**Интерактивная безопасность**

Данная технология относиться к компьютерным технологиям. Она позволяет идентифицировать пользователя посредством его взаимодействия с интерактивной виртуальной средой. Под интерактивной виртуальной средой подразумевается графическое или игровое программное обеспечение, используя которое пользователь сможет идентифицировать себя.

Из существующих подобных технологий на данный момент присутствуют паттерны разблокировки экрана, которые активно используются в аппаратных решениях безопасности, а также на мобильных устройствах. Недостаток существующих решений: однообразие паттернов идентификации, статическое изображение, возможность грубого перебора вариантов для взлома системы. Главная причина несовершенства существующих технологий идентификации пользователей – использование стандартной системы логина и пароля, которая вследствие человеческого фактора может быть взломана с помощью программного обеспечения для прослушивания сетевого взаимодействия.

Задача, которую данное предлагаемое техническое решение решает, является создание нового уровня защиты от проблемы человеческого фактора в работе комплексной системы безопасности. Современные технические средства позволяют обеспечить наивысший уровень безопасности для любого типа информационных систем. Главный недостаток заключается в том, что из-за ошибки пользователя в систему может проникнуть вредоносное программное обеспечение. Также под человеческим фактором нужно расматривать потерю мобильного устройства, посредством которого злоумышленник, зная данные аутентификации, может проникнуть в систему.

Данная задача решается в несколько этапов.

Во-первых, интерактивная безопасность позволяет произвести собственное внедрение в уже существующие решения, для создания нового/уникального метода идентификации в процессе работы с программной системой.

Во-вторых, интерактивная безопасность состоит из двух программных продуктов, которые создают закрытую программную систему безопасности используя архитектуру клиент-сервер.

При регистрации нового пользователя, система интерактивной безопасности выдает пользователю три шаблона, которые случайным образом выбираются из базы данных, находящейся на стороне сервера. Передача данных шаблонов осуществляется через защищенный шифрованный канал связи. При получении шаблонов, пользователю необходимо самостоятельно выбрать алгоритм взаимодействия с каждым из выданных интерактивных шаблонов.

Таким образом, после взаимодействия с каждым из шаблонов, на стороне пользователя система формирует шифрованный слепок его личного решения/взаимодействия с паттернами.

**Структура системы**

Система состоит из следующих частей: **Сервера**, **Клиента** и **Шифровальщика**.

Сервер служит для хранения, обработки интерактивной информации, а также для взаимодействия посредством запросов с **Клиентом**.

**Клиент** представляет собой интерактивное приложение, которое получает данные с **Сервера** и визуализирует их, ожидая определенных входных данных для подтверждения правильного взаимодействия с интерактивными данными.

**Шифровальщик** работает в качестве системы запрос-ответ, и позволяет шифровать данные во время их передачи между **Клиентом** и **Сервером**. Ключи для шифрования **Шифровальщик** получает от **Сервера** по запросу.

Рассмотрим детальнее алгоритм работы системы.

Для работы системы используется три внутренних компонента – **Клиент**, **Сервер**, и **Шифровальщик**.

Сперва система запускает **Сервер**, который после запуска инициализирует локальное хранилище данных, и переходит в статус ожидания запросов от **Клиента**.

На втором этапе система запускает внутренний **Шифровальщик**. Это отдельное приложение. После запуска, **Шифровальщик** становиться в статус ожидания запросов от **Клиента** и **Сервера** для шифрования/дешифровки запросов.

На третьем этапе система запускает внутренний **Клиент**. После запуска, **Клиент** отправляет запрос **Шифровальщику** с пометкой «Первый запуск» и свой уникальный ключ. **Шифровальщик** закрывает запрос защитным ключом и переправляет его **Серверу**.

**Сервер** получает запрос от **Шифровальщика**, дешифрует его, считывает ключ и проверяет его у себя в локальной базе данных. Если **Клиент** с данным уникальным ключом запускается первый раз для данного пользователя, тогда **Сервер** выбирает из локальной базы данных случайный **Интерактивный Паттерн Безопасности**, собирает из него пакет данных и посылает **Шифровальщику**. **Шифровальщик** шифрует пакет данных и пересылает их **Клиенту.**

**Клиент** получает от **Шифровальщика** пакет данных, расшифровывает их с помощью ключа безопасности, и считывает **Интерактивный Паттерн Безопасности**.

**Интерактивный Паттерн Безопасности** представляет собой структуру данных, которая содержит расположение интерактивных элементов на устройстве вывода информации. При считывании **Интерактивного Паттерна Безопасности,** Клиент забирает данные о размещении интерактивных элементов, и расставляет их на **Интерактивной сцене**.

**Интерактивная сцена** представляет собой игровую сцену со статическими и динамическими элементами на экране устройства, с которой взаимодействует система для подтверждения аутентификации.

После расстановки интерактивных элементов на экране устройства вывода, **Клиент** переходит в режим ожидания новых входных данных.

Во время взаимодействия с интерактивными элементами на экране устройства, система формирует определенный порядок перехода от одного интерактивного элемента к другому интерактивному элементу. Данный порядок **Клиент** сохраняет в последовательность данных. После завершения взаимодействия с **Интерактивным Паттерном Безопасности**, **Клиент** формирует пакет данных из собранных входных данных о последовательности действий и посылает его **Шифровальщику.**

**Шифровальщик** получает эти данные, шифрует их с помощью ключа безопасности и передает на **Сервер.**

**Сервер** получает шифрованный пакет данных от **Шифровальщика**, расшифровывает их, и получает готовую последовательность взаимодействия входных данных системы с **Клиентом**. Данную последовательность **Сервер** сохраняет в локальной базе данных с уникальным ключом, который получает посредством хеш-функции, куда передается уникальный идентификатор системы и дата ее регистрации.

**Сервер** создает пакет данных с подтверждением аутентификации пользователя, высылает его **Шифровальщику**, который пересылает данный уже шифрованный пакет данных **Клиенту.**

**Клиент** получает шифрованный пакет данных от **Шифровальщика,** расшифровывает их с помощью ключа безопасности, проверяет ответ и позволяет продолжить работу уже непосредственно с системой, в которую была осуществленна попытка зайти.

При выходе из информационной системы компонеты **Интерактивной Системы Безопасности** прекращают свою работу в следующем порядке – **Клиент, Сервер** и потом **Шифровальщик**.

При повторном запуске система запускает **Сервер**, который после запуска инициализирует локальное хранилище данных, и переходит в статус ожидания запросов от **Клиента**.

На втором этапе система запускает внутренний **Шифровальщик**. После запуска, **Шифровальщик** становиться в статус ожидания запросов от **Клиента** и **Сервера** для шифрования/дешифровки запросов.

На третьем этапе система запускает внутренний **Клиент**. После запуска, **Клиент** отправляет запрос **Шифровальщику** с пометкой «Запуск» и свой уникальный ключ. **Шифровальщик** закрывает запрос защитным ключом и переправляет его **Серверу**.

**Сервер** получает запрос от **Шифровальщика**, дешифрует его, считывает ключ и проверяет его у себя в локальной базе данных. Если уникальный ключ найден в базе, **Сервер** забирает из хранилища по этому ключу данные об **Интерактивном Паттерне Безопасности**, передает его **Шифровальщику**, который в свою очередь его шифрует, и перенаправляет **Клиенту.**

**Клиент** получает данные, расшифровывает их с помощью **Шифровальщика** и ключа безопасности, и на основе этих данных строит интерактивную сцену на экране устройства. После построения интерактивной сцены, **Клиент** переходит в ожидание входных данных.

При взаимодействии с интерактивными элементами на экране устройства, создаются входные данные, которые подаются на **Клиента.** **Клиент** принимает эти данные, упаковывает свой уникальный идентификатор и собранные данные, передает их **Шифровальщику**, который в свою очередь их шифрует, упаковывает и передает **Серверу.**

**Сервер** получает данные, расшифровывает их с помощью **Шифровальщика**. С помощью уникального идентификатора, который **Сервер** получил в текущем пакете данных от **Шифровальщика**, **Сервер** достает из локального хранилища данные о правильном взаимодействии с **Интерактивным Паттерном Безопасности,** и сравнивает с данными о взаимодействии с **Интерактивным Паттерном Безопасности,** полученном от **Клиента.**

При совпадении двух последовательностей данных по размеру последовательностей и по порядку следования данных, **Сервер** создает запрос «Верно», пересылает его **Шифровальщику,** который шифрует этот запрос и перенаправляет **Клиенту.** Если же последовательности не совпали по длинне, или не совпали по порядку следования элементов, или не совпали по последовательности и порядку следования элементов, то **Сервер** создает запрос «Неверно», пересылает его **Шифровальщику**, который шифрует запрос и перенаправляет **Клиенту.**

**Клиент** получает сообщение от **Сервера,** передает его **Шифровальщику**, который с помощью уникального ключа безопасности расшифровывает его, и передает обратно **Клиенту.**

**Клиент** анализирует расшифрованное сообщение и, если сообщение «Верно», то **Клиент** переходит в состояние, при котором все входные данные идут прямиком в систему. Если **Клиент** получил сообщение «Неверно», то **Клиент** переходит в состояние блокировки всех входных данных, которые идут извне в систему.