Криптографические протоколы

Лекция 4

Протоколы аутентификации: классификация, атаки

Протоколы "слабой" аутентификации

Деркач Максим Юрьевич

October 9, 2019

Ссылки

https://habr.com/post/154229/
https://habr.com/en/company/dataart/blog/262817/
https://en.wikipedia.org/wiki/Pass_the_hash

Протоколы аутентификации

Определения

Определение 1
Аутентификация - подтверждение подлинности.

Определение 2
Идентификация - однозначное именование (присвоение уникальных имён или признаков) компонентов автоматизированной системы и всех лиц (пользователей), взаимодействующих с системой.

Определение 3
Протокол аутентификации - криптографический протокол, в ходе которого одна сторона удостоверяется в идентичности другой стороны, вовлеченной в протокол, а также убеждается в том, что вторая сторона активна во время или непосредственно перед моментом выполнения протокола.

Классификация аутентификации

+ По количеству доказывающих сторон:
  * одностороняя - доказывающая сторона A и проверяющая сторона B;
  * двустороняя - обе стороны A и B доказывают свою подлинность друг другу.

+ По устойчивости:
  * протоколы "слабой" аутентификации (на основе фиксированных или одноразовых паролей);
  * протоколы "сильной" аутентификации (на основе запроса типа "вопрос-ответ");
  * протоколы основанные на техники доказательства знания.

Цель протокола - установление того факта, что проверяемая сторона является той, за кого она себя выдаёт.

Возможны два исхода: подтверждение подлинности, не подтверждение.

Протоколы слабой аутентификации

Фиксированные пароли

$A -> S : ID_A || P$

Угрозы:

1. раскрытие пароля (разглашение, восстановление из системной информации);
2. перехват пароля (внутри системы);
3. угадывание пароля.

Атаки на фиксированные пароли:

1. повторное использование пароля;
2. тотальный перебор;
3. атака со словарём.

Фиксированные пароли

Приёмы повышения стойкости

1. Хранение в компьютерной системе файлов паролей в защищенном режиме (с защитой от чтения/записи).
2. Хранение в системе не самих паролей, а их образов.
3. Задание правил выбора паролей.
4. Ограничение попыток ввода пароля.
5. Добавление "соли" к паролю (добавление случайной величины к паролю перед обработкой его однонаправленной функцией).
6. Многофакторная аутентификация.

Фиксированные пароли

Многофакторная аутентификация

1. Смарт-карта
2. Электронный идентификатор
3. Биометрические аутентификаторы
4. SMS-аутентификация

Фиксированные пароли

Использование криптографических методов для повышения стойкости

На сервере обычно хранятся пароли в зашифрованном виде либо хэш от пароля.

1. $A -> S : ID_A$
2. $S -> A : R_S || text_A$
3. $A -> S : ID_A || h_1(R_S || h_2(p_A || text_A))$

где $ID_A, text_A, h_2(p_A||text_A)$ хранятся на проверяющей стороне(сервере).

Однако такой протокол неустойчив к атаке MITM и атаке параллельного сеанса.

Фиксированные пароли [Примеры]

HTTP authentication

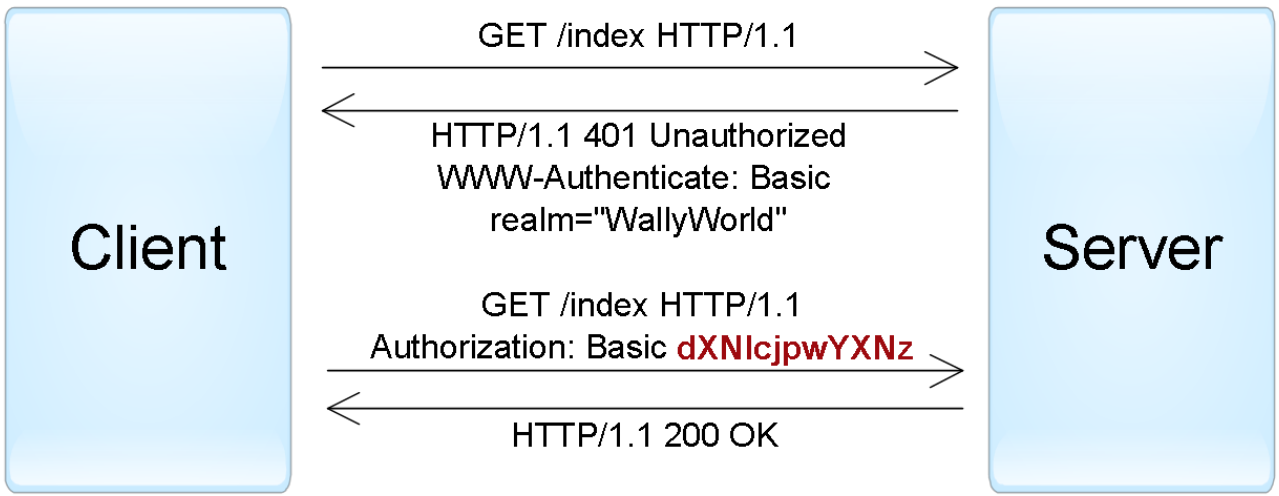
Этот протокол, описанный в стандартах HTTP 1.0/1.1, существует очень давно и до сих пор активно применяется в корпоративной среде. Применительно к веб-сайтам работает следующим образом:

1. Сервер, при обращении неавторизованного клиента к защищенному ресурсу, отсылает HTTP статус “401 Unauthorized” и добавляет заголовок “WWW-Authenticate” с указанием схемы и параметров аутентификации.
2. Браузер, при получении такого ответа, автоматически показывает диалог ввода username и password. Пользователь вводит детали своей учетной записи.
3. Во всех последующих запросах к этому веб-сайту браузер автоматически добавляет HTTP заголовок “Authorization”.
4. Сервер аутентифицирует пользователя по данным из этого заголовка.

Фиксированные пароли [Примеры]

HTTP authentication[Basic]

Basic — наиболее простая схема, при которой username и password пользователя передаются в заголовке Authorization в незашифрованном виде (base64-encoded). Однако при использовании HTTPS (HTTP over SSL) протокола, является относительно безопасной.



Фиксированные пароли [Примеры]

HTTP authentication[Digest]

Digest — challenge-response-схема, при которой сервер посылает уникальное значение nonce, а браузер передает MD5 хэш пароля пользователя, вычисленный с использованием указанного nonce. Более безопасная альтернативв Basic схемы при незащищенных соединениях, но подвержена man-in-the-middle attacks.

Фиксированные пароли [Примеры]

HTTP authentication[NTLM, Negotiate]

NTLM — также основана на challenge-response подходе, при котором пароль не передается в чистом виде. Эта схема не является стандартом HTTP, но поддерживается большинством браузеров и веб-серверов. Преимущественно используется для аутентификации пользователей Windows Active Directory в веб-приложениях. Уязвима к pass-the-hash-атакам.

Negotiate - ще одна схема из семейства Windows authentication, которая позволяет клиенту выбрать между NTLM и Kerberos аутентификацией.

Стоит отметить, что при использовании HTTP-аутентификации у пользователя нет стандартной возможности выйти из веб-приложения, кроме как закрыть все окна браузера.

Фиксированные пароли [Примеры]

Другие протоколы аутентификации по паролю

1. **Forms authentication** - наиболее популярный метод аутентификации.
2. **URL query** — считается небезопасным вариантом, т. к. строки URL могут запоминаться браузерами, прокси и веб-серверами.
3. **HTTP header** — оптимальный вариант, при этом могут использоваться и стандартный заголовок Authorization (например, с Basic-схемой), и другие произвольные заголовки.

Фиксированные пароли [Примеры]

Распространенные уязвимости и ошибки реализации

1. Веб-приложение позволяет пользователям создавать простые пароли.
2. Веб-приложение не защищено от возможности перебора паролей (brute-force attacks).
3. Веб-приложение само генерирует и распространяет пароли пользователям, однако не требует смены пароля после первого входа (т.е. текущий пароль где-то записан).
4. Веб-приложение допускает передачу паролей по незащищенному HTTP-соединению, либо в строке URL.
5. Веб-приложение не использует безопасные хэш-функции для хранения паролей пользователей.

Фиксированные пароли [Примеры]

Распространенные уязвимости и ошибки реализации

6. Веб-приложение использует уязвимую функцию восстановления пароля, которую можно использовать для получения несанкционированного доступа к другим учетным записям.
7. Веб-приложение не требует повторной аутентификации пользователя для важных действий: смена пароля, изменения адреса доставки товаров и т. п.
8. Веб-приложение создает session tokens таким образом, что они могут быть подобраны или предсказаны для других пользователей.
9. Веб-приложение допускает передачу session tokens по незащищенному HTTP-соединению, либо в строке URL.

Фиксированные пароли [Примеры]

Распространенные уязвимости и ошибки реализации

10. Веб-приложение не предоставляет пользователям возможность изменения пароля либо не нотифицирует пользователей об изменении их паролей.
11. Веб-приложение уязвимо для session fixation-атак (т. е. не заменяет session token при переходе анонимной сессии пользователя в аутентифицированную).
12. Веб-приложение не устанавливает флаги HttpOnly и Secure для browser cookies, содержащих session tokens.
13. Веб-приложение не уничтожает сессии пользователя после короткого периода неактивности либо не предоставляет функцию выхода из аутентифицированной сессии.

Одноразовые пароли

1. Разделяемые списки одноразовых паролей: пользователь и система имеют заранее определенный список паролей, который каждый из них хранит самостоятельно. При выполнении очередного сеанса протокола аутентификации выбирается пользователем и проверяется системой очередной пароль из этого списка .
2. Последовательно обновляемые одноразовые пароли: Первоначально пользователь и система имеют только один пароль , условно с номером i. Затем пользователь создает и передает системе пароль под номером i-1, зашифрованный на ключе, вычисленном из i-го пароля. Такой метод затруднительно реализовать при ненадежном канале связи (при возможности обрыва связи).
3. Последовательности одноразовых паролей, основанные на однонаправленных функциях.

Одноразовые пароли

Схема Лэмпорта с одноразовыми паролями (RFC 1760 - The S/Key One-Time Password System)

На проверяющей стороне(сервере) хранятся $ID_A, h^n(p_A)$, где n - достаточно большое.

1. $A -> S : ID_A || h^{n-1}(p_A)$
2. Сервер вычисляет $h(h^{n-1}(p_A))$ и сравнивает с хранящимися данными, если совпало, то аутентификация пройдена успешна, и запись обоновляется на $ID_A || h^{n-1}(p_A)$.

S/Key

1. $A -> S : ID_A$
2. $S -> A : m$
3. $A -> S : h^{m-1}(p_A)$

Одноразовые пароли

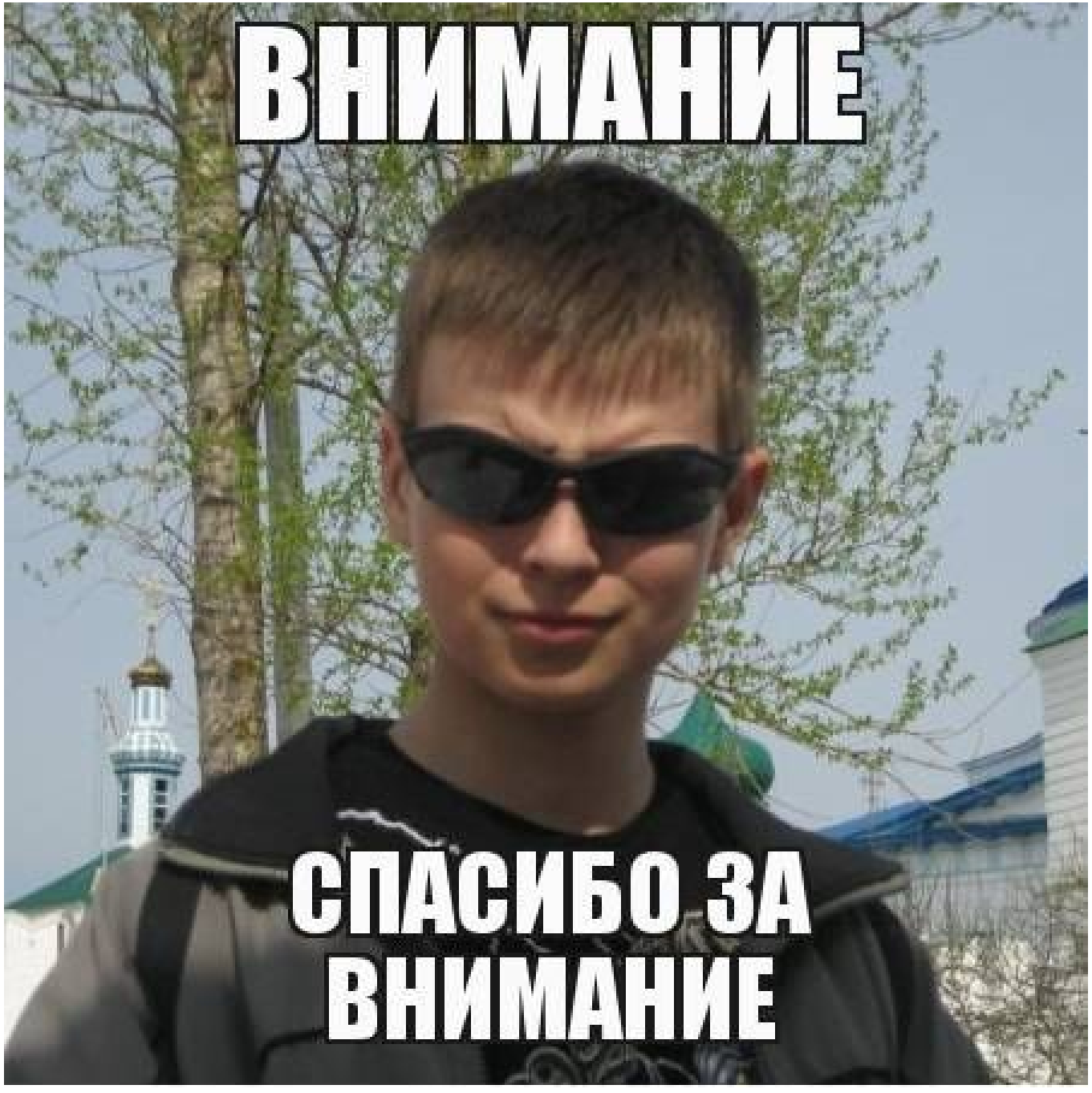
Схема Лэмпорта с одноразовыми паролями (RFC 1760 - The S/Key One-Time Password System)

Существует атака

1. $A -> I(S) : ID_A$
2. $I(A) -> S : ID_A$
3. $S -> I(A) : m$
4. $I(S) -> A : m - 1$
5. $A -> I(S) : h^{m-2}(p_A)$
6. $I(A) -> S : h(h^{m-2}(p_A))$

Следующий раз

1. $I(A) -> S : ID_A$
2. $S -> I(A) : m - 1$
3. $I(A) -> S : h^{m-2}(p_A)$


ВНИМАНИЕ
СПАСИБО ЗА ВНИМАНИЕ