Кроме того, система управления электронными публикациями отвечает всем современным требованиям по безопасности и предоставляет защиту от основных угроз. Были проведены эксперименты, которые показали устойчивость разработанной системы к перечисленным угрозам и ряду сетевых атак.

Представленная система электронных публикаций является модульной системой, что позволяет с минимальной настройкой использовать проект для создания других подобных комплексов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. М. Дизайн: Управление сайтом [электронный ресурс] / Режим доступа: http://mdesign.ru/publications/cms/40b7504e10e58?start=, дата обращения: 03.12.2010, свободный. – Загл. с экрана.
2. Open Questions Blog Optimizer: CMS и все о них [электронный ресурс] / Режим доступа: http://www.oqbo.ru/read.php?block=25, дата обращения: 03.11.2010, свободный. – Загл. с экрана.
3. *Колисниченко Д.И.* Движок для вашего сайта. – СПб.: БХВ-Петербург, 2008. – 368 с.

Пескова Ольга Юрьевна
Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.
E-mail: poy@tsure.ru.
347928, г. Таганрог, пер. Некрасовский, 44.
Тел.: 88634312018.

Горло Надежда Евгеньевна
E-mail: serebro@list.ru.

Peskova Olga Yur'evna
Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education "Southern Federal University".
E-mail: poy@tsure.ru.
44, Nekrasovskiy, Taganrog, 347928, Russia.
Phone: +78634312018.

Gorlo Nadezda Evgen'evna
E-mail: serebro@list.ru.

УДК 004.732.057

М.Л. Лопатин, О.Ю. Пескова

АНАЛИЗ ЗАЩИЩЕННОСТИ СОВРЕМЕННЫХ СИСТЕМ УПРАВЛЕНИЯ КОНТЕНТОМ

Представлено понятие систем управления контентом (CMS). Рассмотрены основные свободно распространяемые CMS, активно применяемые для создания и управления сайтами: Joomla!, WordPress, Drupal. Рассмотрены их основные характеристики, особое внимание уделено вопросам безопасности, таким, как особенности подсистем безопасности, наиболее критичные уязвимости, возможные атаки на CMS. Приведены рекомендации по настройке и применению рассмотренных систем управления контентом, позволяющие повысить защищенность клиентских систем, построенных на их основе.

Сетевые уязвимости; системы управления контентом (CMS); динамическое управление сайтом.

M.L. Lopatin, O.Yu. Peskova

SECURITY ANALYSIS OF CURRENT SYSTEM CONTENT MANAGEMENT

Presented by the concept of content management systems (CMS). The main open-source CMS, is actively used for creating and managing websites: Joomla!, WordPress, Drupal. Considered their main characteristics, special attention is paid to issues of security, such as a particular security subsystems, the most critical vulnerabilities, possible attacks on the CMS. How to configure and use the considered content management systems, allowing to increase security of client systems based on them.

Network vulnerabilities; content management system (CMS); a dynamic website management.

Введение. В наше время большинство организаций (и множество персон) имеют в глобальной сети Интернет собственные ресурсы. Естественно, стали востребованными средства, обеспечивающие автоматизированное создание, поддержание и развитие web-сайта. Традиционные статические технологии, когда сайт состоял из статических страниц и набора дополнительных специализированных скриптов, не могли эффективно реализовать необходимый функционал и требовали привлечения программистов для любых изменений на сайте.

Content Management Software (CMS) успешно решает эту проблему. В России CMS носит название «система управления содержанием», «система управления контентом» или «контент-менеджер». Часто употребляется более простое название – "движок для сайта". CMS представляет собой программный комплекс, позволяющий автоматизировать процесс управления сайтом в целом и его составляющими: макетами страниц, шаблонами вывода данных, структурой, информационным наполнением, пользователями и правами доступа. CMS разделяют сайты на две составляющие: дизайн (внешний вид сайта в целом, отдельных страниц, конкретных блоков информации) и содержимое (контент). Дизайн сайта, как правило, представляет собой шаблоны и изменяется значительно реже, чем контент (либо вообще не изменяется). Для ввода контента не требуется специальных знаний, а простые приемы оформления текста знает практически каждый, кто работал в текстовых редакторах. При вызове страницы происходит выбор необходимой информации из базы данных и на ее основе генерируется web-страница. Администраторский модуль позволяет вносить изменения в структуру web-сайта и его содержимое.

Несмотря на свои достоинства, именно CMS часто являются причиной получения НСД к данным. Постоянно публикуются всё новые уязвимости популярных CMS, возможность эксплуатации которых ставит под угрозу безопасность всего сервера. Защита системы управления содержимым позволит значительно повысить защищённость сервера от внешних угроз, и наоборот – уязвимости CMS могут свести «на нет» все усилия администратора.

Рассмотрим наиболее распространенные свободно распространяемые системы управления контентом, обращая особое внимание на их системы безопасности.

Joomla! Joomla! – это относительно молодая, но уже очень популярная система управления контентом, которая практически с первой версии успешно соревнуется с CMS Drupal за первенство среди систем управления конентом, не выходя за все эти годы из тройки лидеров. Joomla! представляет собой набор скриптов, написанных на языке программирования PHP с использованием MySQL. В веб-приложениях существуют две части разработки и выполнения кода: серверная и клиентская. К клиентской части относятся HTML, CSS, Javascript, а к серверной ASP, JAVA, PHP и т.д. Для использования серверной части Joomla! (как и другие CMS, в частности Wordpress и Drupal) нуждается в установке локального сервера DENWER, который, фактически, организует собственный хостинг для клиента.

Основные характеристики Joomla! следующие:

- модуль приёма от удалённых авторов новостей, статей и ссылок;
- планировщик состояния материала – расписание контента (публикации любых материалов могут быть ограничены сроками – как начало приема материалов, так и окончание);
- многоуровневое утверждение изменений (модерация публикуемых материалов):
- возможность добавления разделов и тем самими пользователями;
- различные модули (последние новости, счётчик посещений, подробная статистика посещений, гостевая книга, форум, голосование и другие);
- менеджер архива для размещения устаревших материалов;
- модуль безопасности для многоуровневой аутентификации пользователей и администраторов;
- фиксация действий пользователей (ведение лога);
- защита от автоматического заполнения форм (CAPTCHA);
- поддержка безопасного протокола при работе с системой (SSL);
- возможность ограничить доступ к определённым разделам сайта только для зарегистрированных пользователей;
- поиск по сайту.

Использование тех или иных функций зависит от тематики и направленности сайта. Примером системы электронных публикаций на основе Joomla! является электронный журнал «Дискуссия».

В мае 2010 года вышла первая бета-версия Joomla! 1.6, в которой наиболее интересным с точки зрения безопасности изменением стала новая система разграничения доступа пользователей. Теперь администратор может создавать новые группы пользователей, назначать для каждой группы конкретный набор прав, назначать им уровни доступа, делать контент доступным только определенным группам пользователей, управлять уровнями доступа (в том числе и добавлять новые).

В целом же система безопасности Joomla! – это в первую очередь непосредственное обновление и доработка ошибок и дыр самой CMS-системы (а не отдельных приложений). Вопросы безопасности решает отдельная группа лиц-разработчиков – Joomla! Security Center. Официальный и постоянно обновляемый список уязвимостей расширений Joomla! с рекомендациями по их устранению содержится по адресу http://docs.joomla.org/.

Можно выделить следующие наиболее интересные и критичные на данный момент уязвимости.

Joomla! в версии 1.5.10 уязвима для атак CSRF (“Cross-Site Request Forgery” – Межсайтовая подделка запроса). Данный тип атак направлен на имитирование запроса пользователя к стороннему сайту. Атаки позволяют каждому зарегистрировавшемуся пользователю, не имеющему привилегий, получить права Super Administrator – добавлять и удалять пользователя, менять контент, структуру сайта и т.д. Кроме того атака позволяет установить расширения на сайт, т.е. загрузить на сервер вредоносный PHP-скрипт.

Также в системе Joomla! обнаружена уязвимость, с помощью которой злоумышленник может манипулировать данными. Она существует из-за ошибки в рамках XML-RPC (Extensible Markup Language Remote Procedure Call – XML-вызов удалённых процедур). XML-RPC, как и любой другой интерфейс RPC, определяет набор стандартных типов данных и команд в сочетании с плагином Blogger API, который может быть использован для манипулирования или исключения статей. Эта уязвимость присутствует во всех версиях Joomla.

Выявлена еще одна уязвимость, которая ведет к раскрытию данных в Joomla! и позволяет удаленному пользователю выполнить межсайтовый скриптинг (XSS, Cross Site Scripting) – это тип уязвимости интерактивных информационных систем, который возникает, когда в генерируемые сервером страницы по какой-то причине попадают пользовательские скрипты. Уязвимость существует из-за того, что программное обеспечение позволяет злоумышленнику получить доступ к vCard (стандартный формат файлов для обмена электронными визитными карточками) других пользователей с помощью специально сформированного URL.

Кроме того, Joomla! использует уязвимую версию TinyMCE (aTiny Moxiecode Content Editor – платформонезависимый редактор контента). Удаленный пользователь может просмотреть содержимое произвольных файлов на системе и выполнить произвольный код сценария в браузере жертвы в контексте безопасности уязвимого сайта.

Перечислим основные рекомендации по повышению безопасности сайтов, построенных на Joomla!:

1. Необходимо своевременно обновлять Joomla! с официального сайта, поскольку свежие обновления, как правило, закрывают очередной набор найденных уязвимостей. Перед обновлением рекомендуется проверить установленные расширения на совместимость с ним.
2. Поскольку протокол SSL является наиболее надежным и универсальным вариантом защиты конфиденциальности транзакций и проведения безопасной двухсторонней аутентификации пользователей, то настоятельно рекомендуется использовать именно его.
3. Для повышения защищенности учетной записи суперадминистратора рекомендуется сменить его имя – в этом случае при нападении необходимо будет подбирать уже пару значений логин/пароль.
4. Особое внимание нужно уделить защите системных файлов, в частности перенести из каталога public_html критический файл configuration.php, закрыть от записи все содержимое каталога public_html и других важных файлов и каталогов, изменить путь к каталогам logs (лог-файлы) и temp (временные файлы). Для дополнительной безопасности можно использовать файл .htaccess сервера Apache для защиты паролем критических каталогов. Этого, как правило, достаточно для того, чтобы блокировать типичные атаки.
5. Следует удалить все неиспользуемые файлы и каталоги, в том числе и каталог, содержащий файлы при инсталляции. При деинсталляции расширений необходимо проверить, были ли удалены каталоги и файлы, связанные с этим расширением, в том числе таблицы баз данных.
6. Желательно удалить сервер XML-RPC, если его присутствие не обязательно; кроме того, рекомендуется выключить эмуляцию глобальных переменных, поскольку она не используется системой, но гарантирует уязвимость сервера, на котором она установлена.

Wordpress. WordPress – еще одна популярная система управления контентом из свободно распространяемых CMS. Последняя стабильная версия – это 2.9.2, но в апреле 2010 года вышла первая бета-версия 3.0 с рядом важных обновлений. Обычно эту платформу применяют для ведения блогов, но она с успехом используется и на web-проектах другого назначения. Сам по себе Wordpress не очень функционален, поэтому для расширения функциональности используются плагины. Для написания и внедрения плагинов для Wordpress используются язык программирования PHP и база данных MySQL.

Основные возможности:

- поддержка веб-стандартов (XHTML, CSS), технологий RSS, Atom, trackback, pingback;
- многоуровневое утверждение изменений (модерация публикуемых материалов);
- планировщик состояния материала (расписание контента);
- настраиваемые формы обратной связи;
- разграничение доступа (распределение прав);
- поддержка безопасного протокола при работе с системой (SSL);
- поддержка подключаемых модулей – плагинов и тем, позволяющих легко менять как внешний вид, так и способы вывода данных;
- поиск по сайту.

На основе движка WordPress функционируют, например, такие web-сайты, как Открытый каталог научных конференций www.konferent.ru, Конференции международного клуба web-разработчиков www.phpconf.ru.

Для расширения возможности управления правами пользователей и повышения защищенности CMS Wordpress написано много плагинов, из которых хотелось бы порекомендовать следующие.

1. Role Manager – позволяет управлять ролями, настраивать права для ролей, создавать и редактировать полномочия зарегистрированных пользователей, что дает возможность администратору определить уникальный круг задач, выполняемых каждым пользователем.
2. Role Scoper – позволяет контролировать права доступа к возможностям, определяя, каким авторам что позволено делать. Можно создать несколько групп, определив для каждой свои права, и потом рассортировать всех авторов по группам. Можно управлять непосредственно публикациями, внося их в определенные группы.
3. Level2Categories 2 – организует связь между уровнем пользователя и категорией публикаций так, что только пользователи определенного уровня могут публиковать записи в выбранной категории.
4. WP Security Scan – сканирует и проверяет сайт на предмет уязвимостей (неверно выставленных прав, простых паролей и т.д.).
5. WordPress Exploit Scanner – сканирует сайт на предмет эксплойтов. Кроме того, он сканирует установленные плагины на предмет безопасности.
6. ChapSecureLogin – применяется, когда хостинг не позволяет использовать SSL. Он дает возможность использовать для аутентификации протокол Chap.
7. Для обнаружения попыток взлома сайта через подбор пароля администратора можно использовать плагин Last Logins, который отслеживает все попытки входа в административную панель и записывает в лог время, IP-адрес и другие параметры при каждой удачной или неудачной попытке входа.

Для эффективной борьбы со спамом в WordPress входит плагин Akismet. Кроме того, можно создать черный список и фильтры из слов. С их помощью можно просто заблокировать любой комментарий, который содержит запрещенное слово или отправить его на ручную модерацию. Наибольшей же эффективности в борьбе со спамом можно добиться с помощью так называемого капчи (CAPTCHA) – картинки с набором символов, которая позволяет защититься от автоматических спам-роботов. В целом же плагинов для защиты от спама несколько десятков.

Выделим следующие наиболее интересные и критичные на данный момент уязвимости (полный перечень уязвимостей опубликован на сайте www.securitylab.ru).

В апреле 2010 года была организована атака на сотни блогов, созданных под управлением WordPress. Она стала возможной потому, что данные авторизации хранились в базе данных в незашифрованном виде. Кроме того, была использована уязвимость CMS, которая разрешает устанавливать параметры, позволяющие читать файлы конфигурации wp-config.php любому пользователю. Злоумышленники внедряли в блоги вредоносные фреймы iFrame, после чего посетители инфицировались вредоносными программами, в том числе – фальшивыми антивирусами. Большинство блогов работало на последней версии платформы WordPress за номером 2.9.2. Предположительно, хакер написал сканер для обнаружения всех файлов конфигурации, содержащих некорректные настройки, после чего извлек из них пароли для доступа к базе данных и начал взламывать.

Уязвимость Comments Html Spam Vulnerability позволяет удаленному злоумышленнику обойти ограничения безопасности на целевой системе. Уязвимость существует из-за того, что возможно создать два аутентификационных cookie ("wordpressuser" и "wordpresspass") из данных в таблице "users". Использование данного модуля приведет к обходу механизма аутентификации и позволит зайти в систему с правами администратора.

Уязвимость «Нарушение конфиденциальности информации в WordPress» позволяет удаленному злоумышленнику получить несанкционированный доступ к конфиденциальной информации на целевой системе. Уязвимость существует из-за того, что WordPress использует предсказуемые cookies для определения автора комментария. Атакующий может угадать и затем подменить cookies, отправленные оригинальным автором, что позволит ему просматривать комментарии других пользователей. Эксплуатирование уязвимости требует знания имени автора комментария и его почтового адреса.

Множественные уязвимости в Wordpress позволяют удаленному злоумышленнику осуществить DoS-атаку и выполнить произвольный код на целевой системе. Уязвимости возникают из-за ошибки в проверке входных данных.

Можно перечислить следующие основные рекомендации по повышению безопасности сайтов, построенных на WordPress:

1. Первые рекомендации, данные нами в предыдущем разделе, полностью подходят и для данной CMS – необходимо регулярно обновлять систему, использовать протокол SSL для повышения защищенности сетевого трафика и сменить имя администратора (точнее, желательно создать новую учетную запись администратора и удалить старую).
2. Рекомендуется изменить путь к каталогу WordPress, содержащему системные файлы и каталоги – для этого достаточно переименовать каталог и настроить в соответствии с изменениями конфигурационный файл wp-config.php. Кроме того, желательно переместить и сам файл wp-config.php, а также использовать файл .htaccess для защиты его содержимого.
3. Желательно изменить стандартный путь к странице авторизации.
4. Рекомендуется организовать защиту каталогов на сервере от просмотра. Часто можно получить список системного каталога wp-includes, просто добавив его имя к имени файла в командной строке браузера, что небезопасно.
5. Рекомендуется скрывать используемую версию WordPress, поскольку знание версии позволит атакующему более конкретно использовать уязвимости системы, которые в большинстве своем привязаны к версиям. Этого можно добиться, если в файл functions.php добавить код remove_action('wp_head', 'wp_generator').

6. Если нет явной необходимости использовать протокол FTP, то рекомендуется полностью закрыть доступ к сайту по FTP.
7. Следует широко использовать возможности плагинов, в частности, плагины, расширяющие возможности по аутентификации и разграничению доступа пользователей.

Drupal. Drupal – система управления контентом, разработанная на языке программирования PHP. Drupal поддерживает основные базы данных, например MySQL MSSQL., а также стандарты XHTML, CSS, XML и др. В 2009 году Drupal признан лучшей Open Source PHP CMS.

В стандартной поставке обеспечивается следующий набор функций:

- готовые решения типовых задач: поддержка блогов, форумов, новостей, книг и опросов;
- единая категоризация всех видов содержимого (таксономия), вложенность категорий любой глубины;
- механизмы рубрикации: каждый документ сайта может входить в одну или несколько рубрик, сами же рубрики могут составлять списки или сложные иерархические структуры;
- многоуровневое утверждение изменений (модерация публикуемых материалов);
- поддержка XML-форматов: вывод документов в RDF/RSS, аггрегация материалов с других сайтов, BlogAPI для публикации материалов с помощью внешних приложений;
- навигация, группировка и поиск;
- возможность создания сайтов с пересекающимся содержимым (например, общей базой пользователей или общими настройками);
- раздельные конфигурации сайта для различных виртуальных хостов (в том числе собственные наборы модулей и тем оформления для каждого подсайта);
- управление версиями и отслеживание обновлений;
- фиксация действий пользователей (ведение лога);
- защита от автоматического заполнения форм (CAPTCHA);
- поддержка безопасного протокола при работе с системой (SSL);
- поддержка общей авторизации между сайтами на Drupal («сайты-партнёры»).

Разграничение прав доступа основано на присвоении одной или нескольких ролей пользователям, непосредственно права доступа к различным функциям сайта закрепляются за ролями. Подключаемые модули сами определяют, к каким из своих функций дать доступ определённым ролям. Для случаев, когда подобной схемы недостаточно, можно использовать более мощный механизм, основанный на присвоении прав (на просмотр, создание, изменение и удаление) каждому отдельному документу. Но интерфейс для управления этим механизмом в текущей версии CMS отсутствует, для его использования предлагаются дополнительные модули.

Функционал дополняется темами и плагинами. Ограниченный объем статьи не позволяет нам описать хотя бы основные плагины, повышающие защищенность сети, тем более, что для решения каждой задачи доступен выбор иногда из десятков плагинов.

Именно по той причине, что основная нагрузка ложится на дополнения, Drupal часто относят к CMF (Content management framework – каркас системы управления контентом). В роли совмещения концепция CMF и CMS в одном продукте у Drupal есть аналоги, но Drupal проще всего для пользователя. Единствен-

ное, что нужно иметь в виду – при выходе новых версий не гарантируется обратная совместимость API. Сейчас разработчики поддерживают две версии Drupal: текущую (6.x) и предыдущую (5.x).

Целый ряд крупных корпоративных сайтов (например, AOL, NASA, Ubuntu, сайты городов Москва и Санкт-Петербург) работает на базе Drupal, поэтому у разработчиков к безопасности серьезное отношение. Организована специальная команда безопасности Drupal Security Team, которая отслеживает все сообщения о проблемах с безопасностью, анализирует код на предмет возможных уязвимостей, осуществляет поддержку разработчиков дополнительных модулей по вопросам безопасности.

Рассмотрим некоторые уязвимости Drupal, которые используются при внедрении стороннего кода в CMS и в основном представляют собой межсайтовый скриптинг.

Обход ограничений безопасности и нарушение конфиденциальности информации в продуктах Drupal – наиболее острая проблема для защищенности системы. Существующие уязвимости позволяют удаленному злоумышленнику обойти ограничения безопасности и получить доступ к конфиденциальной информации на целевой системе. Уязвимость существует из-за неправильного установления полномочий доступа. Атакующий может использовать модуль Tracker Module и страницу "Recent posts" для получения названия проектов и другой информации о проектах, при условии, что проект или публикация продвинуты на первую страницу.

Уязвимость, существующая из-за ошибки в процедуре проверки комментариев, оставленных пользователями, позволяет удаленному злоумышленнику выполнить произвольный код на целевой системе. Атакующий может передать специально сформированные комментарии, что приведет к выполнению произвольного кода.

Ошибка в тексте функций фильтраций может быть использована для обхода фильтров через недействительные UTF-8 последовательности. Это может быть использовано, чтобы вставить произвольный код сценария, который будет выполнен в браузере пользователя в контексте безопасности сайта.

Модуль aggregator позволяет пользователям выполнять определенные действия с помощью HTTP GET-запросов без выполнения любой проверки достоверности данных для проверки запроса. Это может быть использовано для удаления элементов из определенных каналов, когда пользователь посещает специально созданные страницы.

Заключение. Каждая из рассмотренных CMS-систем имеет свою отличительную особенность: дизайн и гибкость (Drupal), удобство и простор в работе (Joomla), легкий и удобный интерфейс (Worpress). Есть очень важная составляющая в каждой из этих систем – это безопасность администратора CMS-системы и защита данных пользователя. Учитывая специфику безопасности CMS, можно сделать вывод, что наиболее безопасная для пользователя – Drupal, она не дает возможности злоумышленнику полностью манипулировать данными пользователей, ядро системы стабильно и регулярно обновляется, но встроенных функций по защите информации явно недостаточно для построения защищенной многопользовательской системы. Кроме того, гибкость системы с точки зрения безопасности играет свою отрицательную роль: раз пользователи вынуждены для решения практически всех задач использовать плагины, выбирая их из тысяч доступных, то стойкость системы будет определяться стойкостью плагинов (а точнее, стойкостью наиболее слабого из них). Конечно, команда безопасности отслеживает уязвимости ядра и наиболее популярных дополнений, но невозможно обработать все из них, и основная нагрузка по проверке своей системы на уязвимость ложится на конечного пользователя – разработчика сайта либо администратора системы. Наи-

более уязвимая система – Wordpress, она имеет слишком много точек доступа и уязвимостей, как для данных пользователя, так и для защиты пароля суперадминистратора. Joomla – самая передовая и стабильная в плане безопасности из всех представленных систем. Производители данной CMS уделяют немало внимания защиты данных пользователя и администратора уже в базовой поставке системы (особенно в последней версии), и в большинстве случаев для устранения критичных уязвимостей достаточно просто обновить систему. Как и в случае с Drupal, положительную роль играет возможность для пользователей сообщить о найденных уязвимостях и критичных ошибках как в ядре системы, так и в дополнениях, что помогает своевременно их исправлять.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Колисниченко Д.И.* Движок для вашего сайта. – СПб.: БХВ-Петербург, 2008. – 368 с.
2. WordPress: Документация [Электронный ресурс] / Режим доступа: http://4wordpress.ru/?cat=3, дата обращения: 06.03.2010, свободный. – Загл. с экрана.
3. Joomla!: Документация [Электронный ресурс] / Режим доступа: http://www.joomla.ru/documentation.html, дата обращения: 06.03.2010, свободный. – Загл. с экрана.

Пескова Ольга Юрьевна
Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.
E-mail: poy@tsure.ru.
347928, г. Таганрог, пер. Некрасовский, 44.
Тел.: 88634312018.

Лопатин Михаил Леонидович
E-mail: ghostrider_001@mail.ru.

Peskova Olga Yur'evna
Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education "Southern Federal University".
E-mail: poy@tsure.ru.
44, Nekrasovskiy, Taganrog, 347928, Russia.
Phone: +78634312018.

Lopatin Mikhail Leonidovich
E-mail: ghostrider_001@mail.ru.