

ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЯ ПО ПОСЛЕДОВАТЕЛЬНОСТИ ОТКРЫВАЕМЫХ ОКОН ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

5

ОБЛАСТЬ ТЕХНИКИ

Настоящее техническое решение относится к области вычислительной техники, в частности, к способу и системе идентификации пользователя по последовательности открываемых окон пользовательского интерфейса.

10 УРОВЕНЬ ТЕХНИКИ

Задача защиты информации от несанкционированного доступа становится всё более актуальной. Один из механизмов несанкционированного доступа к чужим денежным средствам, активно используемый злоумышленниками, подразумевает получение ими учётных данных пользователя (обычно это логин и пароль) и последующее подключение к защищённому ресурсу с использованием этих учётных данных. При этом, вне зависимости от использованного сценария получения доступа к учётным данным (применение кейлоггеров, удалённое подключение к компьютеру, кража не защищённого паролем мобильного устройства, а также другие методы, как и различные их комбинации) с точки зрения системы разграничения доступа, действия злоумышленника неотличимы от действий легитимного пользователя, под чьими учётными данными в систему вошёл злоумышленник.

Из уровня техники известны решения, описывающие идентификации пользователя по поведенческому анализу: US2017032250A1, опубл. 02.02.2017; CN106355450A, опубл. 25.01.2017; CN106202482A, опубл. 07.12.2016.

25 Кроме того, из уровня техники известно решение, RU2649793C1, описывающее способ обнаружения удалённого подключения злоумышленника к компьютеру легитимного пользователя на базе анализа интервалов времени, соответствующих движениям курсора мыши.

Однако, известные из уровня техники решения имеют ограниченную функциональность. В частности, решение, описанное в RU2649793C1, не способно обнаружить злоумышленника, который не пользуется программой удалённого доступа. Многие другие известные решения для реализации требуют значительного количества данных, предварительно размеченных человеком.

Предлагаемый способ выявления злоумышленника, пользующегося чужими учётными данными для доступа к сервису, предусматривающему авторизацию и разграничение доступа, не имеет ограничений по использованному злоумышленником сценарию получения доступа к легитимной учётной записи и не требует ручной разметки данных, используемых для предварительного обучения системы.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

Технической проблемой, на решение которой направлено заявленное техническое решение, является создание компьютерно-реализуемого способа и системы идентификации пользователя по последовательности открываемых окон пользовательского интерфейса, которые охарактеризованы в независимых пунктах формулы. Дополнительные варианты реализации настоящего изобретения представлены в зависимых пунктах изобретения.

Технический результат заключается в автоматической идентификации пользователя по последовательности открываемых окон пользовательского интерфейса.

В предпочтительном варианте реализации заявлен компьютерно-реализуемый способ идентификации пользователя по последовательности открываемых окон пользовательского интерфейса, заключающийся в выполнении этапов, на которых с помощью вычислительного устройства:

- регистрируют учетные данные пользователя в системе;
- присваивают соответствующий учетным данным идентификатор пользователя;
- каждому открываемому пользователем окну пользовательского интерфейса присваивают идентификатор;
- на протяжении всего сеанса работы идентифицированного пользователя фиксируют последовательность открываемых окон пользовательского интерфейса, при этом:
 - сохраняют идентификатор окна, которое в данный момент открыто пользователем, и время открытия данного окна;
 - при каждом переходе к новому окну пользовательского интерфейса сохраняют идентификатор этого окна и время, когда оно было открыто;
- накапливают заранее заданное количество сеансов работы данного пользователя;
- анализируют накопленные данные, а именно выявляют повторяющиеся последовательности посещаемых окон пользовательского интерфейса (паттерны);

- для каждого выявленного паттерна вычисляют набор параметров, характеризующих время, проходящее между переходами данного пользователя от окна к окну пользовательского интерфейса;
- сохраняют заданное количество паттернов данного пользователя и на основе вычисленных для каждого паттерна набора параметров обучают по меньшей мере один классификатор идентифицировать данного пользователя по последовательности посещаемых страниц;
- применяют обученный по меньшей мере один классификатор для последующего подтверждения идентичности пользователя, во время сеансов работы которого был обучен по меньшей мере один классификатор.

В частном варианте окна пользовательского интерфейса являются веб-страницами.

В другом частном варианте из выявленных паттернов дополнительно отбирают такие, длина и количество которых удовлетворяют заранее заданным критериям, и наборы параметров вычисляют только для отобранных паттернов.

- В другом частном варианте отбирают паттерны, имеющие максимально возможную длину, причём их количество должно быть не больше и не меньше заранее заданных значений.

- В другом частном варианте паттернам, содержащим заранее определённые окна пользовательского интерфейса, могут быть присвоены априорные веса, которые учитывают при выборе или обучении по меньшей мере одного классификатора таким образом, чтобы соответствующие паттерны оказывали пропорционально большее или меньшее влияние на итоговое решение по меньшей мере одного классификатора.

- В другом частном варианте в набор параметров, вычисляемых для каждого паттерна, входит время, проходящее между переходами данного пользователя от окна к окну пользовательского интерфейса, и усреднённое по всем сеансам работы данного пользователя, в которых встречается данный паттерн.

В другом частном варианте в набор параметров, вычисляемых для каждого паттерна, дополнительно входит дисперсия, вычисленная как дисперсия случайной величины, для временных интервалов между переходами данного пользователя от окна к окну пользовательского интерфейса.

В другом частном варианте в набор параметров, вычисляемых для каждого паттерна, дополнительно входит частотность данного паттерна в сеансах работы данного пользователя.

- 5 В другом частном варианте частотность учитывают при выборе или обучении по меньшей мере одного классификатора таким образом, чтобы паттерны с большей частотностью оказывали пропорционально большее влияние на итоговое решение по меньшей мере одного классификатора.

В другом частном варианте классификатор обучают методом машинного обучения.

- 10 В другом частном варианте классификатор может быть реализован как графовая вероятностная модель или как SVM-классификатор.

Заявленное решение также осуществляется за счет системы идентификации пользователя по последовательности посещаемых окон пользовательского интерфейса, содержащая:

- блок регистрации учетных данных пользователя в системе;
- системные часы, выполненные с возможностью фиксации времени
- 15 - блок идентификации текущего окна пользовательского интерфейса;
- долговременную память, выполненную с возможностью хранения базы данных;
- вычислительное устройство, выполненное с возможностью выполнения вышеописанного способа.

- 20 В одном из возможных вариантов реализации заявленное решение является составной частью дистанционной системы банковского обслуживания.

В другом из возможных вариантов реализации заявленное решение является составной частью веб-сайта, подразумевающего авторизацию пользователей.

- 25 В другом из возможных вариантов реализации система идентификации пользователя является составной частью многопользовательской компьютерной игры, подразумевающей авторизацию пользователей.

В другом из возможных вариантов реализации система идентификации пользователя является составной частью программно-аппаратного комплекса для совместной работы, подразумевающего авторизацию пользователей.

ОПИСАНИЕ ЧЕРТЕЖЕЙ

Реализация изобретения будет описана в дальнейшем в соответствии с прилагаемыми чертежами, которые представлены для пояснения сути изобретения и никоим образом не ограничивают область изобретения. К заявке прилагаются следующие чертежи:

- 5 Фиг. 1 иллюстрирует компьютерно-реализуемый способ идентификации пользователя по последовательности открываемых окон пользовательского интерфейса;

Фиг. 2 иллюстрирует пример общей схемы компьютерного устройства.

ДЕТАЛЬНОЕ ОПИСАНИЕ ИЗОБРЕТЕНИЯ

- В приведенном ниже подробном описании реализации изобретения приведены
10 многочисленные детали реализации, призванные обеспечить отчетливое понимание настоящего изобретения. Однако, квалифицированному в предметной области специалисту, будет очевидно каким образом можно использовать настоящее изобретение, как с данными деталями реализации, так и без них. В других случаях хорошо известные методы, процедуры и компоненты не были описаны подробно, чтобы не затруднять
15 понимание особенностей настоящего изобретения.

Кроме того, из приведенного изложения будет ясно, что изобретение не ограничивается приведенной реализацией. Многочисленные возможные модификации, изменения, вариации и замены, сохраняющие суть и форму настоящего изобретения, будут очевидными для квалифицированных в предметной области специалистов.

- 20 Настоящее изобретение направлено на обеспечение компьютерно-реализуемого способа и системы идентификации пользователя по последовательности открываемых окон пользовательского интерфейса.

- Способ будет описан на примере системы дистанционного банковского обслуживания (ДБО), однако, этот пример не является ограничивающим. Помимо системы ДБО, на
25 примере которой будет описан способ, описываемый способ может также быть реализован в компьютерных онлайн-играх, программном обеспечении для совместной работы, программном обеспечении интернет-магазинов и т.д.

- Как представлено на Фиг. 1, заявленный компьютерно-реализуемый способ идентификации пользователя по последовательности открываемых окон пользовательского интерфейса
30 (100) реализован следующим образом:

На этапе (101) предварительно регистрируют учетные данные пользователя в системе. Учетные данные пользователя - это имя пользователя и пароль. Это важный компонент обеспечения сетевой безопасности.

5 Далее на этапе (102) присваивают соответствующий учетным данным идентификатор пользователя.

На этапе (103) каждому открываемому пользователем окну пользовательского интерфейса присваивают идентификатор.

10 На этапе (104) на протяжении всего сеанса работы идентифицированного пользователя фиксируют последовательность открываемых окон пользовательского интерфейса, при этом:

- сохраняют идентификатор окна, которое в данный момент открыто пользователем, и время открытия данного окна;
- при каждом переходе к новому окну пользовательского интерфейса сохраняют идентификатор этого окна и время, когда оно было открыто.

15 На этапе (106) накапливают заранее заданное количество сеансов работы данного пользователя, а на этапе (107) анализируют накопленные данные, а именно выявляют повторяющиеся последовательности посещаемых окон пользовательского интерфейса (паттерны).

20 При этом, на этапе (108) для каждого выявленного паттерна вычисляют набор параметров, характеризующих время, проходящее между переходами данного пользователя от окна к окну пользовательского интерфейса.

25 После чего, на этапе (109) сохраняют заданное количество паттернов данного пользователя и на основе вычисленных для каждого паттерна набора параметров обучают по меньшей мере один классификатор идентифицировать данного пользователя по последовательности посещаемых страниц.

И на этапе (110) применяют обученный по меньшей мере один классификатор для последующего подтверждения идентичности пользователя, во время сеансов работы которого был обучен по меньшей мере один классификатор.

30 Описываемый способ может быть реализован в любом варианте сервиса авторизованного удалённого доступа, который подразумевает наличие нескольких окон пользовательского интерфейса либо наличие нескольких рабочих зон в одном сложном окне

пользовательского интерфейса и различных возможных путей переключения между ними (как окнами, так и зонами).

Нижеперечисленные варианты технической реализации даны для примера и также не являются ограничивающими.

- 5 В одном варианте технической реализации способ может быть реализован в виде интернет-программы (скрипта), функционирующей на интернет-сайте банка, предоставляющего услугу ДБО.

- В другом варианте технической реализации способ может быть реализован в виде подсистемы «тонкого клиента» ДБО, например, интернет-приложения, функционирующего на мобильном устройстве пользователя: смартфоне, мобильном телефоне и т.д., и ведущего обмен данными по каналам беспроводной связи с сервером банка, на котором установлена и работает собственно система ДБО.
- 10

- Ещё в одном варианте технической реализации способ может быть реализован в виде подсистемы программно-аппаратного комплекса, функционирующего в «облаке» и обменивающегося данными по беспроводным и проводным каналам с «тонкими клиентами», банкоматами и с процессинговым центром банка.
- 15

- Возможна также реализация описываемого способа в виде распределённой программно-аппаратной системы, находящейся на различных интернет-серверах, например, с целью распределения нагрузки, и обслуживающей одновременно множество каналов ДБО, принадлежащих разным банкам.
- 20

- Ещё в одном варианте способ может быть реализован как часть программы или сервиса для совместной работы, где подразумевается авторизованный доступ пользователей (о таких программах или сервисах см. https://ru.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81%D0%BE%D0%B2%D0%BC%D0%B5%D1%81%D1%82%D0%BD%D0%BE%D0%B9_%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%8B&oldid=80620207). Ещё в одном возможном варианте описываемый способ может быть реализован как часть многопользовательской компьютерной игры, подразумевающей авторизацию пользователей.
- 25
- 30

Вне зависимости от варианта технического выполнения описываемого способа, для его реализации необходимо получение двух идентификаторов: идентификатора пользователя и идентификатора текущего окна пользовательского интерфейса.

- Идентификатор пользователя -- это минимум одно сочетание символов, уникальным образом характеризующее конкретного пользователя. Например, идентификатором может быть хэш-функция от телефонного номера мобильного устройства, с которого осуществляется доступ к системе ДБО. В другом случае это может быть agent_id (внутренний идентификатор клиента, используемый в банковской системе). Ещё в одном случае это может быть номер банковской карты, считанный банкоматом, либо результат биометрической идентификации владельца карты или хэш-функция от такого результата.

Возможен случай, когда идентификатор является функцией от набора сведений о программно-аппаратной конфигурации персонального компьютера, с которого осуществляется доступ к системе ДБО

- Возможен также вариант реализации описываемого способа, при котором идентификатор пользователя генерируется системой, реализующей способ, на основе одного или нескольких идентификаторов аналогичного назначения, поступающих от банковской системы. Генерация такого уникального идентификатора может быть выполнена любым общеизвестным способом, например, хешированием символьной строки, полученной путём конкатенации идентификаторов, полученных из банковской системы.
- Как правило, идентификатор пользователя так или иначе зашифрован на стороне банка; его расшифровка не требуется для реализации описываемого способа, несущественно и то, каким конкретно образом он был сгенерирован. Достаточно, чтобы этот идентификатор был уникальным и стабильно воспроизводимым при каждом сеансе доступа к системе ДБО под учётными данными (логинем и паролем) конкретной учётной записи.
- В дальнейшем для простоты будем обозначать идентификатор пользователя как USER(i), где различное значение i указывает на разных пользователей.

Идентификатор текущего окна пользовательского интерфейса -- это минимум одно сочетание символов, которое однозначно соответствует тому окну пользовательского интерфейса, которое в данный момент открыто пользователем.

- Это окно может представлять собой страницу интернет-сайта, и тогда идентификатор может представлять собой, например, хэш от URL данной страницы или от её заголовка (последовательности символов, помещённой под тэгом <title>). В другом случае это окно

может представлять собой элемент интерфейса программы, работающей на мобильном устройстве или в банкомате, и тогда идентификатором может быть хэш названия данного элемента интерфейса либо иное его обозначение, используемое в программе.

5 Для целей реализации описываемого способа достаточно, чтобы этот идентификатор был уникальным в рамках данной системы ДБО и стабильно воспроизводимым при каждом удачном акте доступа к конкретному окну интерфейса.

Возможен также вариант реализации описываемого способа, при котором идентификатор текущего окна генерируется системой, реализующей способ, на основе одного или нескольких идентификаторов окна, поступающих от банковской системы. Генерация
10 такого уникального идентификатора может быть выполнена любым общеизвестным способом, например, хешированием символьной строки, полученной путём конкатенации идентификаторов, принятых от банковской системы.

В варианте реализации способа, подразумевающим наличие одного сложного окна интерфейса пользователя, имеющего несколько рабочих зон, в роли «идентификатора
15 текущего окна» может выступать фокус ввода, [https://ru.wikipedia.org/w/index.php?title=%D0%A4%D0%BE%D0%BA%D1%83%D1%81_\(%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81\)&stable=1](https://ru.wikipedia.org/w/index.php?title=%D0%A4%D0%BE%D0%BA%D1%83%D1%81_(%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81)&stable=1), т.е. некоторая
20 последовательность служебных символов, порождаемая программой, которой принадлежит сложное окно, и несущая информацию о том, какой из многочисленных элементов интерфейса программы (кнопок, всплывающих списков, радиокнопок, окон ввода текста, и т.д.) в настоящий момент активен и готов к восприятию команд пользователя.

25 Преобразование такой последовательности в унифицированный идентификатор места в пользовательском интерфейсе ДБО может быть выполнено, например, взятием хэш-функции от самой этой последовательности символов или любым другим общеизвестным способом.

Ниже по описанию будем обозначать идентификатор текущего окна как PAGE(j), где
30 различное значение j указывает на разные окна пользовательского интерфейса или разные элементы пользовательского интерфейса в однооконном варианте.

Для реализации описываемого способа в виде распределённой программно-аппаратной системы, находящейся на различных интернет-серверах и обслуживающей одновременно множество каналов ДБО, принадлежащих разным банкам, потребуется также сбор и обработка дополнительного идентификатора, указывающего, из системы какого банка
5 поступили те или иные идентификаторы USER(i); PAGE(j).

Этот дополнительный идентификатор -- обозначим его как BANK(q) -- является статическим, он создаётся один раз при настройке и запуске системы ДБО данного банка. Технически он может представлять собой, например, SSL-сертификат, которым подписаны сайты данного банка, или отдельную символьную строку, например,
10 BANK(q)=SBERBANK, передаваемую ДБО каждого банка. Идентификатор BANK(q) используется для первичной сортировки поступающих в систему сведений и адресации ответных сигналов системы -- банковскому антифрод-департаменту.

Других функций, помимо этих, очевидных и допускающих реализацию любым общеизвестным способом, идентификатор банка BANK(q) не выполняет, поэтому в
15 дальнейшем описании этот идентификатор для простоты изложения опускаем.

Идентификаторы USER(i); PAGE(j) поступают от системы ДБО всегда попарно и одновременно, каждый сопровождается временной меткой (timestamp). Фактически между поступлением первого и второго идентификаторов может проходить какое-то время, обусловленное задержками сигнала в физических линиях связи, используемых системой
20 ДБО, но этим временем в рамках реализации описываемого способа пренебрегаем, считая временем поступления идентификаторов время поступления последнего из них.

Термином сессия (S) или, что-то же самое, сеанс работы данного пользователя, в рамках этого описания будем называть последовательность данных, имеющую вид:

$$S=\{USER(i); PAGE(1); t_1\}, \{USER(i); PAGE(2), t_2\}, \dots \{USER(i); PAGE(N), t_N\} \quad (1)$$

25 где t_1 – временная метка, соответствующая времени открытия пользователем USER(i) окна PAGE(1), t_2 – временная метка, соответствующая времени открытия тем же пользователем окна PAGE(2), и так далее.

Важно, что сессия -- это последовательность пар идентификаторов, в которой идентификатор пользователя в каждой паре один и тот же, т.е. сессия всегда ассоциирована
30 с конкретным пользователем и полностью описывает один сеанс его работы с системой ДБО.

В общем случае, в системе ДБО одновременно существует значительное количество сессий, поскольку одновременно с системой ДБО может работать множество разных пользователей. Общепринятой практикой в банках является запись (сохранение) всех сессий и хранение этих сведений в течение значительного времени (от нескольких месяцев до нескольких лет). В одном из возможных вариантов реализации описываемого способа этот массив информации может быть получен от банка и использован для предварительного обучения системы, реализующей способ.

Очевидно, в массиве данных, охватывающих достаточно длительный временной период (например, месяц), наряду с сессиями легитимных пользователей обязательно будут присутствовать и сессии злоумышленников. Однако, известно, что частотность доступа злоумышленников к системе ДБО крайне незначительна. Легитимные сессии составляют абсолютное большинство сессий, сохранённых за продолжительный период времени.

Поскольку обучение системы по существу представляет собой накопление и статистический анализ информации, наличие небольшого количества нелегитимных сессий не скажется на качестве и результатах обучения.

Система, реализующая способ, сортирует данные, поступающие от системы ДБО либо из обучающего массива информации, по $USER(i)$, и формирует пользовательские сессии. Каждая пара идентификаторов $USER(i); PAGE(j)$, как показано в (1), характеризуется временем поступления, зафиксированным посредством сохранения временной метки t . Поэтому для каждой пары, кроме самой первой пары в каждой сессии, может быть рассчитан временной интервал T между моментом поступления данной пары и моментом поступления предыдущей пары. Этот интервал может быть рассчитан как разность временных меток (timestamps) t , вычисляемая с заданной точностью, например, с точностью до 0.01 секунды. Например, интервал между первой и второй парами идентификаторов в каждом сеансе работы данного пользователя рассчитывается как $T = t_2 - t_1$, и т.д.

Рассчитанные интервалы сохраняют для использования в дальнейших вычислениях.

Последовательность данных (1), записанная для одного конкретного пользователя $USER(i)$, после расчёта временных интервалов T примет вид:

$PAGE(j) \text{ -- } T(j) \text{ -- } PAGE(j+1) \text{ -- } T(j+1) \text{ -- } PAGE(j+2) \text{ -- } T(j+2) \text{ -- } \dots \text{ -- } T_{n-1} \text{ -- } PAGE(j+N); (2)$

где $T(j)$ -- выраженное в секундах и долях секунды (не обязательно целочисленное) время, прошедшее между моментом, когда данный пользователь открыл окно $PAGE(j)$, и моментом, когда этот же пользователь открыл окно $PAGE(j+1)$.

Необходимо отметить, что между самими по себе окнами (например, веб-страницами) PAGE(j), PAGE(j+1) в системе ДБО может не существовать вообще никакой связи, ни логической, например, выраженной в текстовом сообщении просьбы перейти на определённую страницу, ни физической, т.е. гиперссылки, ведущей с одной страницы на другую. Подобные переходы образуются в сессиях ботов, т.е. программ, имитирующих ту или иную активность пользователей ДБО.

Притом для работы системы, реализующей данный способ, не имеет никакого значения, что именно представляют собой окна пользовательского интерфейса PAGE(j) и PAGE(j+1). Они необходимы только в контексте того факта, что пользователь, авторизованный под учётной записью USER(i), сначала открыл окно PAGE(j), а через некоторое рассчитанное время – окно PAGE(j+1).

Таким образом, для каждого USER(i) образуется коллекция сессий (сеансов работы данного пользователя), имеющая следующий вид (номера окон и длительности интервалов между ними в этом примере выбраны произвольно):

15 S1= PAGE(5) - 8 - PAGE(9) - 51- ... - PAGE(2) -6- PAGE(6) - 4 - PAGE(33) - 12 - PAGE(21)
 S2= PAGE(8) - 88 - PAGE(35) - ... - PAGE(6) - 5 - PAGE(33) - 10 - PAGE(21) -31- PAGE(2)
 S3= PAGE(6) - 4 - PAGE(33) - 18 - PAGE(21) - ... - PAGE(54) - 93 - PAGE(5) -11 - PAGE(9)

 Sk= PAGE(94) - 11 - PAGE(62) - 52- ... - **PAGE(5) - 42 - PAGE(9)** - 30 - PAGE(72) (3)

20 Когда общее количество сессий Sk превышает заранее заданный порог (например, становится больше 20), коллекцию сессий анализируют, выявляя паттерны. Термином паттерн в данном случае будем называть повторяющуюся последовательность страниц, наподобие выделенной в (3) последовательности PAGE(6) - PAGE(33) - PAGE(21). Временные интервалы T(j) на этапе выявления паттернов не учитываются.

25 Важно, что как все описанные, так и последующие преобразования данных всегда выполняются именно для данного конкретного пользователя. Сессии из коллекций разных пользователей никогда не анализируют совместно.

Длина паттерна, т.е. то, какое именно количество страниц, открытых последовательно, будет считаться паттерном, выбирается заранее. В одном возможном варианте реализации описываемого способа длина паттерна может быть задана интервалом. Например, если задана длина паттерна от 3 до 6, то показанный выше паттерн PAGE(6) - PAGE(33) -

PAGE(21), имеющий длину 3, подходит под заданный критерий, а паттерн PAGE(5) - PAGE(9), имеющий длину 2, не подходит. В другом варианте реализации длина паттерна может быть заранее задана фиксированной величиной, например, 4.

Ещё в одном возможном варианте реализации длина паттерна может выбираться
5 автоматически в зависимости от количества паттернов, выявляемых для данного пользователя. В этом варианте поиск паттернов начинается, например, при заданной длине 3. Если и когда количество выявленных паттернов для данного пользователя превышает установленный при настройке системы порог (например, паттернов найдено более 10),
10 длину паттерна увеличивают на единицу и повторяют поиск, отбросив ранее найденные короткие паттерны. Такой цикл выполняется до тех пор, пока не будет найдено меньшее пороговой величины количество паттернов, имеющих максимально возможную длину.

Возможен такой вариант реализации, при котором проводится предварительная модификация некоторых найденных паттернов. В ходе модификации некоторым паттернам присваивают дополнительные априорные веса, связанные с назначением окон интерфейса,
15 входящих в состав этих паттернов.

Говоря о «весе паттерна», в данном случае подразумеваем некое наперёд заданное число, ассоциированное с данным паттерном и хранящееся в системе, реализующей описываемый способ.

Например, паттернам, содержащим окно PAGE(m), которое в интерфейсе системы ДБО
20 служит для запроса перевода средств на банковский счёт или на кошелек безналичной платёжной системы, может быть присвоен повышенный вес, например, в 3 раза, превышающий вес других паттернов.

Этот вес будет присваиваться любым паттернам, содержащим окно PAGE(m), вне зависимости от того, в сессиях какого именно пользователя USER(i) найдены данные
25 паттерны.

Аналогично, паттернам, содержащим окно PAGE(x), которое в интерфейсе системы ДБО служит для уплаты коммунальных платежей или налогов, может быть присвоен пониженный вес, например, в 0.3 от веса не модифицированных паттернов.

Указанный вес может использоваться системой, реализующей способ, как на описанном
30 ниже этапе обучения классификаторов, так и на этапе контроля идентичности пользователя. Такой вес имеет смысл чувствительности системы к событию определённого вида. В данном примере нетипичное поведение любого пользователя в высоко рискованном,

содержащем окно PAGE(m), сценарии будет значительно больше “настораживать” систему, чем нетипичное поведение в сценарии уплаты налогов, где вероятность появления злоумышленника, завладевшего чужими учётными данными, стремится к нулю.

5 Для каждого выявленного паттерна вычисляется набор параметров, характеризующих манеру поведения данного пользователя при работе с системой ДБО. Все параметры вычисляются на основании той же коллекции сессий.

Первый параметр представляет собой вектор, который будем называть вектором времени V. Это вектор, размерность которого соответствует количеству временных интервалов внутри данного паттерна. Например, для паттерна PAGE(6) - PAGE(33) - PAGE(21) вектор
10 времени будет двумерным: $V = (T_{cp1}, T_{cp2})$, где T_{cp1} -- рассчитанное по заранее определённому алгоритму характерное для данного пользователя время между открытием окон PAGE(6) и PAGE(33), а T_{cp2} -- между открытием окон PAGE(33) и PAGE(21).

Для вычисления T_{cp1} и T_{cp2} берут все вхождения данного паттерна, найденные во всех сеансах работы данного пользователя, и вычисляют эти усреднённые значения по ранее
15 вычисленным значениям временных интервалов $T(j)$ между страницами PAGE(6) и PAGE(33), а также между страницами PAGE(33) и PAGE(21).

Под усреднением времени, например, T_{cp1} , здесь может пониматься среднее арифметическое от всех времён $T1$ для всех вхождений данного паттерна в коллекцию сессий данного пользователя. В другом варианте реализации может вычисляться среднее
20 геометрическое от всех вхождений данного паттерна. Возможен вариант, при котором вычисляется среднее взвешенное, где в роли веса для каждого временного времени T_i берётся количество страниц, отделяющих данный интервал от начального интервала $T1$ или от конечного интервала данного паттерна. Смысл последнего варианта в том, чтобы наибольшее влияние на T_{cp} оказали интервалы при начале или при окончании сессии
25 данного пользователя.

В другой возможной реализации вектор времени V имеет размерность $2*n$, где n количество временных интервалов внутри данного паттерна. В таком случае вектор времени хранит как рассчитанное значение временного интервала T_{cp} , так и дисперсию D. Дисперсия здесь является показателем того, насколько сильно может варьировать длительность конкретного
30 временного интервала T_{cp} у данного пользователя. Подобный вектор размерности $2*n$ может выглядеть следующим образом: $V = (T_{cp1}, D1, T_{cp2}, D2)$.

Дисперсия временных интервалов D1 может быть вычислена на основании собранной коллекции сессий данного пользователя как дисперсия случайной величины T_{cp1} , либо любым другим общеизвестным способом. Аналогично, дисперсия D2 может быть вычислена как дисперсия случайной величины T_{cp2} .

- 5 Помимо вектора времени для каждого паттерна может также вычисляться частотность, показывающая, насколько часто данный паттерн встречается в сессиях данного пользователя. Частотность равна отношению количества вхождений данного паттерна L к количеству сессий данного пользователя K.

- 10 Этот параметр может быть, как меньше единицы, если паттерн встречается не в каждой сессии, так и больше, если паттерн встречается несколько раз на протяжении каждой сессии.

$$\lambda = L/K \quad (4)$$

Найденные паттерны и вычисленные для каждого из них параметры сохраняют для использования в дальнейших вычислениях.

- 15 Затем для каждого пользователя, используя его коллекцию данных, в первую очередь, набор векторов времени V, обучают одно или несколько решающих правил (классификаторов). Техническая реализация классификатора или классификаторов может быть любой общеизвестной; они могут быть реализованы, например, как графовая вероятностная модель (Random Forest) или как SVM-классификатор.
- 20 Если каким-то паттернам при составлении коллекции паттернов были присвоены повышенные и/или пониженные веса, связанные с назначением одной или нескольких входящих в паттерн окон интерфейса системы ДБО, то эти веса могут учитываться при выборе или обучении классификатора или классификаторов таким образом, чтобы соответствующие паттерны оказывали пропорционально большее или меньшее влияние на
- 25 итоговое решение. Аналогичным образом, в роли повышающего или понижающего коэффициента, на данном этапе может использоваться также частотность.

В одной из возможных реализаций описываемого метода один классификатор может обучаться на рассчитанных значениях усреднённых временных интервалов, а второй – на рассчитанных значениях дисперсии временных интервалов.

- 30 Все вышеописанные действия, начиная с сохранения пар идентификаторов {USER(i), PAGE(j)} и соответствующих им отметок времени t, и заканчивая обучением

классификаторов для данного пользователя, могут выполняться в режиме онлайн, непосредственно в ходе функционирования сервиса ДБО.

- В другом возможном варианте реализации система, реализующая описываемый способ, может проходить предварительное обучение в режиме офлайн. В этом варианте реализации
- 5 система получает для анализа заранее сохранённый сервисом ДБО массив данных, выполняет, как было описано выше, анализ данных, обучает классификаторы для всех пользователей, идентификаторы которых были в поступившем массиве данных. Когда анализ завершён, и классификаторы обучены, система, реализующая данный способ, переводится в режим онлайн и далее функционирует совместно с сервисом ДБО.
- 10 Применение обученного классификатора или классификаторов может выглядеть следующим образом. Система, реализующая способ, сортирует данные, поступающие от сервиса ДБО, по USER(i) и формирует пользовательские сессии (сеансы работы конкретных пользователей). С самого начала сессии, как только получен самый первый USER(i), выбирается и запускается классификатор или классификаторы, обученные на
- 15 коллекции паттернов соответствующего этому USER(i) пользователя. На вход классификатора или классификаторов подаются все данные текущего сеанса работы данного пользователя. Таким образом, система, реализующая описываемый способ, ведёт анализ идентичности пользователя непосредственно в ходе работы этого пользователя с сервисом ДБО.
- 20 Решение классификатора или классификаторов представляет собой численную оценку вероятности того, что под данным USER(i) вход в систему ДБО совершил именно тот пользователь, на коллекции паттернов которого были обучены классификатор или классификаторы. Это решение постоянно обновляется в ходе сеанса работы данного пользователя, т.е. решение классификатора или классификаторов пересматривается при
- 25 поступлении каждой очередной пары {USER(i), PAGE(j)}.

Решения каждого классификатора поступают в антифрод-систему банка, где могут использоваться для генерации тревожных оповещений, имеющих смысл предупреждения о нетипичном поведении конкретного пользователя USER(i), или других сигналов аналогичного назначения.

- 30 В тех вариантах реализации описанного способа, которые не подразумевают наличия антифрод-системы (например, интернет-магазин, веб-сервис для коллективной работы, многопользовательская компьютерная игра) решения обученных классификаторов могут поступать в систему разграничения доступа, используемую данным сервисом или сайтом,

- на рабочую панель модератора сервиса или специалиста по информационной безопасности, либо использоваться программным обеспечением системы для совершения действий, предупреждающих нелегитимный доступ к системе. Например, для инициирования прямой связи с данным пользователем и запроса дополнительной идентифицирующей информации
- 5 (контрольного вопроса, номера телефона и так далее).

В одном из возможных вариантов реализации данного способа такой сигнал может формироваться при условии отсутствия типичных паттернов на протяжении наперёд заданного количества открытых пользователем страниц, т.е. когда данный пользователь достаточно долго ведёт себя нетипично.

- 10 Возможен также вариант реализации системы, при котором условием формирования тревожного сигнала является отсутствие на протяжении наперёд заданного времени по меньшей мере одного паттерна, имеющего наибольшую частотность из всех паттернов данного пользователя.

- Описываемый способ может быть реализован в любом варианте сервиса авторизованного удалённого доступа, который подразумевает наличие нескольких окон пользовательского интерфейса либо наличие нескольких рабочих зон в одном сложном окне
- 15 пользовательского интерфейса и различных возможных путей переключения между ними (как окнами, так и зонами).

- На Фиг. 2 далее будет представлена общая схема компьютерного устройства (200), обеспечивающего обработку данных, необходимую для реализации заявленного решения.
- 20

В общем случае устройство (200) содержит такие компоненты, как: один или более процессоров (201), по меньшей мере одну память (202), средство хранения данных (203), интерфейсы ввода/вывода (204), средство В/В (205), средства сетевого взаимодействия (206).

- 25 Процессор (201) устройства выполняет основные вычислительные операции, необходимые для функционирования устройства (200) или функциональности одного или более его компонентов. Процессор (201) исполняет необходимые машиночитаемые команды, содержащиеся в оперативной памяти (202).

- Память (202), как правило, выполнена в виде ОЗУ и содержит необходимую программную логику, обеспечивающую требуемый функционал.
- 30

Средство хранения данных (203) может выполняться в виде HDD, SSD дисков, рейд массива, сетевого хранилища, флэш-памяти, оптических накопителей информации (CD,

DVD, MD, Blue-Ray дисков) и т.п. Средство (203) позволяет выполнять долгосрочное хранение различного вида информации, например, вышеупомянутых файлов с наборами данных пользователей, базы данных, содержащих записи измеренных для каждого пользователя временных интервалов, идентификаторов пользователей и т.п.

- 5 Интерфейсы (204) представляют собой стандартные средства для подключения и работы с компьютерным устройством, например, USB, RS232, RJ45, LPT, COM, HDMI, PS/2, Lightning, FireWire и т.п.

- Выбор интерфейсов (204) зависит от конкретного исполнения устройства (200), которое может представлять собой персональный компьютер, мейнфрейм, серверный кластер, тонкий клиент, смартфон, ноутбук, быть частью банковского терминала, банкомата и т.п.
- 10

В качестве средств В/В данных (205) могут использоваться мышь, джойстик, дисплей (сенсорный дисплей), проектор, тачпад, клавиатура, трекбол, световое перо, динамики, микрофон и т.п.

- Средства сетевого взаимодействия (N06) выбираются из устройства, обеспечивающий сетевой прием и передачу данных, например, Ethernet карту, WLAN/Wi-Fi модуль, Bluetooth модуль, BLE модуль, NFC модуль, IrDa, RFID модуль, GSM модем и т.п. С помощью средств (N05) обеспечивается организация обмена данными по проводному или беспроводному каналу передачи данных, например, WAN, PAN, ЛВС (LAN), Интранет, Интернет, WLAN, WMAN или GSM.
- 15

- 20 Компоненты устройства (200) сопряжены посредством общей шины передачи данных (210).

- В настоящих материалах заявки было представлено предпочтительное раскрытие осуществление заявленного технического решения, которое не должно использоваться как ограничивающее иные, частные воплощения его реализации, которые не выходят за рамки испрашиваемого объема правовой охраны и являются очевидными для специалистов в соответствующей области техники.
- 25

Формула изобретения

1. Способ идентификации пользователя по последовательности открываемых окон пользовательского интерфейса, заключающийся в выполнении этапов, на которых с помощью вычислительного устройства:
- 5 - регистрируют учетные данные пользователя в системе;
- присваивают соответствующий учетным данным идентификатор пользователя;
- каждому открываемому пользователем окну пользовательского интерфейса присваивают идентификатор;
- на протяжении всего сеанса работы идентифицированного пользователя фиксируют
- 10 последовательность открываемых окон пользовательского интерфейса, при этом:
- сохраняют идентификатор окна, которое в данный момент открыто пользователем, и время открытия данного окна;
 - при каждом переходе к новому окну пользовательского интерфейса сохраняют идентификатор этого окна и время, когда оно было открыто;
- 15 - накапливают заранее заданное количество сеансов работы данного пользователя;
- анализируют накопленные данные, а именно выявляют повторяющиеся последовательности посещаемых окон пользовательского интерфейса (паттерны);
- для каждого выявленного паттерна вычисляют набор параметров, характеризующих время, проходящее между переходами данного пользователя от окна к окну
- 20 пользовательского интерфейса;
- сохраняют заданное количество паттернов данного пользователя и на основе вычисленных для каждого паттерна набора параметров обучают по меньшей мере один классификатор идентифицировать данного пользователя по последовательности посещаемых страниц;
- 25 - применяют обученный по меньшей мере один классификатор для последующего подтверждения идентичности пользователя, во время сеансов работы которого был обучен по меньшей мере один классификатор.
2. Способ по п.1, характеризующийся тем, что окна пользовательского интерфейса являются веб-страницами.

3. Способ по п.1, характеризующийся тем, что из выявленных паттернов дополнительно отбирают такие, длина и количество которых удовлетворяют заранее заданным критериям, и наборы параметров вычисляют только для отобранных паттернов.
4. Способ по п.3, характеризующийся тем, что отбирают паттерны, имеющие максимально возможную длину, причём их количество должно быть не больше и не меньше заранее заданных значений.
5. Способ по п. 1, характеризующийся тем, что паттернам, содержащим заранее определённые окна пользовательского интерфейса, могут быть присвоены априорные веса, которые веса учитывают при выборе или обучении по меньшей мере одного классификатора таким образом, чтобы соответствующие паттерны оказывали пропорционально большее или меньшее влияние на итоговое решение по меньшей мере одного классификатора.
6. Способ по п. 1, характеризующийся тем, что в набор параметров, вычисляемых для каждого паттерна, входит время, проходящее между переходами данного пользователя от окна к окну пользовательского интерфейса, и усреднённое по всем сеансам работы данного пользователя, в которых встречается данный паттерн.
7. Способ по п. 6, характеризующийся тем, что в набор параметров, вычисляемых для каждого паттерна, дополнительно входит дисперсия, вычисленная как дисперсия случайной величины, для временных интервалов между переходами данного пользователя от окна к окну пользовательского интерфейса.
8. Способ по п.6, характеризующийся тем, что в набор параметров, вычисляемых для каждого паттерна, дополнительно входит частотность данного паттерна в сеансах работы данного пользователя.
9. Способ по п. 8, характеризующийся тем, что частотность учитывают при выборе или обучении по меньшей мере одного классификатора таким образом, чтобы паттерны с большей частотностью оказывали пропорционально большее влияние на итоговое решение по меньшей мере одного классификатора.
10. Способ по п. 1, характеризующийся тем, что классификатор обучают методом машинного обучения.
11. Способ по п. 10, характеризующийся тем, что классификатор может быть реализован как графовая вероятностная модель или как SVM-классификатор.

12. Система идентификации пользователя по последовательности посещаемых окон пользовательского интерфейса, содержащая:

- блок регистрации учетных данных пользователя в системе;
- системные часы, выполненные с возможностью фиксации времени

5 - блок идентификации текущего окна пользовательского интерфейса;

- долговременную память, выполненную с возможностью хранения базы данных;
- вычислительное устройство, выполненное с возможностью выполнения способа по любому из пп. 1-11.

10 13. Система по п. 12, характеризующаяся тем, что система идентификации пользователя по последовательности посещаемых окон пользовательского интерфейса является составной частью дистанционной системы банковского обслуживания.

14. Система по п. 12, характеризующаяся тем, что система идентификации пользователя по последовательности посещаемых окон пользовательского интерфейса является составной частью веб-сайта, подразумевающего авторизацию пользователей.

15 15. Система по п. 12, характеризующаяся тем, что система идентификации пользователя по последовательности посещаемых окон пользовательского интерфейса является составной частью многопользовательской компьютерной игры, подразумевающей авторизацию пользователей.

20 16. Система по п. 12, характеризующаяся тем, что система идентификации пользователя по последовательности посещаемых окон пользовательского интерфейса является составной частью программно-аппаратного комплекса для совместной работы, подразумевающего авторизацию пользователей.