SOLUTION TO THE INVERSE PROBLEM OF INFORMATION SECURITY ECONOMIC MANAGEMENT

For finding solution to the inverse problem of economic management parameters of information system must be specified and required resource amount and optimal resource allocation between objects must be found. Criterion of optimality is minimization of total loss, which includes loss from information leakage and security costs, or maximization of security investment benefit and its profitability. System of two objects is considered; objects are differed in vulnerability and information quantity. Solution is found in area of saddle point existence, what provides reality of findings, since neither of sides interested in changing their strategy. Fulfilling both conditions – guaranteeing of saddle point state and optimization of resource allocation – is achieved by controlling of information system parameters: object vulnerabilities and information distribution between the objects. The solution to the inverse problem is the first step to system synthesis. Next step is choosing security mechanisms for each object with a glance of their price and threat neutralizing probabilities.

Keywords: information security, mathematical model, vulnerability, optimization, saddle point.

Левченко Євген Григорович, кандидат фізико-математичних наук, доцент кафедри засобів захисту інформації Національного авіаційного університету.
E-mail: ruslana_prus@meta.ua
Левченко Евгений Григорьевич, кандидат физико-математических наук, доцент кафедры средств защиты информации Национального авиационного университета.
Levchenko Evgen, PhD in Physics and Math, Associate Professor of Information Security Equipment Department, National Aviation University (Kyiv, Ukraine).

Прус Руслана Богданівна, аспірант кафедри засобів захисту інформації Національного авіаційного університету.
E-mail: ruslana_prus@meta.ua
Прус Руслана Богдановна, аспирант кафедры средств защиты информации Национального авиационного университета.
Prus Ruslana, postgraduate student Academic Department of Information Security Means, National Aviation University (Kyiv, Ukraine).

УДК 004.056.53

БЕЗОПАСНОСТЬ ПОЛЬЗОВАТЕЛЬСКИХ ПРОЦЕДУР АУТЕНТИФИКАЦИИ WEB-ПРИЛОЖЕНИЙ

Михаил Коломыцев, Светлана Носок, Николай Грайворонский

Интерактивные Web-приложения в настоящее время являются важной частью информационных систем самого разного назначения – бизнеса, государственных структур и других. Основной особенностью таких систем является организация доступа клиентов, деловых партнеров и собственных сотрудников к информационным ресурсам через Интернет. Для доступа к онлайн-услугам, определения уровня полномочий, пользователи должны однозначно идентифицировать себя. Существует множество способов организации процесса аутентификации пользователей, из которых чаще всего используется аутентификация с помощью форм. В статье рассматриваются рекомендации, направленные на повышение безопасности процесса аутентификации и управления сессиями пользователей.
Ключевые слова: *Web-приложения, аутентификация, атака, процес аутентификации, информационная система.*

Вступление. Целью статьи – дать рекомендации разработчикам по повышению безопасности WEB-приложений.

Интерактивные Web-приложения в настоящее время являются важной частью информационных систем самого разного назначения – бизнеса, государственных структур и других. Основной особенностью таких систем является организация доступа клиентов, деловых партнеров и собственных сотрудников к ресурсам системы через Интернет. Для доступа к онлайн-услугам, определения уровня полномочий, пользователи должны однозначно идентифицировать себя. Существует множество способов организации процесса аутентификации пользователей [1,3], из которых чаще всего используется аутентификация с помощью форм [2]. Разработчики понимают важность корректной реализации механизма аутентификации, однако использование собственных разработок для реализации этого механизма зачастую приводит к возникновению уязвимостей. В статье рассматриваются вопросы, связанные с повышением безопасности процесса аутентификации и управления сессиями пользователей.

Встроенные в протокол HTTP схемы аутентификации пригодны для использования [5], особенно если устанавливается защищенное соединение по протоколу SSL. Однако разработчи-

ки зачастую используют не их, а процедуры собственной разработки, стараясь сделать их более защищенными за счет усложнения.

Применяются, в частности, такие методы [4]:

– требование к пользователям более полно идентифицировать себя, указывая не только имя и пароль;

– использование защиты от атак подбора пароля (brute force);

– введение механизма управления пользовательскими сессиями, в частности, их завершение по истечению определенного периода.

Основная часть. *Меры повышения безопасности пользовательских процедур аутентификации.* Классическая HTML форма для доступа к приложению содержит два поля ввода текста. Клиент вводит учетные данные для входа в виде имя пользователя и пароля, а затем передает данные. Web-сервер проверяет полученные данные и, если они верны, разрешает клиенту доступ к ресурсу. Если проверка подлинности дала отрицательный результат, клиенту вновь предлагается ввести имя и пароль.

Даже в такой, распространенной форме аутентификации желательно предусмотреть ряд мер, повышающих безопасность и надежность процесса аутентификации. К таким мерам можно отнести:

– Во всех случаях, когда пересылаются конфиденциальные данные клиента, необходимо использовать защищенный канал. По крайней мере, должен использоваться протокол SSL. Рекомендуется использовать максимально возможный уровень шифрования.

– В случае, если аутентификация клиента закончилась с отрицательным результатом, информация об этом должна возвращаться в самом общем виде (например, "Authentication Failure"), без подробностей. Если выдаются сообщения типа "Пользователь не существует" или "Неверный пароль", то злоумышленник может организовать атаку путем перебора имен пользователей (половина информацию, необходимую для входа в систему) и паролей.

– Задача подбора пароля становится для злоумышленника тривиальной, если не используется механизм блокировки учетных записей. Существуют многочисленные инструменты для атак подбора пароля в HTML-формах. И для организации атаки нет необходимости знать имя конкретного пользователя. Если пользователям разрешается задавать пароли самим, то злоумышленник может перебирать имена учетных записей, а в качестве пароля использовать какое-либо популярное слово (например, “password” или имя учетной записи). Если возможно, необходимо использовать сервис блокировки учетной записи, предоставляемый хостом или сервером аутентификации. Как правило, блокировка настраивается таким образом, что после трех неудачных попыток аутентификации, учетная запись блокируется.

– Рекомендуется, перед отправкой аутентификационных данных, проверить их на стороне клиента. Эта проверка должна использоваться для коррекции непреднамеренных ошибок и уменьшить необходимость в избыточных корректирующих связях клиент-сервер. На стороне сервера должны быть выполнены те же проверки представленных данных. Любые данные, не прошедшие такую двойную проверку, должны рассматриваться как очень подозрительные.

– Для отправки данных клиента разработчики могут использовать два метода: GET и POST. Предпочтительным является использование метода POST. Хотя задача изменение данных клиента для злоумышленника является тривиальной при использовании обоих методов, метод GET требуется от злоумышленника меньше навыков для организации атаки. Кроме того, информация, содержащаяся в отображаемых URL-адресах может быть сохранена на локальном компьютере, а информация об учетных записях использована другим пользователем данного клиентского компьютера. Кроме локального компьютера, эта информация часто сохраняется в журналах регистрации веб-сервера, брандмауэра, прокси-сервера.

Блокировка учетной записи. Для корпоративных сетей, использующих технологию Интранет, реализация механизма блокировки учетной записи является простой задачей, решаемой с помощью операционной системы. Однако, в случае удаленного доступа через Интернет, реализовать такой механизм значительно сложнее. Важно так же, корректно организовать процесс разблокировки учетной записи, опираясь на установленную политику безопасности. К рекомендуемым мерам можно отнести:

– Популярным способом парирования атак подбора пароля является последовательное увеличение временного интервала между моментом неудачной попытки аутентификации и отправкой запроса на повторную аутентификацию (обычно в два раза). В этом случае серверное приложение должно хранить и обрабатывать информацию о количестве неудачных попыток и

величине последнего интервала задержки по времени. Кроме этого, необходимо ограничивать максимально допустимое значение интервала задержки и предусмотреть сброс счетчика например, раз в сутки.

– Если учетная запись автоматически блокируется после определенного количества неудачных попыток, то клиент не должен знать причину блокировки. Таким образом, любые дальнейшие попытки аутентификации приведут к выводу на дисплей одного и того же сообщения об ошибке. Если никаких попыток аутентификации не произошли в течение заданного промежутка времени (например, 1 час), учетная запись может быть автоматически разблокирована.

– Если после того, как учетная запись была заблокирована, но до момента ее разблокировки, будет указана верная аутентификационная информация, клиенту должна быть предоставлена информация о том, что запись заблокирована с инструкцией по ее разблокировки.

– Информация, необходимая для процесса аутентификации не должна быть легко угадываемой. Например, если для аутентификации пользователей портала интернет-банкинга требуется указать номер лицевого счета. В большинстве случаев номера банковского счета выделяются последовательно, и может быть организована атака по блокированию большого количества учетных записей.

– Для Web-сайтов, требующих более строгой аутентификации, чем указание имени и пароля, необходимо организовать процедуру аутентификации в несколько этапов в виде запрос-ответ. Это позволит защититься от многих популярных средств атаки.

– В Web-приложении должна быть реализована возможность отслеживать и регистрировать соединения по IP-адресу клиента. При этом приложение должно определять наличие неудачных попыток аутентификации разных учетных записей с одного и того же IP адреса и предпринимать соответствующие меры. Если позволяют средства защиты внешнего периметра (брандмауэров или роутеров), то нужно динамически блокировать внешние IP адреса.

– Дополнительно, для пользователей, успешно прошедших процедуру аутентификации, можно предоставить возможность просмотра истории неудачных попыток после последней удачной попытки. Такая информация может быть использована для дополнительной защиты учетной записи либо для информирования пользователя о том, что используемые средства обеспечивают защиту от таких атак.

Защита от атак подбора пароля и других автоматизированных атак. Существует множество инструментов для организации атак на сайты, с целью получения доступа к приложению и ресурсам. Многие из этих инструментов используются сложные, автоматизированные процедуры преодоления процессов аутентификации веб-приложений. Для приложений или сайтов, требующих минимального участия службы поддержки и постоянного доступа, или там, где процедуры блокировки учетной записи не могут быть использованы, следует изучить и применить другие меры. К таким мерам можно отнести:

– Мощным способом сдерживания автоматизированных атак, направленных как на подбор паролей, так и на нарушение стабильности веб-приложения, является добавление случайных данных на страницу, используемую для ввода идентификационной информации. Клиент должен указать эти данные как часть процесса аутентификации. Например, потребовать от клиента ввести шестое по порядку слово из выданного ему случайного текста. Усложнит задачу представление случайного текста в формате GIF или JPG. Все это должно сопровождаться случайным изменением шрифта и цвета текста. На сленге программистов такой прием называется «капча» (captcha).

– Одним из ключевых методов, используемых автоматизированными инструментами для атаки, является использование кода ошибки и информации на странице, возвращенной сервером. Хорошим способом избавиться от такой уязвимости является возврат одного и того же ответа на любую неудачную попытку доступа, например, «HTTP 200 OK». В идеале, когда веб-приложение сталкивается с любым некорректным запросом от клиента, неважно успешно прошел проверку подлинности или нет, ответ должен быть один: аннулировать любые ID сессий и cookie, и направить клиента на страницу входа в приложение.

Защита при доступе клиента к общедоступным компьютерам. Если доступ к защищенному приложению возможен с общедоступных компьютеров, таких как интернет-кафе, или других, потенциально незащищенных систем, необходимо предпринять дополнительные меры безопасности. В частности, необходимо предусмотреть защиту от таких угроз, как троянские кони, клавиатурные шпионы, атак перехвата.

К желаемым мерам защиты можно отнести:

– Изменение процедуры ввода пароля. Например, не вводить весь пароль, а указать только некоторые его символы, номера которых сервер генерирует случайным образом. Использование такого приема позволит защититься от клавиатурных шпионов и перехвата.

– По причинам, отмеченным выше, и связанных с кешированием данных, предпочтительным является использование метода POST для пересылки данных. Чтобы гарантировано защититься от записи информации на коммуникационных серверах, в ответ сервера необходимо включить теги рис. 1.

```
Pragma: no-cache
Cache-Control: private, max-age=0, no-cache
Expires: 01/01/99 20:00:00 GMT
```

Рис. 1. Теги

В раздел HEAD страницы должны быть включены теги рис. 2.

```
<meta http-equiv="expires" content="01/01/99
20:00:00 GMT">
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content = "max-
age=0">
<meta http-equiv="cache-control" content = "no-
cache">
<meta http-equiv="cache-control" content = "no-
store">
```

Рис. 2. Теги

Браузер клиента запросит новую копию страницы, если в поле «Expires» будет установлена уже прошедшая дата. Этот метод позволяет защититься от использования кнопки «back button» браузера.

– Современные браузеры могут запоминать данные введенные в поля формы, и впоследствии могут заполнять их автоматически. Необходимо убедиться, что разработчики используют скрипты для автоматической очистки полей ввода формы. Когда браузер клиента открывает страницу, встроенный скрипт должен очищать все поля ввода формы.

Управление сессиями. Основная задача Web-сервера – доставка прикладного контента клиенту. И с этой задачей они справляются отлично. В то же время, протокол HTTP не содержит средств управления клиентскими сессиями (например, проверку того, аутентифицирован пользователь или нет). Следовательно, Web-приложения должны обладать возможностью управлять сессиями пользователей.

Как правило, управление клиентскими сессиями базируется на идентификаторах сессии. При этом ID сессии используется приложением для однозначной идентификации браузера клиента, а подсистема авторизации приложения связывает идентификатор сессии с уровня доступа. Таким образом, если клиент успешно аутентифицирован Web-приложением, идентификатор сессии можно использовать в качестве подтверждения его подлинности, избавляя клиента от необходимости повторно вводить свои данные после каждого запроса страницы.

Разработчики в своем распоряжении имеют три метода доступа к информации, хранящейся в ID сессии:

– ID сессии, содержащейся в URL, который получает приложение через HTTP GET запрос, когда клиент нажимает на ссылки;

– ID сессии, хранящийся в полях формы, которую приложение отображает в браузере. Как правило, такие поля являются скрытыми;

– с помощью cookies.

Предпочтительным способом проверки подлинности клиента является использование *cookies*. Если *cookies* существуют, то каждый раз, когда происходит обращение к какому-либо URL, браузер клиента должен представить определенную информацию из *cookies* как часть HTTP-запроса. Таким образом, *cookies* может быть использован для сохранения хранения информации о браузере клиента для доступа к разным страницам и в разные моменты времени. *Cookies* могут быть постоянными (persistent cookies). Они хранятся на жестких дисках клиентских компьютеров. Место и способ хранения определяются используемым браузером. Кроме того используются сессионные *cookies*. Они хранятся только в оперативной памяти и удаляются при закрытии браузера.

Если процесс аутентификации является достаточно защищенным, то идентификаторы сессии защищены гораздо слабее и могут быть использованы злоумышленником для организации атаки. Один из распространенных методов атак является подбор ID сессии методом перебора. Легкость организации такой атаки зависит от степени уникальности идентификаторов и степени защиты канала связи. К рекомендуемым мерам защиты можно отнести:

– Использование защищенного протокола для передачи идентификатора сессии, например, SSL. Особенно это касается случая, когда ID сессии используется в приложении для идентификации клиента.

– Идентификатор сессии не должен содержать информацию об учетной записи клиента (имя, пароль,…).

– Последовательность создаваемых идентификаторов сессии должна быть не предсказуемой. Очень важно, чтобы для генерации уникального идентификатора сессии использовался криптографически стойкий алгоритм. В идеале ID сессии должен быть случайной величиной. Нельзя использовать линейные алгоритмы, основанные на предсказуемых переменных, таких как дата, время и IP-адрес клиента.

– Идентификатор сессии должен быть достаточно длинным. В этом случае можно быть уверенным, что подбор ID сессии путем перебора или угадыванием не приведет атакующего к успеху. С учетом производительности современных процессоров и полосы пропускания каналов связи, можно рекомендовать длину ID сессии не менее 50 символов.

– Поскольку одним из основных способов подбора ID сессии является их перебор (brute-forcing), очень важно, чтобы приложение могло обнаружить факт такой атаки и парировать ее. Если с некоторого IP-адреса последовала серия запросов с недопустимым идентификатором сессии, приложение должно в течение определенного интервала времени игнорировать любые запросы с данного IP. При этом нужно, чтобы таких некорректных запросов было несколько, поскольку одиночная ошибка может быть вызвана неправильным кэшированием ID.

– Серверное приложение должно корректно управлять сроком действия ID сессии и аннулировать его по истечении срока действия. Идентификатор сессии должен быть активен только в течение ограниченного периода времени, величина которого зависит от типа приложения и ценности информации. В идеале приложение должно отслеживать период бездействия для каждой ID сессии и удалять или отменять идентификатор сессии при превышении определенного порога.

– Подсистема управления идентификаторами сессии должна быть защищенной от атак. Получаемые от клиентов идентификаторы сессии должны тщательно проверяться перед дальнейшей обработкой. Необходимо контролировать, чтобы ID сессии были установленного размера и типа. Например, ID сессии нестандартного размера могут быть использованы для атаки переполнения буфера. Кроме того, идентификатор сессии не должен содержать нестандартную информацию. Например, если идентификатор сессии будет использоваться в серверной базе данных приложения, следует проверить, что он не содержит строк, которые могут быть интерпретированы как расширение запроса Select на языке SQL (SQL injection) либо, могут использоваться для внедрения кода (command injection)).

Выводы. Популярные Web-сервера становятся все более надежными и защищенными. Поэтому злоумышленники будут все чаще стремиться нарушить безопасность сервера через уязвимости приложений. В этом случае, процесс аутентификации следует рассматривать как важнейший рубеж обороны, способный отразить широкий спектр атак.

Существуют различные методы организации процесса аутентификации пользователей. Однако разнообразие условий функционирования и требований к приложению зачастую не позволяют разработчикам реализовать этот механизм корректно. При реализации механизма аутентификации в Web-приложениях необходимо тщательно проанализировать, как требования к доступу пользователей, так и меры защиты от возможных атак. Система защиты должна строиться с обязательным учетом того обстоятельства, что на стороне клиента возможны любые манипуляции с информацией, в обход любых проверок. Все данные, поступившие извне, должны тщательно проверяться серверным приложением как в процессе аутентификации так и при последующей обработке.

ЛИТЕРАТУРА

[1]. Authentication and Session Management on the Web [Электронний ресурс] – Режим доступа. http://www.westpoint.ltd.uk/advisories/Paul_Johnston_GSEC.pdf свободный. –Загл. с экрана.

[2]. Коломыцев М.В. Анализ уязвимостей протоколов аутентификации WEB. [Текст]/ Коломыцев М.В., Носок С.А. // Захист інформації , НАУ, Киев, №3(50), 2012, с.41-45.

[3]. Web Authentication Security [Электронний ресурс] – Режим доступа. http://www.sans.org/ reading_room/whitepapers/webservers/web-authentication-security_1250 свободный. – Загл. с экрана.

[4]. Аутентификация с помощью форм [Электронний ресурс] – Режим доступа. http://professorweb.ru/my/ASP_NET/security/level2/2_1.php свободный. –Загл. с экрана.

[5]. Дайджест аутентификация [Электронний ресурс] – Режим доступа. http://ru.wikipedia.org/wiki/Дайджест_аутентификация свободный. – Загл. с экрана.

REFERENCES

[1]. Authentication and Session Management on the Web :http://www.westpoint.ltd.uk/advisories/Paul_Johnston_GSEC.pdf.

[2]. Kolomytsev M., Nosok S. (2012) «Analysis of vulnerabilities of protocols of authentification of WEB», «Ukrainian Information Security Research Journal», №3(50), pp. 41-45.

[3]. Web Authentication Security: http://www.sans.org/reading_room/whitepapers/webservers/web-authentication-security_1250.

[4]. Forms-Based Authentication http://professorweb.ru/my/ASP_NET/security /level2/2_1.php.

[5]. Digest Access Authentication Elektronniy resource] it is access Mode. http://ru.wikipedia.org/wiki/Digest_access_authentication.

БЕЗПЕКА ПРИЗНАЧЕНИХ ДЛЯ КОРИСТУВАЧА ПРОЦЕДУР АУТЕНТИФІКАЦІЇ WEB-ДОДАТКІВ

Інтерактивні web-додатки в даний час є важливою частиною інформаційних систем різного призначення – бізнесу, державних структур і інших. Основною особливістю таких систем є організація доступу клієнтів, ділових партнерів і власних співробітників до ресурсів системи через інтернет. Для доступу до онлайн-послуг, визначення рівня повноважень, користувачі повинні однозначно ідентифікувати себе. Існує безліч способів організації процесу аутентифікації користувачів, з яких найчастіше використовується аутентифікація за допомогою форм. У статті розглядаються питання, пов'язані з підвищенням безпеки процесу аутентифікації і управління сесіями користувачів.

Ключові слова: Web-додатки, аутентифікація, атака, процес аутентифікації, інформаційна система.

SECURITY USER AUTHENTICATION PROCEDURES OF WEB-APPLICATIONS

Nowadays interactive Web-applications are an essential part of the information systems of business, governmental structures, etc. The main feature of such systems is to organize the client access, business partners and employees to the resources of the Internet. For access to online services and defining the level of authority, users must clearly identify themselves. There are many ways to organize the process of users authentication; the most common way is authentication with the help of forms. In this article the questions connected with increasing security of the authentication process and control user sessions are considered.

Keywords: Web-applications, authentication, attack, authentication process, information system.

Коломыцев Михаил Владимирович, кандидат технических наук, доцент Физико-технического института НТУУ «КПИ».
E-mail: box144a@ukr.net

Коломицев Михайло Володимирович, кандидат технічних наук, доцент Фізико-технічного інституту НТУУ «КПІ».

Kolomytsev Myhailo, PhD, associate professor of Institute of Physics and Technology NTUU «KPI».

Носок Светлана Александровна, кандидат технических наук, доцент Физико-технического института НТУУ «КПИ».
E-mail: svetlana@pti.kpi.net

Носок Світлана Олександрівна, кандидат технічних наук, доцент Фізико-технічного інституту НТУУ «КПІ».

Nosok Svitlana, PhD, associate professor of Institute of Physics and Technology NTUU «KPI».

Грайворонский Николай Владленович, кандидат физико-математических наук, доцент Физико-технического института НТУУ «КПИ».
E-mail:graiv@voliacable.com

Грайворонський Микола Владленович, кандидат фізико-математичних наук, доцент Фізико-технічного інституту НТУУ «КПІ».

Grajvoronskij Mykola, PhD in physics, associate professor of Institute of Physics and Technology NTUU «KPI».