信権等」といった権利行使がなされる)、言い換えれば他人が当該ソフトウェアを利用できる状態におくことをいい、これは当該ソフトウェアの複製媒体による供給、ネットワーク(LAN、WANにかかわらず)を通じた供給あるいは当該ソフトウェアが格納されたサーバーに対するアクセス権の付与等、形式形態にはこだわりません。

ところで、我が国の著作権法第30条では、個人的に又は家庭内その他これに準ずる限られた範囲内における使用(私的な使用)を目的とするときには、著作権者の許諾を要することなく著作物を複製(及び配布)することを認めています。が、企業の内部的使用のために複製(及び配布)は、私的使用の範囲に該当しない(東京地判昭和52年7月22日)とされています。

しかし、OSSライセンスの多くは法の規定とは多少異なる概念を有し、一企業・団体内における内部的使用も私的使用の範囲として、複製した著作物を社内で使用しても「配布」の権利を行使したことにはあたらないとしています。

また、企業がソフトウェア開発を外部の企業に委託する際、委託元の企業が有するOSSを使って委託先企業が開発する場合や、委託先企業がOSSを使って開発したシステムを委託元企業に納品する場合があります。が、これらの場合においても、当該OSSについては「配布」の権利が行使されたとはみなされないとされているのが、一般的です。

以上、「複製」「改変」「配布」とOSSライセンスが規定する著作権の代表的な許諾の実際について述べてきましたが、「複製」「改変」と「配布」には著作権者に対する影響度において若干の差異があります。

たとえば、一般的な著作物においても、許諾した相手が、自身のためにだけに複製することは勿論のこと、改変(仮に違法でも)して、当該著作物を使用しても、著作権者に、与える実態上影響は実際それほど大きくありません。

しかし、これを第三者に配布するとすると、影響は(それが、合法違法に関わらず)異なってきます。

したがって、ソフトウェアを使用(実行)するための条件に言及した多くの商用ソフトウェアライセンスとは異なり、OSSライセンスは、ソフトウェアの(改変を含む)再利用を公平かつ広く遍く即すことを前提としていることから、「配布」の権利行使については、当該ライセンスごとに、その条件規定が多様です。

一方、OSSは、その使用(改変を行うことの有無にかかわらず)が個人あるいは一企業・団体内にとどまるならば、ライセンス内容を、それほど意識することなく利用できるソフトウェアとも言えるでしょう。

【ご参考URL】

<http://moe3.co.jp/archives/39>

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.6

実際、OSSライセンスは「配布」に対する条件規定で構成されているといっても過言ではありません。

これは、先にも述べましたが、The Open Source Initiative(OSI)が発表した「The Open Source Definition」(OSD)の冒頭、『The distribution terms of open-source software must comply with the following criteria: –OSSの配布条件は、以下の基準を満たしていなければなりません。』と配布条件に言及していることから、ご理解頂けるものと思います。

また、OSSライセンスは、当該ライセンスごとに、その条件規定が多様であるとも述べましたが、勿論、OSDに従っているものについては、その基準を満たした上での条件規定であることは言うまでもありません。

では、その基準について、改めてOSDの条項をみてみましょう。

『1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources.

The license shall not require a royalty or other fee for such sale.

第1条 再配布の自由

「ライセンス(OSSライセンス)は、出自の様々なプログラムを集めたソフトウェア配布物の構成要素として、ソフトウェアを販売あるいは無料で配布することを制限してはなりません。

ライセンスは、このような販売に関して使用料その他の報酬を要求してはなりません。』

一見、矛盾しているようですが、OSSのライセンサーを「A」、ライセンシーを「B」そして当該ライセンス契約の当事者である「A」「B」以外の第三者を「C」(この条項では"any party")として、この条項を読んでみましょう。

すると(「A」が権利を有する)ソフトウェアを(「B」が「C」に)販売(有償)しても無償で配布しても構わないが、(「B」が「C」に)ソフトウェアを(有償無償にかかわらず)配布する際は、(「A」が「B」に対する)使用料やその他の報酬を要求する条件をつけてはならないと読めます。

よく、「OSSは無償」と表現されていることから、「OSSはタダ」と認識されている方も少なくないと思いますが、無償とは、この条項で言えば、ライセンシーによる「配布権」の行使についてライセンサーがその許諾を無償で行うことを求められているのであって、ソフトウェア自体の無償配布を求められているのではないことがわかります。

逆に、有償のOSSを不思議に思う方もいらっしゃるかもしれませんが、こちらが矛盾していないこともおわかり頂けると思います。

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.7 ①

今回も、引き続きOSD条項について解説します。

『2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form.

Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge.

The source code must be the preferred form in which a programmer would modify the program.

Deliberately obfuscated source code is not allowed.

Intermediate forms such as the output of a preprocessor or translator are not allowed.

第2条 ソースコード

「オープンソース」であるプログラムは、ソースコードを含んでいなければならない、コンパイル済形式と同様にソースコードでの配布も許可されていなければならない。

何らかの事情でソースコードと共に配布しない場合には、ソースコードを複製に要するコストとして妥当な額程度の費用で入手できる方法を用意し、それをはっきりと公表しなければならない。
方法として好ましいのはインターネットを通じての無料ダウンロードです。

ソースコードは、プログラマがプログラムを変更しやすい形態でなければならない。

意図的にソースコードを分かりにくくすることは許されませんし、プリプロセッサや変換プログラムの出力のような中間形式は認められません。』

本条は、OSD準拠のOSSライセンスが適用されるプログラムを配布する際に、配布先に対して、ソースコード配布の用意を義務付ける規定です。

なぜソースコードを配布する必要があるのでしょうか？

これはOSSに付与される著作権法上の権利の許諾と関係しています。通常、プログラムはバイナリーコード（実行形式）で配布されますが、これを単に使用（実行）する上で不都合はありませんし、再配布をしたい場合でも、複製権や配布権が許諾されていれば、問題なくこれを行うことができます。



P18に続く

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.7 ②



P17の続き

では、改変したい場合はどうすれば良いでしょうか？

バイナリコードのプログラムに対して改変を行うことは、決して一般的とはいえません。

事実、バイナリコードはこれに相当するソースコードから生成されます。

つまり、改変権の行使には、ソースコードは不可欠なのです。

そこで、OSSの配布は原則、ソースコードによる配布を考慮し、バイナリコードのみを配布した場合でも、改変権の許諾を担保するために、ソースコードを提供する用意が規定されているのです。

この時、ソースコードの入手に対する費用負担について、これが実費程度に抑えられているのは、バイナリコードの受領者が、高額を理由にソースコードの入手が叶わず、結果、改変権が許諾されているにもかかわらず、改変ができないという矛盾を防ぐための配慮と解釈できますし、また、中間形式の配布を認めない規定も同様と解釈できます。

OSD準拠とされるOSSライセンスであっても、特にバイナリコードの配布に伴うソースコードの配布についての明示のないことから、そのライセンスが提供されたOSSにおいては、ソースコードを提供する義務がないと理解されている方も少なくありませんが、改変権が許諾されている以上、受領者によるソースコードの入手は許可されているとの解釈が妥当といえるでしょう。

これならわかるOSSライセンス Vol.8 ①

今回も、引き続きOSD条項について解説します。

『3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

第3条 派生したソフトウェア

ライセンスは、ソフトウェアの変更および派生ソフトウェアの作成並びにそれらのソフトウェアを元のソフトウェアと

同じライセンスの下で配布することを許可しなければなりません。』



P19に続く

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.8 ②

P18の続き

本条は、オリジナルソフトウェア(OSD準拠のOSSライセンスが適用されるソフトウェアを示しています。)について、改変物および派生物の生成を許諾するとともに、改変物または派生物を配布する際の条件は、オリジナルソフトウェアのライセンスと同じ条件(すなわち、改変物または派生物に対しても、改変物または派生物の生成を許諾することと、再配布時には、この条件を継承すること)に従うということを規定しています。

この規定は、前回ご説明したソースコードの提供にも関連しますが、オリジナルソフトウェアのみならず、これに改変を加えた、または派生したソフトウェアに対しても、それらを配布する際にはソースコードの提供が前提にあると解釈できます。

では、派生物は何を意味しているのでしょうか？

米国著作権法(第101条)では、derivative work(派生的著作物)として、「…翻訳、…またはその他著作物を改作し、変形もしくは翻案した形式のように、一つまたはそれ以上の既存の著作物を基礎とする著作物をいう。」とされています。

また、我が国の著作権法(第2条11)では、二次的著作物として、「著作物を翻訳し、編曲し、若しくは変形し、又は脚色し、映画化し、その他翻案することにより創作した著作物」との規定があります。

こうした法令を踏まえると、本条項で規定された改変物とは、オリジナルソフトウェアに何らかの変更が加えられたオリジナルと同一でないソフトウェアであると言えます。

さらに派生物とは、オリジナルソフトウェアの表現上の本質的な特徴の同一性を維持しつつ、具体的な表現形式を変更して新たに創作されたソフトウェアと解することができるでしょう。

ところで、ソフトウェアは、実現する機能が多様化、複雑化傾向にあること、当該機能を実現するために、プログラムサイズが必然的に巨大化してきた背景がありますが、拡張性や保守性の観点からも、従来より、多数のモジュールと称するプログラム群で構成されているのが通常です。

著作権法では、このモジュールもひとまとまりの独立したプログラム著作物として解されていますが、OSSライセンスをソフトウェア全体で見たとき、モジュール間の独立性と依拠性はどのように考えれば良いのでしょうか？

すなわち、OSSライセンスが適用されたモジュールに対して、この改変物または派生物はオリジナルソフトウェアに依拠するソフトウェアとして、本条項が適用されることに、疑問を抱かれる方は少ないと思われます。

一方で、OSSライセンス適用モジュールとそれ以外のモジュールの組み合わせによって構成されたソフトウェアに対するライセンスの適用については多くの方が疑問を残されているのではないのでしょうか？

次回は、他の著作物ではあまり見られない、ソフトウェアプログラム特有のモジュールに着目し、派生物の考え方、ライセンスの適用についてさらに詳しく触れてみたいと思います。

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.9 ①

前回、触れたソフトウェアプログラムにおけるモジュールの考え方について解説します。

その前に、まず

- ・ソースコードファイル
- ・オブジェクトコードファイル
- ・実行イメージファイル

に対する著作物としての基本的な考え方について説明しておきましょう。

ご存じのように、プログラムは基本的には作成したソースコードファイルをコンパイル(またはアセンブル)することで、オブジェクトコードファイルが生成され実行させることができます。

OS等種々の環境上で稼働する今日では、こうした実行環境が提供するシステムソフトウェアとリンクして、実行イメージファイルが生成されるのが一般的と言えるでしょう。

では、この場合のソースコードファイル、オブジェクトコードファイル、実行イメージファイルの関係について整理してみましょう。

最初にソースコードファイルとオブジェクトコードの関係です。

「ソース・コードの複製物にあたるオブジェクトコードをコピーすることは、ソース・コードの著作権を侵害したことになる」(東京地方裁判所1982年12月6日判決、判例時報1060号18頁)

との判例もあるように、著作権法上では、ソースコードファイルに対してオブジェクトコードファイルは複製物にあたりとされています。

では、オブジェクトコードファイルと実行イメージファイルの関係はどうでしょうか？

なるほど単一のオブジェクトコードファイルとして実行できる環境であるならば、ソースコードファイルがコンパイラ(アセンブラ)を通してオブジェクトコードファイルに複製され、次にリンクを通して実行イメージファイルが生成されることになりますので、実行イメージファイルも同じように複製物にあたるといえるでしょう。

言い換えれば、ソースコードファイルも、オブジェクトコードファイルも、実行イメージファイルも基本的には同じ一つのプログラムであるということになります。

しかしながら、そもそもソフトウェアは大規模かつ複雑多岐にわたる機能を実現する関係上、実際は単一のプログラムでこれを構築することはまずなく複数のプログラムで構築されているのが通常です。

そこでモジュールの必要性がでてくるわけですが、そもそも、モジュールとはどういうものなのでしょうか？



P21に続く

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.9 ②



P20の続き

1972年、デビッド・L・バルナスが、彼の論文

『On The Criteria To Be Used in Decomposing Systems into Modules』で、

「大規模なソフトウェアを複数の開発者が、何の配慮もせずに分担開発しようとするコミュニケーションの調整に負担がかかるばかりではなく、事実上開発が不可能になるため、ソフトウェアを機能的に意味あるまとまりに整理、調整コストの軽減に向けて、まとまり同士の相互依存性も低くした単位」

として提唱したのが、ソフトウェア分野におけるモジュールの概念とされています。

モジュールは開発を行う上でモジュール間相互の依存性は低いほど望ましく、また依存性が低ければ低いほどその関係は独立しているといえます。

モジュール間の独立性が高ければ派生関係も生じ得ないことになるわけですが、他方、モジュール化は、複数のソースコードファイルが生成されることを意味します。

つまり、上述したソースコードファイル、オブジェクトコードファイル、実行イメージファイルの関係は、それぞれのソースコードに対するオブジェクトコードファイルが複製され、それらが組み合わさることで一つの実行イメージファイルが形成されることになるので、1:1:1からn:n:1へと変化することになります。

実は、この時に実行イメージファイルとその構成要素たるオブジェクトコードファイル群との間に、派生という関係が生じるのです。

すなわち、OSSライセンス適用モジュールとそれ以外のモジュールの組み合わせによって構成されたソフトウェアに対するライセンスの適用というのは、

「ソースレベルあるいはオブジェクトレベルにおけるモジュール間の相互関係に対して」ではなく、「モジュール化されたオブジェクトコードファイル群とそれらから生成された実行イメージファイルとの間の派生関係に対して」ということなのです。

ところで、この派生関係は、実行イメージファイルを生成する創作形態で法的な解釈も異なります。

次回は、このことについて順を追って解説していきましょう。

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.10 ①

OSSを利用するうえで、ライセンスの理解が難しいのがモジュールにおける派生の考え方です。

そこで、まず、ソフトウェアを開発するときの原点に立ち返って考えてみましょう。

ある機能を実現するために、プログラムを開発しますが、まずソースコードファイルを生成します。機能がシンプルであれば、一つのソースコードファイルをコンパイルし実行イメージが生成し、これを実行します。

では、プログラムを書く際、同じか同じように処理できる部分があったらどうするでしょうか？あるいは、異なる機能が存在した場合はどうでしょうか？

こういった場合、開発者の誰でもそうですが、趣味や学習用のプログラムでなければ、当然、ある部分に変更が生じたときやテストを行うときの「容易性」や、同時にそのプログラムの「再利用性」も考慮するでしょう。これがモジュール分割の一般的な方法としての「機能分割」の行動になっていくわけです。

機能分割は、機能としての「独立性」に着目して分割を行っていきます。従って、分割された機能要素は開発時における「相互作用」が小さくなっているでしょうから、結果、別々に開発することを可能にします。

この時、一人の開発者(一つの開発会社)ではなく、二名以上の開発者(二社以上の開発会社)が、分割されたモジュールそれぞれを開発した場合、著作権上の権利関係はどうなるでしょうか？

例えば、「X」というソフトウェアシステムを構築する過程において、機能分割を行なった結果、(A)(B)(C)3つのモジュールに分けられ、それぞれの開発を「A」「B」「C」3名が開発者が行ったとしましょう。

プログラムの著作権法上の定義は、「電子計算機を機能させて一の結果を得ることができるようにこれに対する指令を組み合わせたものとして表現したものをいう」(第2条第1項 第10号の2)となります。

条文上にある「一の結果」とは、処理による機能実現と認識されていますので、当然機能要素として分割されたモジュールも一つプログラムであり、それが創作性を有するものであれば著作物と認められ、開発者それぞれが当該モジュールの著作権者として権利を行使することができます。

つまり、モジュール(A)の著作権者は「A」、モジュール(B)(C)の著作権者は同じく「B」「C」となります。

他方、一の機能を得るためには、電子計算機を機能させなければならないとされているので、それぞれのモジュールをコンパイルし、オブジェクトコードファイル(a)(b)(c)を生成したとします。

オブジェクトコードはソースコードの複製にあたと解釈されていますから、(a)(b)(c)の著作権者もそれぞれ「A」「B」「C」と言えそうです。触れていきます。


 P23に続く

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.10 ②


 P22の続き

では、このオブジェクトコードファイルから実行イメージファイルを生成した場合、権利関係はどうなるのでしょうか？

実行イメージを生成する場合に使用されるリンカーはオブジェクトファイルを解析し、参照している他のオブジェクトファイルやライブラリなどを探し出し、必要なものを結合して実行イメージファイルを生成します。

リンクの方式には、実行イメージファイル(例えば<X>とする)に対してすべてのモジュールを結合する静的リンクがあります。

また、実行イメージファイルとして<A><C>を生成し、例えば<A>の実行時に他の実行イメージファイル<C>をメモリ上に呼び出す動的リンクがあります。

おそらく読者の大半は、静的リンクの場合は一つの実行イメージファイルにまとまっているが、動的リンクの場合は<A><C>は独立しているので著作権法上の権利関係も異なると考えていらっしゃるのではないのでしょうか？

でも、ちょっと待ってください。

静的リンクの場合でもオブジェクトファイルである(a)の中に他の(b)(c)のオブジェクトが含まれているのではなく、相互の参照(依存)関係の上に一つの実行イメージファイル<X>が生成されているのであって、(a)(b)(c)は<X>の実行イメージにおいて不可欠な構成要素という位置付けとなります。

一方、実行イメージファイルとして生成された<A><C>についても、確かにファイルとしての独立性はあっても、オブジェクトファイル(a)(b)(c)相互の参照(依存)関係は静的リンクのそれと何ら変わらず、(a)(b)(c)は<A><C>による一連の実行イメージにおいて不可欠な構成要素という位置付けであることには変わりはありません。

このことから、<X>はもちろん<A><C>の実行イメージも、(a)(b)(c)からみた場合、(a)の(b)のそして(c)の派生物と位置付けられ、結果、著作権者「A」「B」「C」は、<X>、<A><C>いずれに対しても権利を行使できるのです。

つまり、リンクという結合方式が、あくまでもプログラムがメモリ上で実行される際のアドレス割付けのメカニズムである以上、静的か動的かという形式の違いは、あくまでもアドレス形成のタイミングという技術的な違いに過ぎず、その違いによって、著作権法上の権利関係までが異なることにはならないのです。

次回は、権利関係の考え方とそれに伴うモジュール取扱いについて触れていきます。

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.11 ①

今回は、二つ以上の著作物あるいは二人以上の著作者が伴う著作物に関する著作権法上の形態とモジュールの取扱いについて整理してみましょう。

まず、ある既存の著作物を翻訳または翻案(著作物の内面形式(思想・感情)を維持しつつ、その外面形式(表現)を変えること)して創作された二次的著作物があります。

次に、2人以上の者が共同して創作する意志をもって創作した著作物で、その各人の寄与を分離して個別に利用することができない(不可分といいます)共同著作物があります。

また、その対比として、我が国の著作権法上、明文化はされていませんが、概観上は1個の著作物のようでありながら、著作物全体の創作に関しては共同行為がみられず、それぞれ独立した著作物が結合している結合著作物という考え方があります。
(米国著作権法では、それ自身が別個独立の著作物となる多数の寄与物が1つの集合体に編成されている形態を集合著作物として明文定義しています。)

さらには、その集合著作物も含まれますが、著作物であるなしにかかわらず複数の素材の選択または配列による創作性をもつことで権利が生ずる編集著作物もあります。

では、こうした形態に対してモジュールは、どう位置付けて考えれば良いのでしょうか？

例えばモジュール(O)があったとします。これが「O」と「O'」という複数の開発者によって作成された場合、この(O)は「O」、「O'」による共同著作物となります。

では、この(O)に改変を加えてモジュール(P)を作成したとします。この場合、モジュール(P)はモジュール(O)の二次的著作物になります。

また、モジュール(O)と別に存在するモジュール(Q)を組み合わせた場合、その組み合わせた結果は、結合著作物となります。

さらに、(O)と(P)を組み合わせて(Y)という成果物が生じる場合、(Y)は(O)(P)の集合著作物あるいは編集著作物になり得るかもしれません。

それゆえに、昨今のソフトウェアはモジュールが集まってシステムを構成しているので集合著作物と明言される方もいらっしゃると思います。

しかし、一概にそう定義づけて良いものでしょうか？



P25に続く

3. OSSコンプライアンス紹介

これならわかるOSSライセンス Vol.11 ②



P24の続き

モジュール(O)と(Q)の組み合わせを結合著作物とするならば、動的リンクであれ静的リンクであれ、結合著作物に変わりはないでしょう。

にもかかわらず、リンク形式の違いで権利関係を異なるものと認識されている方が多いのはなぜでしょうか？

また、成果物(Y)ですが、実は、モジュール(O)(Q)が(Y)を開発や保守の都合上で分割した結果(機能分割を行う上で開発効率や保守の容易性は当然分割の要因になっているものと思われます)による作成物だったとすると(O)と(Q)は、(Y)を実現する上では不可分な要素と位置付けられます。

その結果、成果物(Y)は、開発者の「O」と「Q」に加えて「Q」も著作者となる共同著作物との解釈も可能性になります。

つまり一口にモジュールといっても、単体個別のモジュールの組み合わせによる結果なのか、モジュール分割という所作に起因した再構成なのかで、著作権法上の形態は全く異なるものになるのです。

ここまでの説明はソースコードファイルレベルでの並列な関係を想定したものですので、二次的著作物は(O)の改変による(P)ということで、この形態については皆さんも異論がないものと思います。

しかし、前回のおさらいにもなりますが、(P)が(O)を基にした著作物であるという点をとらえれば、ソースコードファイルから実行イメージファイルが生成されるという垂直な関係においては、二つ以上のソースコードから生成される実行イメージファイルは、その生成が動的リンクであれ静的リンクであれ、ソースコードのプロシージャが明らかに含まれているわけです。

それで、個々のソースコードファイルに対する二次的著作物にあたるという解釈も成り立つのです。

事実、LGPLv (GNU 劣等一般公衆利用許諾契約)2.1のセクション5には、こうした解釈の内容が明文化されています。

従って、モジュールの取扱いについては、単にモジュールという形態のみにとらわれることなく、個々のモジュールが各々モジュールとして形成された経緯にまで視野を広げることが権利関係を把握する上では本質的な重要性を持ちます。

その上で、モジュール自体と当該モジュールから直接構成されている全体との関係を再認識することが大切になってきます。

そこで、次回、このモジュールと全体の関係をどう考えるかについて、もう少し掘り下げてみます。

4. バグ&セキュリティ情報

4. バグ&セキュリティ情報

<2012/03/22 配信>

Samba の smbd の処理に任意のコードを実行される問題

Samba は UNIX および UNIX 互換マシンを Windows 互換のファイルサーバ/プリントサーバとして提供するプログラムです。

Samba は、smbd の処理が原因でセキュリティホールが存在します。

攻撃者にこのセキュリティホールを利用された場合、リモートから任意のコードを実行される可能性があります。

これは Samba の chain_reply 関数 (source/smbd/process.c) において、Batched ("AndX") リクエストを適切に処理していないため、バッファオーバーフローを起こす可能性があることが原因となります。

本セキュリティホールによりシステムは、リモートの攻撃者に細工された SMBリクエストを送信されることで、smbd プロセスの権限 (root) で任意のコードを実行される可能性があります。

<2012/04/25 配信>

Samba の PIDL の処理に任意のコードを実行される問題

Samba は UNIX および UNIX 互換マシンを Windows 互換のファイルサーバ/プリントサーバとして提供するプログラムです。

Samba は、Perl ベースの DCE/RPC IDL (PIDL) コンパイラの処理が原因で、セキュリティホールが存在します。

攻撃者にこのセキュリティホールを利用された場合、リモートから任意のコードを実行される可能性があります。

<2012/05/24 配信>

CGI 版 PHP のクエリ文字列の処理に任意のコードを実行される問題

PHP (Hypertext Preprocessor) は HTML に記述するスクリプト言語で CGI として使用でき PHPモジュールを Apache サーバに組み込むことにより Perl/CGIと比較して処理速度の高速化やサーバ負荷の低減が可能です。

PHPは、CGI 版 PHP のクエリ文字列の処理が原因でセキュリティホールが存在します。攻撃者にこのセキュリティホールを利用された場合、リモートからCGIで動作する PHP プログラムを経由して、任意のコードを実行される可能性があります。

4. バグ & セキュリティ情報

<2012/06/27 配信>

PHP の Phar Extension の処理に任意のコードを実行される問題

PHP (Hypertext Preprocessor) は HTML に記述するスクリプト言語で CGI として使用でき PHP モジュールを Apache サーバに組み込むことにより Perl/CGI と比較して処理速度の高速化やサーバ負荷の低減が可能です。

PHP は、Phar Extension の処理が原因でセキュリティホールが存在します。

攻撃者にこのセキュリティホールを利用された場合、Phar Extension を利用する PHP プログラムで細工された tar ファイルを処理することで任意のコードを実行される可能性があります。

Ruby on Rails の Active Record の処理に SQL インジェクションの問題

Ruby on Rails は Ruby による Web アプリケーションフレームワークです。

Ruby on Rails は、Active Record の where メソッドの処理が原因でセキュリティホールが存在します。

このセキュリティホールを利用された場合、Rails アプリケーションへ細工されたリクエストを送信されることで、SQL インジェクションによる攻撃を受けデータベースを不正に操作される可能性があります。

<2012/07/24 配信>

Perl の Digest モジュールに任意のコードを実行される問題

Perl (Practical Extraction and Report Language) は、インタープリタ型のプログラミング言語です。

Perl は、Digest モジュールの処理が原因でセキュリティホールが存在します。

攻撃者にこのセキュリティホールを利用された場合、リモートから任意のコードを実行される可能性があります。

4. バグ & セキュリティ情報

<2012/08/23 配信>

GNU C ライブラリ (glibc) の フォーマット文字列の処理に任意のコードを実行される問題

GNU C ライブラリ (glibc) は、GNU (GNU is Not Unix) が提供する ISO C および POSIX をサポートした C ライブラリです。

GNU C ライブラリ (glibc) は、フォーマット文字列の処理が原因でセキュリティホールが存在します。

このセキュリティホールを利用された場合、printf ファミリー関数のフォーマット文字列として細工された文字列を入力されることで、セキュリティ機構を迂回して不正に任意のコードを実行される可能性があります。

<2012/09/26 配信>

PostgreSQL の XML の処理に システム上の任意のファイルへアクセスされる複数の問題

PostgreSQL は POSTGRES, Version 4.2 をベースにした Object Relational Database Management System (ORDBMS) です。

PostgreSQL は、xml2 contrib モジュールの処理および XML の解析処理が原因で複数のセキュリティホールが存在します。

これらのセキュリティホールを利用された場合、データベースのユーザにデータベースサーバ上の任意のファイルを、不正に読み書きされる可能性があります。

<2012/10/25 配信>

Apache の mod_negotiation の処理に HTTP レスポンス分割攻撃を受ける問題

Apache はモジュールを組み込むことにより様々な機能を追加することが可能なウェブサーバプログラムです。

Apache は、mod_negotiation の処理が原因でセキュリティホールが存在します。攻撃者にこのセキュリティホールを利用された場合、リモートから HTTP レスポンス分割 (Http Response Splitting) 攻撃を受ける可能性があります。

4. バグ & セキュリティ情報

<2012/11/22 配信>

Ruby のハッシュテーブルの処理に DoS 攻撃を受ける問題

Ruby はオブジェクト指向プログラミングの機能を持つインタープリタ型のプログラミング言語です。

Ruby は、ハッシュテーブルの処理が原因でセキュリティホールが存在します。

<2012/12/25 配信>

MySQL の SQL の処理に任意のコードを実行される問題

MySQLはマルチユーザ/マルチスレッド対応のSQLデータベースプログラムです。

MySQLは、SQLの処理が原因でセキュリティホールが存在します。

このセキュリティホールを利用された場合、認証されたデータベースユーザによって、サーバ上で任意のコードを実行される可能性があります。

<2013/01/23 配信>

Ruby on Railsのパラメータ解析処理に任意のコードを実行される問題

Ruby on Rails は Ruby による Web アプリケーションフレームワークです。

Ruby on Rails は、パラメータ解析処理が原因でセキュリティホールが存在します。

このセキュリティホールを利用された場合、リモートからRails アプリケーションで、任意のコード実行、SQL インジェクション、サービス妨害、および認証の迂回による攻撃を受ける可能性があります。

5. オープンソースまるごとコンサルティング

5. オープンソースまるごとコンサルティング

第20回 Liferayのマルチテナント機能について知りたい

Q.

自社クラウドサービスとして顧客向けポータルを提供を検討しています。OpenStandiaでサポートしているLiferayのマルチテナント機能について、詳細を教えてください。また、Liferay導入事例、商用製品との機能比較もあれば教えてください。

A.

マルチテナント機能は、1つのサーバ環境を区切って複数の企業様(テナント)向けに単一/複数のサービス(インスタンス)を提供する機能です。テナントとインスタンスの関係は下記になります。

(1)マルチテナント・シングルインスタンス

A社
 B社 ———— ポータルサービス(3社共通アプリケーション)
 C社

(2)マルチテナント・マルチインスタンス

A社 ———— ポータルサービス(A社用アプリケーション)
 B社 ———— ポータルサービス(B社用アプリケーション)
 C社 ———— ポータルサービス(C社用アプリケーション)

Liferayは、ユーザー認証機能のほか、セキュリティ・ポリシーに従った適切なシステムの利用権限が設定できるため、上記のようなマルチテナント型クラウドサービスを実現することができ、OpenStandiaでは国内大手家電メーカークラウドサービスでのLiferay導入実績があります。

また、Liferayの導入事例および商用製品との機能比較は、OpenStandiaホームページにも掲載しています。Liferay紹介ページをご覧ください。

•Liferay導入事例:

http://openstandia.jp/oss_info/liferay/case-study.html?mm=55_2

•Liferay商用製品との機能比較:

http://openstandia.jp/oss_info/liferay/index.html?mm=55_3

Liferayを活用した企業情報ポータル構築ソリューションのご紹介も併せてご覧ください。

http://openstandia.jp/solution/eip/index.html?mm=55_4

★まとめ:

最近では、ポータル機能を有するOSSのCMS製品やグループウェア製品が出てきていますが、Liferayの機能の豊富さと決め細やかなアクセス制御機能はこれらの製品と一線を画しています。Liferayを活用することで、クラウド型サービスで重要なポイントとなるセキュリティを確保したサービス基盤を構築することができます。

OpenStandia営業マネージャー まるコン担当 谷川

5. オープンソースまるごとコンサルティング

第21回 Jaspersoftをエンタープライズ版に切り替えたい

Q.

エンドユーザー向けにJaspersoftで帳票を開発、すでに本番稼働中です。使用しているのがコミュニティ版なのですが、今後帳票の拡充が想定されているため、エンタープライズ版に切り替えてサポート契約を結ぶことを検討したいと思っています。NRIではどのようなサポート契約があるのか、サービス内容、費用等を教えてもらえますか？

A.

Jaspersoftエンタープライズ版の製品構成と価格(年間サポート代金含む)は、下記ページにてご紹介しています。

http://openstandia.jp/solution/jaspersoft/index.html?mm=56_1

Jaspersoftサポートは、スタンダード(年間12インシデント、対応時間平日9時~18時)と、プレミアム(インシデント無制限、24時間365日対応(重大度1の場合))の2種類です。

サポートでは Jaspersoftの利用方法や技術仕様についての質問への回答、障害対応、パッチに関する情報提供などを実施しています。詳細は、サービス契約書をご確認ください。

弊社は、日本初のJaspersoftの日本語サポートを2010年10月より実施しており、他のOSSのサポートと合わせてワンストップでサポートサービスをご提供することができます。

★まとめ:

Jaspersoftは、海外の大手企業や政府系機関での導入実績が数多くあり、商用BI製品と遜色ない機能を持ちながら、導入+5年間保守費用は4分の1程度ですみます。まずBIを試してみたい、スモールスタートで導入したいというケースには最適な製品です。お気軽にNRI-OSSCIにご相談ください。

OpenStandia営業マネージャー まるコン担当 谷川

5. オープンソースまるごとコンサルティング

第22回 グループネットワーク内にいない海外拠点との情報共有

Q.

弊社海外子会社のIT部門との情報共有・ノウハウ共有を実現するシステムをOSSで構築することを検討しています。グループネットワークに繋がった会社もありますが、そのネットワーク内にいない、まだ何もつながっていない会社に対しても情報共有を実現する必要があります。セキュリティ面で考慮すべきことを教えていただけますか？

A.

「インターネット経由の社内システムへのアクセス時に、セキュリティをどうやって担保するか」が大きな課題になると思います。セキュリティを担保する方法として、ID・パスワードでの認証のほかに以下の認証を行うのがお勧めです。

- ・PKI(クライアント証明書)
- ・ワンタイムパスワード
- ・マトリックス認証

また、さらに強固なセキュリティを確保するために、以下のようなことも検討されるとよいと思います。

- ・IPアドレスによる制限
- ・時間帯による制限
- ・ブラウザ種別による制限
- ・リスクベース認証

★まとめ:

ここ数年、会社利用のPCのほか、モバイルPC、スマートフォン、タブレットといった端末からのインターネット接続が増えてきました。多様な接続環境でのセキュアなアクセスを実現しながら、OpenAMとの連携によるシングルサインオンを行う方法をご紹介します。お気軽にNRI OSSCにご相談ください。

OpenStandia営業マネージャー まるコン担当 谷川

5. オープンソースまるごとコンサルティング

第23回 Apache Tomcatサポートサービスの卸売販売について

Q.

弊社の販売店様より、「Apache Tomcatに、Red Hat Enterprise LinuxやMySQLのようなサポート付きサブスクリプションはありますか？」とご質問いただきました。弊社で調べたところ、Apache Tomcatには、サポート付きサブスクリプションはないようですが、NRI OpenStandiaのApache Tomcatサポートサービスを見つけ、販売店様のご要望を満たせると思いました。弊社はディストリビュータとなりますが、Apache Tomcatサポートサービスの卸売は可能でしょうか？

A.

Apache Tomcatサポートサービスの卸売、すなわちサポートサービスを御社のサービスとして再販いただくことはもちろん可能です。Apache Tomcatサポートサービスは、OpenStandiaの中でも導入実績が多いサービスの一つで、エンドユーザー企業様だけでなく、システムの運用を担当されているSIer企業様、御社のようなディストリビュータ企業様にも数多くご利用いただいています。仕切り価格は別途ご相談させていただきます。

★まとめ:

NRI OpenStandiaのサポートサービスは、企業システムでよく利用されるOSSをまとめてサポートできることと、導入実績の豊富さが特徴です。OSS導入後のサポートサービスも、お気軽にNRI OSSCにご相談ください。

OpenStandia営業マネージャー まるコン担当 谷川

第24回 GoogleAppsとOpenStandia/Portalとの連携について

Q.

現在、GoogleAppsを利用していますが、OpenStandia/Portalとの連携はどのようなものが実現可能なのでしょうか。

A.

GoogleAppsで利用しているIDの管理や認証統合については、OpenStandiaで提供している別のソリューション「OpenStandia/SSO&IDM」(OpenAM)の標準機能で実現可能です。SAMLを利用しています。必要となるソフトウェアは下記ページをご覧ください。

<「OpenStandia/SSO&IDM」詳細ページ> http://openstandia.jp/solution/id_management/

また、一般的なWebサイト向けにGoogleから提供されているガジェットを利用すれば、パーソナライズ機能で画面の統合も可能です。ユーザごとにページレイアウトを自由にカスタマイズでき、GoogleAppsガジェットを自由自在に配置することができます。

★まとめ:

下記のページから、OpenStandia/Portalのご紹介資料(PDF) をダウンロードできます。

よろしければこちらのシステム連携概要図(P.27)もご覧ください。

http://openstandia.jp/solution/eip/?mm=61_4

OpenStandia営業マネージャー まるコン担当 谷川

6. 編集後記サマリー

6. 編集後記サマリー

2012.03.22発行 Vol.55

最後までご覧いただき、どうもありがとうございました。

先日購入した低速ジューサーを活用してニンジン生ジュースを楽しんでいます。

このニンジンの絞りカスでスコーンを作るのがマイブームで、これがまたおいしくてつい食べ過ぎてしまいます。健康にいいものでも食べ過ぎては意味がないですね。最近話題の長寿遺伝子は、カロリーを抑えることでスイッチが入るそうです。やはり、腹八分目&カロリー控えめが大事なんですね。実践するのはなかなか難しいですけど…。次回もどうぞよろしく願いいたします。

2012.04.25発行 Vol.56

最後までご覧いただき、どうもありがとうございました。

私の住む地区は保育園激戦区。認可保育園に入れるのが難しいため、1年前から認可外の保育園をいろいろ探してきましたが、先日ようやく第一希望の保育園に入園できることになりました。認可外といえどもかなりの人数が順番待ちで、運よく娘の年齢のクラスに空きが出て入れた次第です。

女性の就労支援と少子化対策には抜本的な改革が必要と何年も前から言われていながら、平成23年10月1日現在の保育所入所待機児童数は全国で46,620人！末端の保育現場はまだこんなに厳しい状況です。

つい先日、2030年代以降の日本の経済成長率がマイナスに陥り、1人当たりのGDPで見ると、先進国から転落しかねないという試算が経団連のシンクタンクから発表されました。その対策のひとつとして、また女性就業率を上げるべきという議論になりますが、まずは保育所を増やすことが先ですよ！保育所として建てた施設であれば、使い終わった後も、学童保育やグループホーム等の用途で使えそうですね。企業側の育児支援制度の充実も期待したいですが、まずは国・地方自治体が保育所増設に力を入れるべきです！と、いつになく熱くなってしまいました(^_^;)次回もどうぞよろしく願いいたします。

2012.05.24発行 Vol.57

最後までご覧いただき、どうもありがとうございました。

5月から、全10回 初心者向けお試しレッスンの「バレエ」を始めてみました。

日常生活での姿勢を良くしたくて気軽な気持ちで参加したのですが、かなり難しいです。「足だけで体を動かしたり支えようとすると、ふくらはぎがたくましくなるよ！体全体を使って！」と注意されながら、必死にポーズを取っています(笑)バレエでは、常に体の軸を意識して、後ろ太ももとお尻に力を入れることが大事だそうですよ。緊張感と筋力がない私の太ももとお尻でどこまでついていけるのか不安はありますが…あと7回、頑張りたいと思います。

そういえば一昨日、世界一の電波塔スカイツリーがオープンしました。スリムなイメージですが、やはり間近で見ると大きいですね！先日のレディー・ガガさんのスカイツリー訪問が世界に向けた大きなアピールになりましたし、墨田区の試算 年間約1746億円超を上回る経済効果があるといいですね。

http://www.mlit.go.jp/kankocho/topics08_000083.html

次回もどうぞよろしく願いいたします。

6. 編集後記サマリー

2012.06.27発行 Vol.58

最後までご覧いただき、どうもありがとうございました。

ファーストサーバの大規模障害で、データ消失された企業、個人の方々はシステム復旧に大変な御苦勞をされたことと存じます。お客様のシステム構築や、運用・保守を実施している我々も、このような重大な障害を起こさないよう、今後より一層気を引き締めてまいる所存です。

さて、話は変わって、先日久しぶりにOSSコミュニティ主催のイベントに参加させていただきました。(XOOPS Cube Conference 2012、100人規模のイベントでした) 平日仕事をされているコミュニティメンバーの方々イベントを企画し、当日の運営、講演をされていましたが、その作業の大変さを知っているだけに、「仕事ではないのに、こんなにも情熱を注ぐことができるOSSコミュニティの人達は本当にすごい！」と改めて感服いたしました。

これまでは、「私がOpenStandiaの仕事を通じてOSSを広めることで、間接的にコミュニティの皆さんを応援しているんだ」という気持ちでいました。でも、最近では、「私にももっと直接的に応援できることがあるかな？」と考えるようになりました。でも、自分に何ができるか、分からないのです。「私市さんにこうしてもらったら助かるなぁ」「こういう支援があるとうれしいなぁ」ということや、ご意見などがございましたら ossc@nri.co.jp にお知らせいただければ幸いです。できることを少しずつ実践していきたいと思えます。

次回もどうぞよろしく願いいたします。

2012.07.24発行 Vol.59

最後までご覧いただき、どうもありがとうございました。

高校野球にロンドンオリンピックと、スポーツの見どころ満載の夏がやってきました。最近、体を動かす楽しみを再発見したので、選手たちの熱いプレーを見て元気をもらいながら、トレーニングを頑張りたと思っています。

そういえば、8月7日に発表されると報じられたiPhone5は、結局秋(10月頃?)の発売になるそうですね。NTT Docomoのスマホに替えてまだ1年なのに、早くもiPhoneに戻したい気持ちがムクムクと...(笑)。我が家はSoftBankの電波の入りが悪かったから次回はau? にしようかな、家族でキャリアが違おうと面倒かな?なんて考えながら、新発売を心待ちにしています! 次回もどうぞよろしく願いいたします。

2012.08.23発行 Vol.60

最後までご覧いただき、どうもありがとうございました。

2012年夏のロンドン五輪、皆さんはどの競技が印象に残りましたか?

いろいろな事件も起こって、飛び火していますが...

ともかく日本の獲得メダル個数は、五輪史上 過去最高!

日本選手全員に心から盛大な拍手を贈り、称えずにはいられません。

それにしても、スポーツ選手、特に国を代表してオリンピックに出場する選手たちを、競技前の段階からも高く評価してもっともっと優遇すべきだと思いませんか。

少なくとも選手全員が往復で飛行機のビジネスクラスに乗れるようにしてあげてほしいなぁ。

文部科学省スポーツ予算増額のオンライン署名があったら絶対に参加するのに、と思う今日この頃です。

次回もどうぞよろしく願いいたします。

6. 編集後記サマリー

2012.09.26発行 Vol.61

最後までご覧いただき、どうもありがとうございました。

今年の夏は、長く暑い夏でした。秋分の日が近づいた頃から急に秋らしくなってきた、厳しかった残暑もこれで終わったのだとホッとしています。

そして、これからイベントに参加するにはとても良い時期になりますね。OpenStandiaでも10月は数多くのイベントを予定していますが、目玉は、やはりITPro EXPO2012内での「NRI Insights 2012」です。

10/10のNRI嵯峨野執行役員による特別講演、10/12の1Dayセミナー(7講演)でIT基盤サービス最新事例をご紹介します。1Dayセミナーは、まさに一日でいろんな講演を効率よく聞けます。是非、ご来場くださいませ。

話変わって、9/10からの遅めの夏休みを取って、伊豆下田市の海に行ってきました。白浜中央、鍋田浜、外浦、九十浜と、海水浴場めぐり。照りつける日差しは夏そのものでしたが、人がほとんどいなくて、プライベートビーチ感覚でゆったりしました。道路も、地魚料理のお店も空いている、9月&平日の下田は穴場でしたね。次回もどうぞよろしく願いいたします。

2012.10.25発行 Vol.62

最後までご覧いただき、どうもありがとうございました。

先日、仕事用のノートPCを新調しました！レッツノートN (CF-NX1)です。サクサク動いて本当に快適です。購入したのはMicrosoft Officeが入っていないモデルなので、現在は「LibreOffice」、「Apache OpenOffice」の両方を使って業務を行っています。

先日10月18日(米国時間)に、Apache Software Foundationより、これまでインキュベータプロジェクトとして扱っていた「Apache OpenOffice」をトッププロジェクトレベルに引き上げたことが発表されました。

2013年の第1及び第3四半期の公開に向けて、新機能を追加したメジャーリリースの作業も開始されたそうです。

企業買収によって先行きが危ぶまれた「Apache OpenOffice」ですが、これを弾みに、より充実した機能拡張を進めていってほしいですね。名実ともに世界トップのOSSコミュニティ、Apache Software Foundationが抱えるトッププロジェクトは100以上もあります。OpenStandiaでは、企業システムでよく使われるOSSのサポートラインナップが、現在およそ60種類。お客様のご要望に合わせて、今後も取り扱いOSSを拡大してまいります！

次回もどうぞよろしく願いいたします。

2012.11.22発行 Vol.63

最後までご覧いただき、どうもありがとうございました。

またまた先日購入したノートPCのお話。モバイルPCなので、ハードディスクの暗号化をしたいと思い、「TrueCrypt」というOSSの暗号化仮想ドライブソフトウェアを教えてもらって、インストールしました。

<http://www.truecrypt.org/>

あまり技術的なことに詳しくない私でも、マニュアルを見ながら一人でインストールできました。こんな分野でも、OSSの製品があって、しかも無料で使えるなんてと、正直言って驚きました。OSSのラインナップは本当に幅広いですね！

さて、今月は3歳になる私の娘の七五三を予定していましたが、日程調整やいろいろな予約のスタートダッシュが遅れて12/1(土)に参拝&写真撮影をすることにしました。家の中でも被布を着て、お出かけの日を楽しみにしている娘の姿は本当にかわいいです。これからも健やかに成長していってほしいと願うばかりです。次回もどうぞよろしく願いいたします。

6. 編集後記サマリー

2012.12.25発行 Vol.64

最後までご覧いただき、どうもありがとうございました。

先日12/3の日本情報産業新聞 朝刊5面に、オープンソースのERP製品である「ADempiere」のサポートを実施している企業として、NRI OpenStandiaが紹介されました。その記事の中で、「ADempiereはジェネリック医薬品に近いもの。ある程度同じ機能を持つ安価なOSS ERPが選択肢の一つに。」という、日本ADempiereの会 田口さんの言葉が印象的でした。

確かに、ジェネリック医薬品のように、ある機能を持った商用製品が先行して販売されていて、同機能のOSS製品が後継で開発されるといったケースが多いと思います。しかし、OSS製品の中には、商用製品にはない機能や商用製品よりも優れた機能を持つ製品も数多くあります。商用製品では、開発の人件費などの投資費用がライセンス費とサポート費に転嫁されていますが、OSS製品では、ライセンス費用が無料のものがほとんどです。コスト削減効果が高く、商用製品をしのぐ性能、安定性、シェアを持ち、世界中で広く使われているような成功したOSS製品をもっともっとお客様に認知していただくことが、私たちのミッションなんだ、と改めて感じた出来事でした。 http://openstandia.jp/oss/?mm=64_6

次回もどうぞよろしくお願いいいたします。

2013. 1.23発行 Vol.65

最後までご覧いただき、どうもありがとうございました。

今年は、仕事始めの1/7(月)に、3ヶ月前に購入したばかりのノートPCのハードディスクが壊れるという、さんざんなスタートでした。日本製の家電はそうめったに壊れることはない、という気持ちでいました。甘かったです…。フリーソフトをインストールし直すために、exeファイルのダウンロードページを見ていましたが、最近は必要ないファイルのダウンロードボタン(広告)の方が目立っていて、間違えてそちらのボタンをクリックしてしまいそうになりました。そういえば、過去には、フリーソフトのダウンロードの際に、あまり意識しないで某検索エンジンの提供するポータルサイトまでインストールしてしまい、ブラウザのホームページが書き換わってしまって気味が悪いと思ったこともありました。IT業界に長くいるのに、何をやっているの～！とご叱責を受けそうです(笑)でも、どちらのケースも、インストールしてもらうためとはいえ、やり方に無理やり感がありますよね！フリーソフトのダウンロードは自己責任とはいえ、PC操作に不慣れな方の場合、こういった不要なソフトをインストールしてしまって、困ることも多いのだらうなあと思います。

第三者の客観的な評価をチェックする、信頼できるサイトからダウンロードという基本動作が大事ですね。まだまだ空気の乾燥が続いています。皆様、風邪を引かれませんかようご自愛くださいませ。

次回もどうぞよろしくお願いいいたします。

- OpenStandiaは、「攻めのIT」を支援します。
- オープンソースのことなら、なんでもご相談ください！

オープンソースまるごと



- メールマガジン配信の申し込みはこちら ●
<https://openstandia.jp/mm/mailmag.html>
- バックナンバーはこちらでご覧いただけます ●
<http://openstandia.jp/mm/index.html>



お問い合わせは、NRIオープンソースソリューションセンターへ



osscc@nri.co.jp



<http://openstandia.io/>