**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.

Для передачи шифрованного слепка, система создает 16 каналов связи с сервером, каждый из которых является отдельным потоков внутри программного обеспечения на стороне клиента. Далее шифрованный слепок разбивается на случайное к-во частей, и отправляется по заранее выбранным каналам. Выбор производиться сервером, и высылается шифрованным пакетом клиенту после установки защищенного соединения.

**Структура системы**

Система состоит из **Сервера**, **Клиента** и **Шифровальщика**.

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

**Клиент** является приложением, с которым работает пользователь системы. **Клиент** представляет собой интерактивное приложение, которое получает данные с **Сервера** и визуализирует их, ожидая определенных действий пользователя для подтверждения его аутентичности.

**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 комплексный алгоритм работы системы.

Для работы системы должно быть запущено три компонента – **Клиент**, **Сервер**, и **Шифровальщик**.

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

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

На третьем этапе пользователь системы запускает **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 экране устройства пользователя. 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 пользователь для подтверждения своей аутентичности.

После расстановки интерактивных элементов на экране устройства, **Клиент** переходит в режим ожидания действий пользователя.

Во время взаимодействия с интерактивными элементами на экране устройства, пользователь формирует определенный порядок перехода от одного элемента к другому. Данный порядок **Клиент** сохраняет в последовательность данных. После завершения взаимодействия с **Интерактивным Паттерном Безопасности**, **Клиент** формирует пакет данных из собранных действий пользователя 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 пользователя с **Клиентом**. Данную последовательность **Сервер** сохраняет в локальной базе данных с уникальным ключом, который получает посредством хеш-функции, куда передается имя пользователя и дата его регистрации в системе.

**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 пользователю продолжить работу уже непосредственно с системой, в которую пользователь пытается зайти.

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