XOOPS オープンカンファレンス

第4回 XOOPS イベント

The XOOPS Open Conference

2005 in Tokyo

配布資料

2005 年5月 21 日

XOOPS オープンカンファレンス発表用資料

「XOOPS のセキュリティ」～情報赤貧のススメ

後藤峰陽 (GIJOE)

2005/5/21

序論 Web アプリケーションを取り巻く環境

価格.com クラッキング事件
http://www.kakaku.com/ 「最高レベルのセキュリティが破られた」
http://www.itmedia.co.jp/enterprise/articles/0505/16/news077.html (IT Media)
AWStats 脆弱性による XOOPS 本家サイト改竄事件
http://www.itmedia.co.jp/enterprise/articles/0502/03/news015.html (IT Media)
Agenda-X の脆弱性による sourceforge.jp クラッキング
http://www.itmedia.co.jp/enterprise/0402/13/epn04.html (IT Media)
水桶の底理論 「Web アプリケーションは水桶の底板。どこかに穴があれば、すべての水は流出する」

本論 攻撃者とは？ その手口は？

Script Kiddie～無差別クラッカー～愉快犯 ←対極→ 特定サイトを狙う攻撃
ハニーネットプロジェクト http://project.honeynet.org/
http://book.mycom.co.jp/book/4-8399-1648-9/4-8399-1648-9.shtml
Kiddie 共の武器（アイテム）一覧
・XSS (Cross Site Scripting) 経由のセッションハイジャック
・SQL Injection 経由の各種 DB 内情報引き出し
・様々なルートを使ったリモートコマンド実行 他多数
攻撃側はどこを狙ってもよい
・同一サーバに存在する別の Web アプリケーション経由の攻撃 AWStats, phpMyAdmin, phpBB etc.
・より下層のソフトウエアに存在する脆弱性経由の攻撃 PHP, Apache, MySQL, Kernel etc.

結論 「持たない」ことのススメ

厳然たる事実 「防御よりも攻撃の方がはるかにたやすい」
攻撃を受けた・クラッキングされた場合の被害を最小限にする努力

個人情報保護法（抜粋）
個人情報の保護に関する法律第二条 この法律において「個人情報」とは、生存する個人に関する情報であって、当該情報に含まれる氏名、生年月日その他の記述等により特定の個人を識別することができるもの（他の情報と容易に照合することができ、それにより特定の個人を識別することができることとなるものを含む。）をいう。
法律施行令 3 個人情報取扱事業者から除外される者
その事業の用に供する個人情報データベース等を構成する個人情報によって識別される特定の個人の数の合計が過去６月以内のいずれの日においても５千を超えない者

最終結論 「必要のない情報は持たない」
白扇モジュール http://xoops.suin.jp/

ITMedia エンタープライズ 2005/05/19 公開記事
http://www.itmedia.co.jp/enterprise/articles/0505/19/news001.html

月刊「XOOPS コミュニティ通信」―5月号：
国内 OSS コミュニティーの新たな決断―日本独自を歩み出す XOOPS

国内コミュニティーの在り方を問う決断が XOOPS 日本公式サイトで行われた。日本人が開発当初から牽引してきた XOOPS は、主導権もある存在として、そして優れた国内有志の開発者に支えられてきた。その決断とは？

みなさんこんにちは。4 月には 2 冊の XOOPS 関連書籍（「XOOPS によるポータルサイト構築」「Customizing XOOPS」）が新たに発売され、これで XOOPS 関連書籍は 8 種類になりました。台湾で初めて XOOPS 関連書籍が発売された当時「日本ではまだ本は出ないのか」と長いこと期待されたことを考えると隔世の感があります。

さて、4 月には XOOPS 開発体制に大きな動きがありました。公式発表としてまとめ上げるために記事公開が遅れてしまいました。その分、今回の XOOPS コミュニティ通信は、いっそう内容が盛りだくさんです。

これまで全世界を通してひとつのチームで開発を行ってきた XOOPS ですが、今後は日本独自の開発チームで開発を行うことが決まりました。5 月号は、この情報を中心として解説していきたいと思います。

新体制に向けて XOOPS 開発の歴史を振り返る

まず、XOOPS 開発のこれまでの流れを概説します。

今でこそ「XOOPS」という名前はずいぶんと広まってきましたが、この名前が世の中に登場したのは 2002 年の初頭でした。

PHP-Nuke からのフォークという形で始まり、onokazu 氏（日本）を中心とした、南京、台湾、アルゼンチン、ノルウェー各国の 5 人の開発メンバーで「XOOPS」の開発がスタートしました。現在のバージョン命名ルールで言うと、「XOOPS バージョン 1」系列です。

当初より英語サイトを中心に、各国語サイトとコミュニティーが次々と立ち上がり、各国有志により各国言語での XOOPS サポートが行われました。英語サイト（「本家」サイトと称する）および日本サイト（「日本公式」サイトと称する）の代表を onokazu 氏が務めていました。

2003 年 4 月ごろ、XOOPS バージョン 2（XOOPS 2）の正式公開と時期を同じくして、英語サイトに集まる人たちで「より XOOPS を発展させるための体制作り」が検討され始めました。さまざまな担当ごとにチームが作られたのです。コアチーム、テーマデザインチーム、品質コントロールチーム、文書チームなどです。

身の丈に合わない開発体制から決別する真意

しかし、コントロールの役割を担う部分が不十分なまま組織が大きくなったことが、疑問視の 1 つとなりました。

意志決定の遅れが目立つようになり、そのほとんどが日本から報告されていたという実情もあります。比較的大きなものでは、セキュリティー修正情報が取り込まれるにも日を要するような状態が続いていました。また、「日本語（2 バイト文字）処理に強い」が特長のひとつであった XOOPS で、2 バイト文字を考慮しない変更が加えられることも頻発するなど、日本の XOOPS ユーザに不利益な状況も生まれてきたのです。

開発体制変更以前、XOOPS は特にセキュリティー上の問題が発覚した後の対応の早さには目を見張るものがあったのです。このため、大きな後退といえます。

英語圏主導で進められた「大組織化」は、こと XOOPS の開発においては結果として失敗だったと判断せざるを得ないでしょう。必要であるならチーフ開発者である onokazu 氏の独断で新バージョンをリリースすればよいと思われるかもしれませんが、「開発体制の整備」の名の元、同氏にはバージョン番号を付けてリリースする権限がなくなっている状態でした。

日本独自の新開発チームに聞く Q＆A

このような経緯により、XOOPS が元から持っていた「迅速な開発」を取り戻すことが、改めてチーフ開発者である onokazu 氏を中心とした開発チームを再編成する目的となりました。

「日本版 XOOPS」です。

以下、要点を一問一答形式でまとめてみました。

Q：開発メンバーは？
A：onokazu 氏を中心に、minahito、nobunobu、Ryuji の各氏。すべて日本人。

Q：目指すところは日本ローカルの XOOPS か？
A：日本人開発者を中心としているが、日本語専用の XOOPS ではない。開発するのは「日本発の」国際対応 XOOPS である。

Q：2 バイト文字対応は？
A：もちろん 2 バイト文字(特に日本語) 環境で問題なく動作することが最優先の条件。

Q：「日本語版 XOOPS」と考えて良いか？
A：日本語処理のみをターゲットにしているわけではないため、「日本語版 XOOPS」ではなく「日本版 XOOPS」である。

Q：英語版との互換性は？
A：バージョン 2.x.x のうちは、モジュール、テーマセットの互換性を保っていく予定。それ以降のバージョンは未定だが、なるべく保っていきたいと考えている。

Q：現時点で英語版との違いは？
A：トークンシステムの実装方法に一部違いがある。ただし、日本版 XOOPS には互換性コードを加えているので、モジュールから使う場合には、英語版 XOOPS 用に開発されたモジュールを日本版でも使用可能である。逆に、日本版 XOOPS 専用に作られたモジュールは日本版でのみ動作保証の可能性が高い。

Q：今後の「日本版 XOOPS」のロードマップは？
A：当面、バージョン 2.0.x の開発継続を行う。その後の詳細については現在検討中だが、ミッション別に 3 系統開発バージョンの平行開発を行うプランもある。

Q：結局 XOOPS ユーザにとって何のメリットがあるのか？
A：さまざまなメリットがある。セキュリティパッチ提供の迅速化やフィードバックなど意志決定の高速化、また、日本人中心というチーム環境を生かすことで、より日本人になじみやすいユーザーインタフェース改善なども着手できればと考えている。

この移行が XOOPS 日本公式サイトで発表された内容は、記事最後の関連リンクにまとめました。

[坂井 恵, ITmedia]

関連リンク
・XOOPS 日本公式サイト
http://jp.xoops.org/

・本家 2.0.10 リリースおよび日本版 XOOPS について (XOOPS 日本公式サイト：ニュース)
http://jp.xoops.org/modules/news/article.php?storyid=292

・日本離脱が話題になっとります (XOOPS 日本公式サイト：フォーラム投稿)
http://jp.xoops.org/modules/newbb/viewtopic.php?forum=19&topic_id=8624

・ITmedia ウェブデベロッパー dev blog/CMS
http://www.itmedia.co.jp/developer/

協力：ITmedia エンタープライズ
http://www.itmedia.co.jp/enterprise/

本家版 Xoops 2.2 Development Roadmap　By Jan "Mithrandir" Pedersen

Introduction

The XOOPS 2.0.x development team, lead by Onokazu, set a high standard that is flexible and adaptable to a wide range of uses. Thus, 3rd party developers and designers can work on extensions and implementations, while the development team works on making the core fast and secure, and as stable as possible. But there has been a feature freeze on the XOOPS core since the first XOOPS 2.0 stable release with core development limited to patches and fixes.

The incremental development of the kind we've had since XOOPS 1.0 is not enough for a healthy open source project. Many successful open source, community powered development projects evolve into better organised, more professional development projects, and it's clear that XOOPS needs a new way of development.

Why

- With the amount of users, who need more functionality and flexibility, we have to evolve while waiting for the revolution that will come with the development of XooSphere with Skalpa as lead developer.

How

A publicly available CVS branch is available from the XOOPS CVS repository and nightly archives are available from www.xoops.org

Module developers at dev.xoops.org will also be consulted on items such as usability, processes and everyone is welcome to contribute with improvements where they see fit.

Who

The lead developer on the project is Mithrandir, helped by Marcan, phppp and Herve, who will take advantage of submitted hacks, patches and the help from the XOOPS community in general.

Which

Which features will be developed are not entirely frozen at present, but a general focus is on these features:

- Implementation of dynamic user profiles
- Enhancements to the Private Message system
- LDAP/CAS Authentication
- Theme-ability and a completely new look for the administration area
- Installation process re-developed to allow for module installation during XOOPS install
- Many bugfixes, patches and good ideas submitted in the past by the XOOPS community

When

The development team aims for a release of a stable XOOPS version 2.2 by June 30th.

This document is available in a more extensive form at www.xoops.org

---

本家版 Xoops 2.2 Development Roadmap (和訳)

Introduction

Onokazu 氏を中心とした XOOPS 2.0.x の開発チームは、XOOPS を柔軟性に富み広範囲の用途に適用できる、水準の高い XOOPS を提供してきました。このことにより、開発チームが XOOPS のコアの動作スピードやセキュリティと安定性の向上を図りつつ、様々な開発者やデザイナーが機能の拡張・実装を図ることが出来るようになったのです。しかしながら XOOPS 2.0 の安定版のリリース以降は、セキュリティパッチの提供とバグフィックスにコア開発は限定されてきました。

XOOPS1.0 以降に行われてきた漸進的な開発は、健全なオープンソースプロジェクトしては不十分なものでした。たくさんのオープンソースコミュニティがプロフェッショナルなプロジェクトとして組織化に成功していく中で、XOOPS も新しい開発体制を必要としていたのです。

Why

Skalpa 氏(チーム)による革新的な XoopSphere の開発を待つ間に、私たちは、より多くの機能と柔軟性を求める多くのユーザーと共に、XOOPS を発展させて行かなければなりません。

How

一般向けに公開された CVS を XOOPS CVS repository から利用することができ、毎晩アーカイブされたファイルを www.xoops.org から得ることが出来ます。

モジュール開発者は dev.xoops.org にて、ユーザービリティ、プロセス(処理)について相談を受けることが出来て、適当と思われる改良への貢献であれば、どなたでも歓迎されます。

Who

Mithrandir を主開発者とし、Marcan 氏・phppp 氏・Herve 氏の助けも借りながら、XOOPS のコミュニティから寄せられたハックやパッチ、助言の良いところを取り込んで行く予定です。

Which

現在、どの様な機能強化を図っていくかの方針は完全には固まっていませんが、大まかな重点項目は以下の通りです。

- **ユーザ情報の柔軟な項目管理**
- **プライベートメッセージ(PM)の機能拡張**
- **LDAP/CAS 認証**
- **管理者画面デザインの一新、テーマの適用**
- **XOOPS 初期インストール時にモジュールのインストールを出来るようにする機能拡張**
- **XOOPS コミュニティより寄せられた多くのバグフィックスやアイデアの適用**

When

6月 30 日に XOOPS 2.2 のリリースを出来る事を目標に、開発チームは作業中です。

更に詳細な情報は、www.xoops.org から得ることが出来ます。

XOOPS 実践カスタマイズ

Malaika System 早川知道
(Tom_G3X)

Malaika System

2005.5.21

カスタマイズ時の注意点

Malaika System

- コア・モジュールファイルに手を加えない
  - [まず最初] XOOPSやモジュールの管理画面で設定
  - [その次] テーマ、テンプレートでの編集
  - [最後の手段] コア、モジュールをハック

  Smartyを使いこなす事が、高度なカスタマイズの近道
- サイト運用時も考慮したサイト設計
  - 可能な限り、管理画面より変更・修正がベター
  - サイト運用者にやさしく

  サイト運営が煩雑になっては、逆効果

Smartyを使いこなす為には

Malaika System

- Smartyの使い方を理解する
  - 日本語マニュアル http://sunset.freespace.jp/smarty/
- コア、モジュールの標準のSmarty変数を知る
  - テーマで使えるテンプレート変数 http://xoops.sourceforge.jp/wiki/xoops2/
- 独自のSmarty変数を追加する

Smarty変数の独自追加方法

Malaika System

- テーマに直書き方式

```
<{php}>
〜〜PHPコードを記述〜〜
<{/php}>
```

- 別ファイル読み込み方式

```
<{include_php file="$xoops_rootpath/ex_tpl_assign.php"}>
```

- Smartyプラグイン追加作成

表示中のモジュール情報を取得する

Malaika System

- 表示中のモジュール名、モジュールディレクトリ名を取得したい
  - <{$xoops_pagetitle}> は、厳密には表示中のモジュール名では無い
  - 表示中のモジュールディレクトリ名は取得出来ない
- 取得出来ればモジュール毎に異なった表現が可能

表示中のモジュール情報を取得して、テーマに活用してみよう！

表示中のモジュールを取得

Malaika System

- Smarty変数を得る

```
<{php}>
global $xoopsModule;
if ( is_object($xoopsModule) ) {
    $this->assign('ex_module_name', $xoopsModule->getVar('name')); ←モジュール名
    $this->assign('ex_module_dir', $xoopsModule->getVar('dirname')); ←モジュールDir名
}
<{/php}>
```

- 表示例

Smarty変数	表示例1	表示例2
<{$ex_module_name}>	ニュース	●○フォーラム
<{$ex_module_dir}>	news	newbb

表示中のモジュールを取得(記述例)

- **記述例 (表示個所)**

```
<{if $ex_module_dir }>
  <img src="<{$xoops_imageurl}>title_<{$ex_module_dir}>.gif"
    alt="<{$ex_module_name}>" />
  〔モジュール毎に異なった画像を表示させる〕
<{else}>
    〔index.php などXOOPSルートファイルでの表示を記述〕
<{/if}>
```

- **使用例**
  - 桜ケ丘病院 http://www.sakuragaoka-hp.jp/
  - TCP-IP http://yours.tcp-ip.or.jp/

表示中のモジュール情報を取得(使用例)

- **トップページ** (index.phpではFlashを表示)

- **ニュース** (モジュール内では画像を表示)

メインメニューをテーマに

- **最も利用頻度の高いブロックは、「メインメニュー」か？**
- **ブロック以外の場所に配置してみたい**

テーマに取込んで、多彩な表現をしてみよう！

メインメニューをテーマに(ブロック関数)

- **コード**

```
<{php}>
global $xoopsModule;
require_once XOOPS_ROOT_PATH."/modules/system/blocks/system_blocks.php";
$MainMenu = b_system_main_show(); ←〔メインメニューのブロック関数を呼び出し〕
$this->assign( "ex_mainmenu", $MainMenu );  ←〔メインメニューの情報〕

if ( is_object($xoopsModule) ) {
    $this->assign( 'ex_module_name', $xoopsModule->getVar('name') );
    $this->assign( 'ex_module_dir', $xoopsModule->getVar('dirname') );
}
<{/php}>
```

- **同様に、モジュールのブロック関数などを呼び出す事で、応用可能だ。**

メインメニューをテーマに(使い方)

- **メニュー表示**

```
<{foreach item=module from=$ex_mainmenu.modules }>
  <a href="<{$xoops_url}>/modules/<{$module.directory}>/"><{$module.name}></a> |
<{/foreach}>
```

- **サブメニュー表示**

```
<{if $ex_module_dir }>
  <{foreach item=module from=$ex_mainmenu.modules }>
    <{if $module.directory == $ex_module_dir && $module.sublinks }>
      <div class="blockTitle"><{$ex_module_name}> :</div> ←(ブロックタイトル)
      <div class="blockContent"> ←(ブロックコンテンツ)
        <{foreach item=sublink from=$module.sublinks}>
          <a id="mainmenu" href="<{$sublink.url}>"><{$sublink.name}></a>
        <{/foreach}>
      </div>
    <{/if}>
  <{/foreach}>
<{/if}>
```


メインメニューをテーマに(使用例)

- Malaika System http://malaika.s31.xrea.com/

- Sereno http://www.edg.jp/

TinyDをテーマに使う

Malaika System

- テーマとテンプレートの編集方法の違い
  - テンプレート
    - テンプレートマネージャーよりブラウザーで編集可能
  - テーマ
    - FTPなどの方法でアップロードする必要がある

これをTinyDで解決しちゃおう！
しかも、もっと、多彩に使える！！ワイワイ(^o^)ノ

TinyDをテーマに使う(Smartyプラグイン)

Malaika System

- Smartyプラグインを使う
  - 龍司さんが、「TinyD表示プラグイン」を公開されていた。
    ところがその後、TinyDデフォルトのディレクトリ名が変更された為、
    最新のTinyDに、完全には対応できなくなった。
  - 最新のTinyDにも対応出来るように変更してみた。
    一応、龍司さんのバージョンと互換性あり。
    Malaika Systemよりダウンロード出来る。
    /class/smarty/plugins/ に入れておく。
- 使い方

```
<{tinyD dir="(tinyDディレクトリ名)" id="(tinyDコンテンツID)"}>
```

- 記述例

```
<{tinyD dir="tinyd1" id="20"}>
```


TinyDをテーマに使う(使用例:テーマ)

Malaika System

- テーマに適用
  - 更新が楽になる
    - ヘッダーのロゴやバナーなどを頻繁に変更したい
- 使用例
  - TCP-IP http://yours.tcp-ip.or.jp/

TinyDをテーマに使う(使用例:テンプレート)

Malaika System

- テンプレートに適用
  - コンテンツ管理が楽になる
    - 注意書きなどを表示
    - 別のコンテンツを割り込ませる
- 使用例
  - うえこみ春日井小牧xoopsfaq http://www.kasugai-komaki.jp/modules/xoopsfaq/