

Хранение секретов и обеспечение безопасности .NET приложений: уроки и выводы

Руслан Каменский
SpbDotNet 2024

Обо мне

- 12 лет в FinTech индустрии
- Более 10 крупных финансовых проектов для заказчиков из Европы, Великобритании, США
- Несколько блокчейн проектов
- Запуск и сопровождение проектов проходящих регуляцию (FCA, CySEC, FinCom, PCI DSS)

Примеры секретов

- Пароли и ключи API, которые приложения используют для доступа к внешним сервисам.
- Ключи шифрования и сертификаты, используемые для защиты данных.
- Токены доступа, которые подтверждают права доступа приложений к ресурсам.
- Конфигурационные строки, включающие подключения к базам данных.

Неадекватная защита секретов может привести к:

- **Утечкам данных**: Конфиденциальная информация, такая как личные данные пользователей или финансовая информация, может быть скомпрометирована.
- **Финансовым потерям**: Злоумышленники могут использовать украденные ключи API для создания мошеннических транзакций или для доступа к платным ресурсам.
- **Повреждению репутации**: Утечки данных могут серьезно навредить репутации вашей компании, уменьшая доверие клиентов и партнеров.
- **Юридическим последствиям**: Нарушения защиты данных могут привести к судебным искам, штрафам и другим правовым последствиям.

Секреты

- Девелоперские
- Тестовые
- Production

Секреты

- Девелоперские
- Тестовые
- Production

Очень
секретные

Менее
секретные

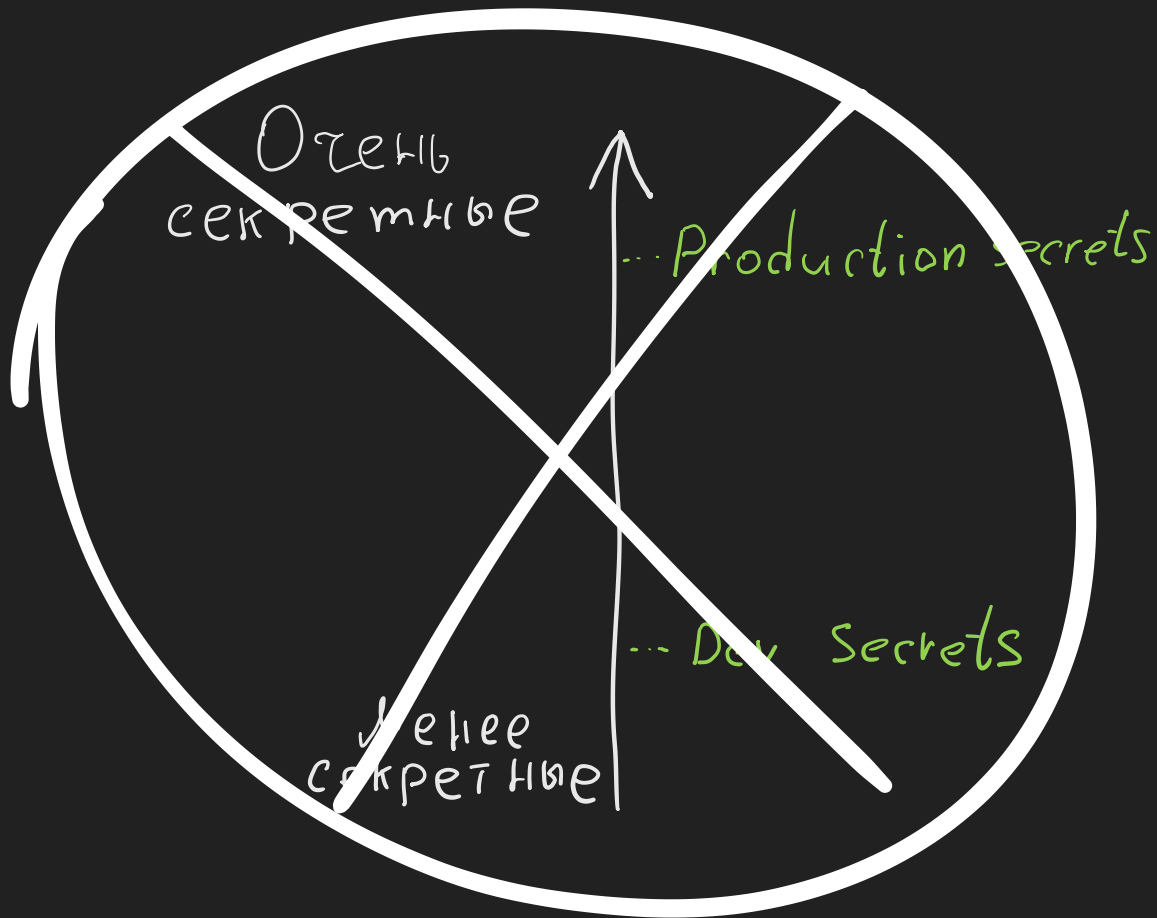


... Production secrets

... Dev Secrets

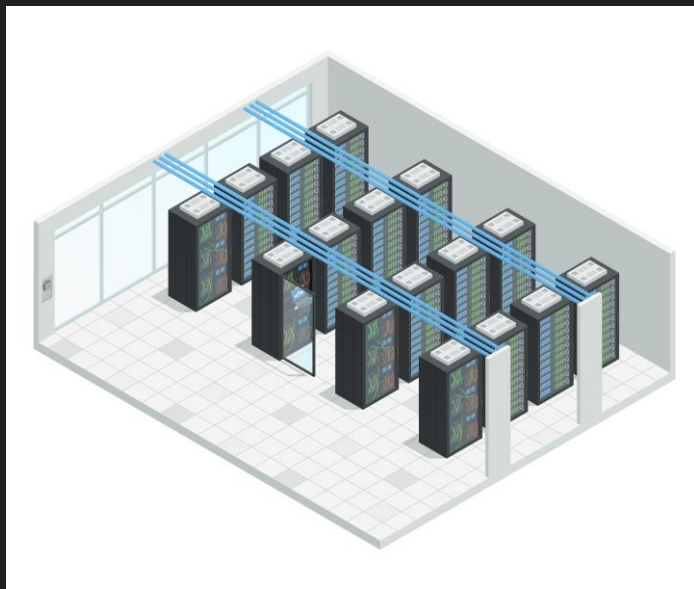
Секреты

- Девелоперские
- Тестовые
- Production



Контуры секретов

Production



Developers

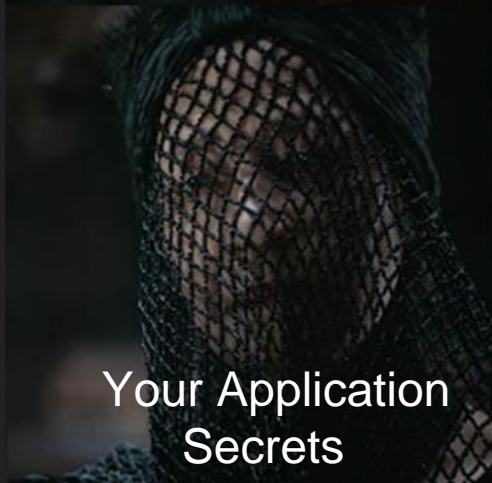




Вынешь руку из
коробки – умрёшь



А что в коробке?



Your Application
Secrets



Методы хранения секретов

- **Файлы конфигураций**
- **Переменные окружения**
- **Хранилище секретов**
 - Cloud secrets managers (Azure Key Vault, AWS Secrets Manager, ...)
 - Self hosted secrets managers (infisical, ...)
- **Secret Manager tool**

Файлы конфигураций

- **Простота использования**
- **Поддержка среды:** Интегрированная поддержка в Visual Studio и других инструментах .NET.
- **Низкий уровень безопасности:** Файлы конфигураций могут быть легко прочитаны, если злоумышленник получит доступ к файловой системе.
- **Трудности с управлением в разных средах:** Необходимо поддерживать разные версии файлов для разных сред, что увеличивает риск утечки секретов.

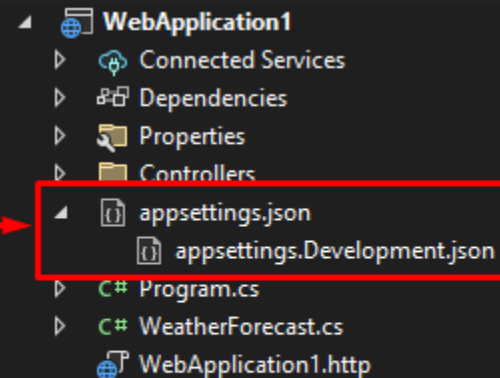
{ } appsettings.json •

C: > tpm > { } appsettings.json > ...

```
1  {
2    "DatabaseSettings": {
3      "Password": "EncryptedPasswordHere"
4    }
5  }
6
```

```
1  public class Startup
2  {
3      public void ConfigureServices(IServiceCollection services)
4      {
5          var configuration = new ConfigurationBuilder()
6              .AddJsonFile("appsettings.json")
7              .Build();
8
9          services.Configure<DatabaseSettings>(configuration.GetSection("DatabaseSettings"));
10     }
11 }
```

bin	7/23/2024 3:17 AM	File folder
Controllers	7/23/2024 3:17 AM	File folder
obj	7/23/2024 3:17 AM	File folder
Properties	7/23/2024 3:17 AM	File folder
appsettings.Development.json	7/23/2024 3:17 AM	JSON File
appsettings.json	7/23/2024 3:17 AM	JSON File
Program.cs	7/23/2024 3:17 AM	C# Source File
WeatherForecast.cs	7/23/2024 3:17 AM	C# Source File
WebApplication1.csproj	7/23/2024 3:17 AM	C# Project File
WebApplication1.csproj.user	7/23/2024 3:17 AM	Per-User Project O...
WebApplication1.http	7/23/2024 3:17 AM	HTTP File



Переменные окружения

- **Изоляция конфигурации:** Секреты не хранятся вместе с кодом приложения.
- **Поддержка различных платформ:** Переменные среды поддерживаются большинством операционных систем.
- **Видимость для других процессов:** Другие процессы, запущенные под теми же учетными записями, могут получить доступ к этим переменным.
- **Управление:** Требует настройки переменных среды на каждом сервере или в контейнерах, что может быть утомительным при большом количестве развертываний.

Системы управления секретами (Azure Key Vault, AWS Secrets Manager)

- **Централизованное управление:** Все секреты управляются из одного места, что упрощает управление, аудит и ротацию секретов.
- **Высокий уровень безопасности:** Эти системы предоставляют сильные меры безопасности, такие как шифрование, а также мониторинг и логирование доступа к секретам.
- **Стоимость:** Использование облачных сервисов несёт дополнительные расходы.
- **Зависимость от стороннего сервиса:** Потенциальные риски, связанные с простоями и доступностью сервиса.

Azure Key Vault

- **Secrets Management** - Azure Key Vault обеспечивает безопасное хранение токенов, паролей, сертификатов, ключей API и других секретных сведений со строгим контролем доступа к ним.
- **Key Management** - Azure Key Vault можно использовать как решение по управлению ключами. Эта служба позволяет легко создавать и контролировать ключи шифрования, используемые для шифрования данных.
- **Certificate Management** - С помощью службы Azure Key Vault можно с легкостью подготавливать, администрировать и развертывать общедоступные и частные сертификаты TLS/SSL для использования в Azure и внутренних подключенных ресурсах.

Установка пакетов

```
1 dotnet add package Azure.Security.KeyVault.Secrets
2 dotnet add package Azure.Identity
```

Получение значения секрета

```
1 public static void Main(string[] args)
2 {
3     var kvUri = "https://<your-key-vault-name>.vault.azure.net/";
4     var client = new SecretClient(new Uri(kvUri), new DefaultAzureCredential());
5
6     KeyVaultSecret secret = client.GetSecret("<your-secret-name>");
7     Console.WriteLine($"Secret: {secret.Value}");
8 }
```

Интеграция Azure Key Vault с .NET Core Configuration

```
1 public static IHostBuilder CreateHostBuilder(string[] args) =>
2     Host.CreateDefaultBuilder(args)
3         .ConfigureAppConfiguration((context, config) =>
4             {
5                 var builtConfig = config.Build();
6                 var keyVaultEndpoint = new Uri(builtConfig["KeyVault:VaultUri"]);
7                 var azureCredential = new DefaultAzureCredential();
8
9                 config.AddAzureKeyVault(keyVaultEndpoint, azureCredential);
10            })
11        .ConfigureWebHostDefaults(webBuilder =>
12            {
13                webBuilder.UseStartup<Startup>();
14            });
15
```

Secret Manager tool

- **Изоляция конфигурации:** Позволяет хранить секреты разработки локально и вне исходного кода.
- **Простота использования:** Хорошо интегрируется с .NET Core, легко добавлять и извлекать секреты.
- **Предназначен только для разработки:** Не предназначен для использования в продакшн среде.
- **Ограниченная функциональность:** Не предоставляет функций аудита, ротации ключей и др.

Инициализация

```
dotnet user-secrets init
```

Установка секрета

```
dotnet user-secrets set "Movies:ServiceApiKey" "12345"
```



Search Solution Explorer (Ctrl+;)



Solution 'WebApp1' (1 of 1 project)

WebApp1

- Build
- Rebuild
- Clean
- View
- Analyze and Code Cleanup
- Pack
- Publish...
- Configure Application Insights...
- Overview
- Scope to This
- New Solution Explorer View
- File Nesting
- Edit Project File
- Add
- Manage NuGet Packages...
- Manage Client-Side Libraries...
- Manage User Secrets
- Remove Unused References...
- Set as Startup Project
- Debug
- Cut

Ctrl+X

Ключи конфигурации:

- **Не учитывают регистр.** Например `ConnectionString` и `connectionstring` обрабатываются как эквивалентные ключи.
- Если ключ и значение заданы в нескольких поставщиках конфигурации, используется значение из последнего добавленного поставщика.
- Иерархические ключи
 - При взаимодействии с API конфигурации разделитель-двоеточие (:) поддерживается на всех платформах
 - В переменных среды разделитель-двоеточие может не работать на всех платформах. Двойной знак подчеркивания (__) поддерживается на всех платформах и автоматически преобразовывается в двоеточие — :
 - В Azure Key Vault иерархические ключи используют -- в качестве разделителя

Так как хранить секреты?

Так как хранить секреты?

- Как проводить ротацию секретов?
- Что делать, если уволился сотрудник имеющий доступ к секретам?
- Как выдать секреты новому разработчику?
- Как проводить аудит и расследование в случае инцидентов?
- Как бэкапить секреты?

Так как хранить секреты?

Менеджер секретов

+

Secret Manager tool (локально)

Соблюдайте регуляцию

Regulation and compliance

- GDPR
- FIPS
- HIPAA
- PCI DSS
- SOX
- ISO/IEC 27001
- FISMA
- CCPA
- NIST

Основные требования PCI DSS:

1. Установка и поддержка межсетевого экрана
2. Не использовать стандартные пароли по умолчанию
3. Защита сохраненных данных карты
4. Шифрование передачи данных карты по открытым, общедоступным сетям
5. Использование и обновление антивирусных программ
6. Разработка и поддержка безопасных систем и приложений
7. Ограничение доступа к данным карты по необходимости
8. Присвоение уникального ID каждому лицу с доступом к компьютеру
9. Ограничение физического доступа к данным карты
10. Отслеживание и мониторинг всего доступа к ресурсам сети и данным карты
11. Тестирование безопасности систем и процессов
12. Политика информационной безопасности

FIPS (*Federal Information Processing Standards*)

Открыто публикуемые стандарты, разработанные правительством США, используемые всеми гражданскими правительственными учреждениями и контрагентами в США

- **FIPS 140**
- **FIPS 186**
- **FIPS 199**
- **FIPS 200**
- **FIPS 201**
- **FIPS 202**
- ...

FIPS 140

Стандарт, который определяет требования к криптографическим модулям, включая аппаратные и программные компоненты, используемые для защиты чувствительной информации.

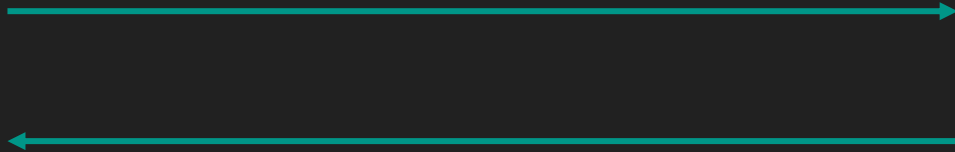
- **Level 1**
Обеспечивает базовую безопасность без значительных требований к физической защите.
- **Level 2.**
Требует некоторой степени физической защиты и аутентификации пользователей.
- **Level 3**
Обеспечивает защиту от несанкционированного физического доступа, включая защиту ключей шифрования.
- **Level 4**
Защищает модуль от воздействия внешней среды, таких как температура и электромагнитные излучения. Включает меры для защиты от сложных атак, таких как атаки по сторонним каналам.

Примеры использования FIPS 140-3

- **Финансовый сектор:** Защита данных транзакций и личной информации клиентов.
- **Здравоохранение:** Защита медицинских записей и конфиденциальной информации пациентов.
- **Государственные учреждения:** Защита данных, связанных с национальной безопасностью и конфиденциальной информацией.
- **Телекоммуникации:** Защита данных о пользователях и коммуникационных каналах.



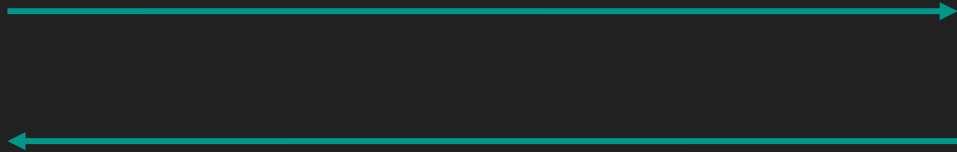
Our service



Provider



Our service

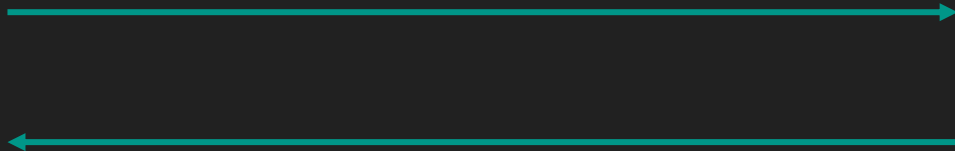


Provider

HTTP



Our service

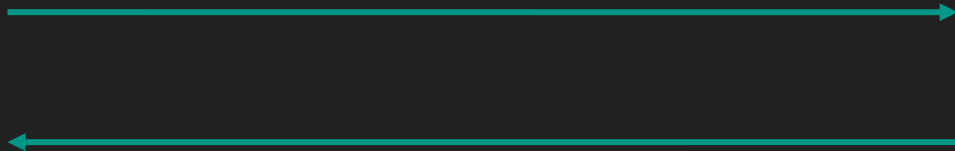


Provider

HTTP → HTTPS



Our service

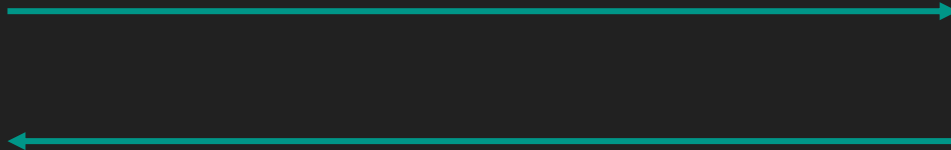


Provider

HTTP → HTTPS → Self Signed
Certificates



Our service

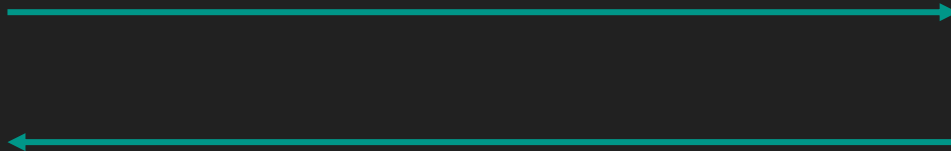


Provider

HTTP → HTTPS → Self Signed
Certificates → Trusted
Certificates



Our service



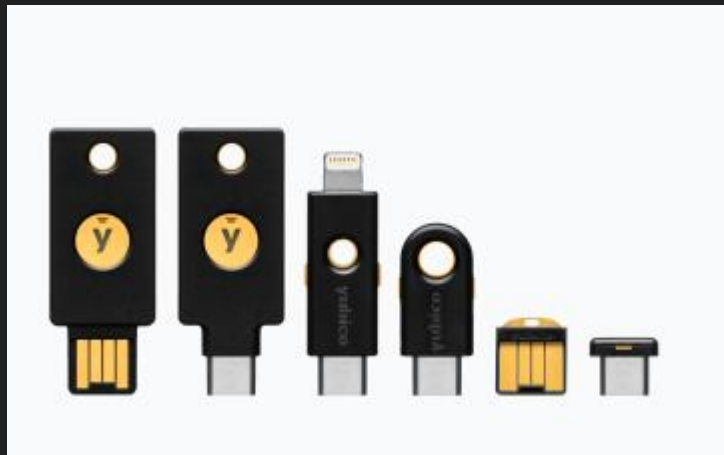
Provider

HTTP → HTTPS → Self Signed
Certificates → Trusted
Certificates → HSM
Modules

HSM (Hardware security module)



Хранение личных секретов



YubiKey



Trezor



Ledger

Немного о безопасности приложений

Требования к безопасности

- Best practice
- Здравый смысл
- Regulation and compliance

.NET Security Best Practices

- Следуйте Microsoft Security Standards
<https://learn.microsoft.com/en-us/dotnet/standard/security>

Рекомендации

- Используйте современные методы аутентификации и авторизации
- Всегда валидируйте входные данные
- Храните секреты в безопасных хранилищах
- Шифруйте данные
- Регулярно обновляйте ваше ПО
- Проводите регулярные аудиты безопасности
- Обрабатывайте ошибки корректно
- Минимизируйте использование сторонних библиотек
- Защищайте от атак типа SQL Injection
- Реализуйте правильное управление сессиями
- Ограничивайте права доступа

No custom cryptography

```
public static string GenerateResetToken()
{
    var random = new Random();
    return random.Next(100000, 999999).ToString();
}
```

```
public static byte[] Encrypt(byte[] data, byte[] key)
{
    using (AesCryptoServiceProvider aes = new AesCryptoServiceProvider())
    {
        aes.Key = key;
        aes.Mode = CipherMode.CBC;
        aes.Padding = PaddingMode.PKCS7;
        using (ICryptoTransform encryptor = aes.CreateEncryptor())
        {
            return encryptor.TransformFinalBlock(data, 0, data.Length);
        }
    }
}
```

```
public static bool CheckPassword(string password, byte[] correctHash)
{
    byte[] testHash = ComputeHash(password);
    for (int i = 0; i < correctHash.Length; i++)
    {
        if (testHash[i] != correctHash[i])
            return false;
    }
    return true;
}
```

Минимум сторонних зависимостей

Race conditions in the code

```
public static void Main()
{
    Counter counter = new Counter();
    Task t1 = Task.Run(() => { for (int i = 0; i < 1000; i++) counter.Increment(); });
    Task t2 = Task.Run(() => { for (int i = 0; i < 1000; i++) counter.Increment(); });

    Task.WaitAll(t1, t2);
    Console.WriteLine(counter.Value); // Ожидаем 2000, но результат может быть меньше
}
```

Методы предотвращения Race Conditions in the code

- **lock:** Блокирует доступ к критической секции кода для одного потока.
- **Mutex:** Обеспечивает взаимное исключение для потоков, работающих с общими ресурсами.
- **Monitor:** Позволяет блокировать и отслеживать состояние объектов.
- **Interlocked:** Предоставляет атомарные операции для примитивных типов данных.

Race conditions in the DB level

```
public void CreditWallet(int walletId, decimal amount)
{
    using (var context = new MyDbContext())
    {
        var wallet = context.Wallets.Single(w => w.Id == walletId);
        wallet.Balance += amount;
        context.SaveChanges();
    }
}
```

Методы предотвращения Race Conditions in the DB level

- **Транзакции:** Группирование операций в атомарные единицы работы.
- **Блокировки:** Использование блокировок для синхронизации доступа к данным.
- **Оптимистическая конкуренция:** Проверка изменений перед выполнением обновлений.

Race conditions by design (неизбежные)

- Синхронизация данных между системами
- Распределенная обработка задач
- Асинхронная обработка данных

Методы смягчения неизбежных Race Conditions

- **Использование версионности данных:** Применение контрольных сумм или временных меток для проверки актуальности данных.
- **Retry логика:** Повторение операций при обнаружении состояния гонки.
- **Консистентность в конечном счете:** Прием, что данные будут согласованы в конечном счете, и разработка системы, устойчивой к временной несогласованности.
- **Централизованное управление данными:** Использование централизованных служб для управления доступом к данным.

Balance

1000 GBP

Send



Deposit

Account Name	Hapag-Lloyd (New Zealand)
Bank Name	Citibank, N.A.
Bank Address	23 Customs Street East, Auckland
Account Number	3161056
Swift Address	CITINZ2X
Swift Code	CITIUS33
Intermediary	
Currency	USD

HOLDER NAME:

SURNAME:

ENTER YOUR CREDIT CARD NUMBER:

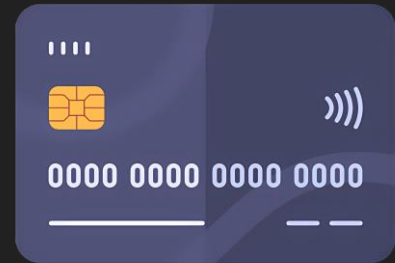
  EXPIRES ON:



Application



Bank



Cards provider



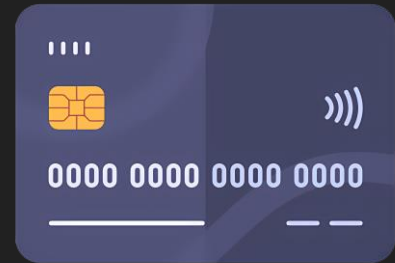
Application

Balance: £1000



Bank

Balance: £1000



Cards provider

Balance: £1000

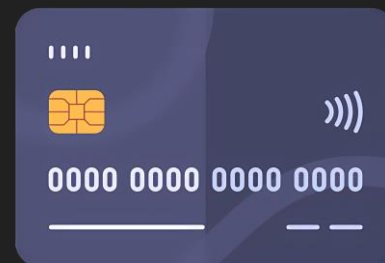
Transaction Web Hook



Bank

Balance: £1000

Transaction Web Hook



Cards provider

Balance: £1000



Application

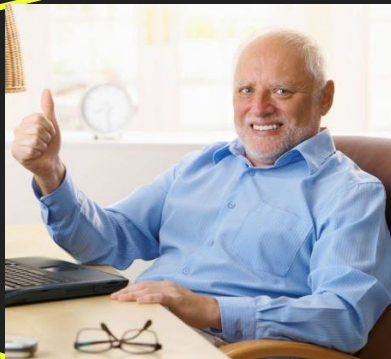
Balance: £1000



Application

Balance: £1000

Transaction Web Hook

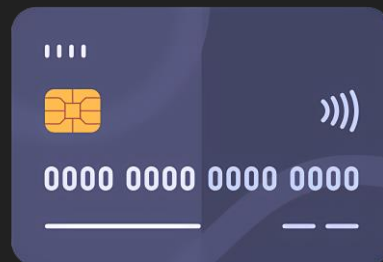


Transaction Web Hook



Bank

Balance: £1000



Cards provider

Balance: £1000

ФИШИНГ

UPBIT
to me

<upbit_sg@upbit.com>

Sep 12, 2019, 3:14 PM



UPBIT LISTING PROPOSAL

Hello **Team**

We are reaching out to you considering getting your project listed with us.

UPBIT has assessed your coin and has decided to contact you due to that, cause your coin seems like it will yield and grow better in terms of project awareness and coin popularity after listing with us

KINDLY FILL OUR LISTING FORM PROVIDED BELOW SO WE CAN GET THE REQUIRED DETAILS NEEDED ABOUT YOUR PROJECT

<https://forms.gle/Uyf3HuijsREa8wUy5>

UPBIT PROPOSAL

External



UPBIT <upbit_sg@upbit.com>
to me ▾

Sep 12, 2019, 3:14 PM



← Reply

→ Forward

≡ Filter messages like this

🖨 Print

🗑 Delete this message

🚫 Block "UPBIT"

⚠ Report spam

👤 Report phishing

🚫 Report illegal content

<> Show original

🗨 Translate message

⬇ Download message

✉ Mark as unread

UPBIT LISTING PROPOSAL

Hello **Team**

We are reaching out to you considering getting your project listed with us.

UPBIT has assessed your coin and has decided to contact you due to that, cause your coin seems like it will yield and grow better in terms of project awareness and coin popularity.

Original Message

Message ID	<7c46b8139c3f022f3e8fe20588576888@upbit.com>
Created at:	Thu, Sep 12, 2019 at 3:14 PM (Delivered after 1 second)
From:	UPBIT <upbit_sg@upbit.com>
To:	
Subject:	UPBIT PROPOSAL
SPF:	SOFTFAIL with IP 173.201.192.110 Learn more
DMARC:	'FAIL' Learn more

[Download Original](#)

[Copy to clipboard](#)

Delivered-To:

Received: by 2002:a67:f8d4:0:0:0:0:0 with SMTP id c20csp2109352vsp;

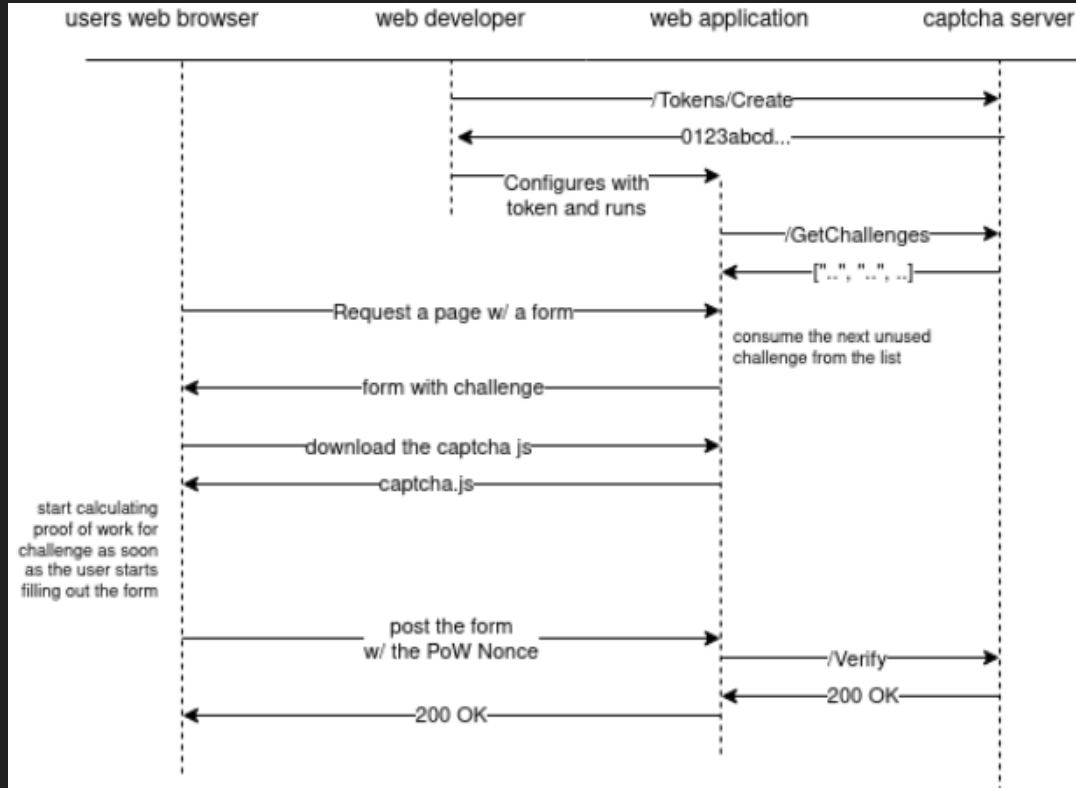
Thu, 12 Sep 2019 05:14:44 -0700 (PDT)

X-Google-Smtp-Source: APXvYqwANUViC4T3kZokzJiE3No6ezBRrg3jJTZtEaFDvo7Upj07h8jG/9d5BcQDOyWvQyAiM1W+

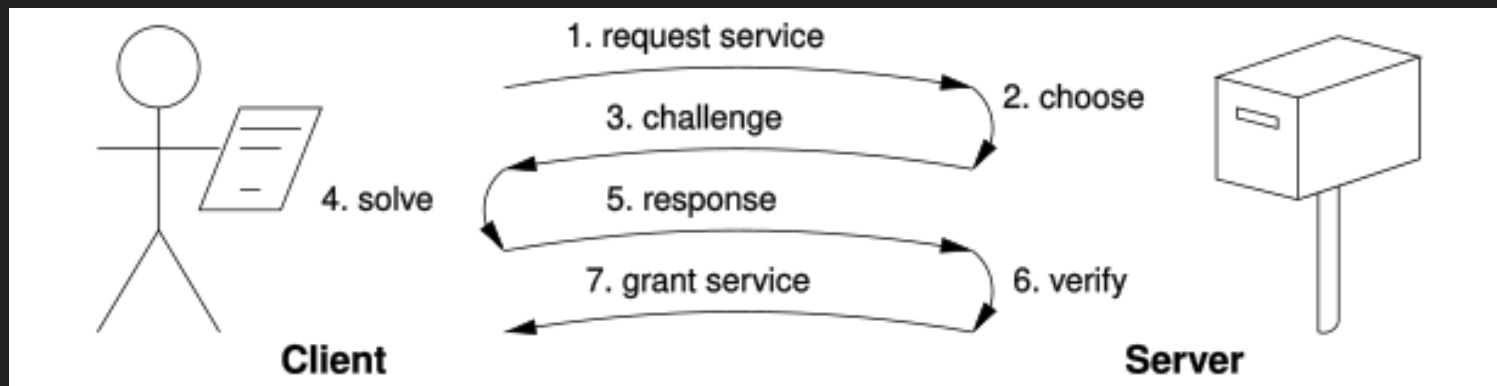
Перебор по базам паролей. Капчи

Атака по словарю основана на использовании списка общих паролей или паролей, полученных из предыдущих утечек данных. Злоумышленники автоматизируют процесс проверки этих паролей на сайте, чтобы найти совпадения и получить доступ к учетным записям.

Proof Of Work Captcha



Proof Of Work алгоритм



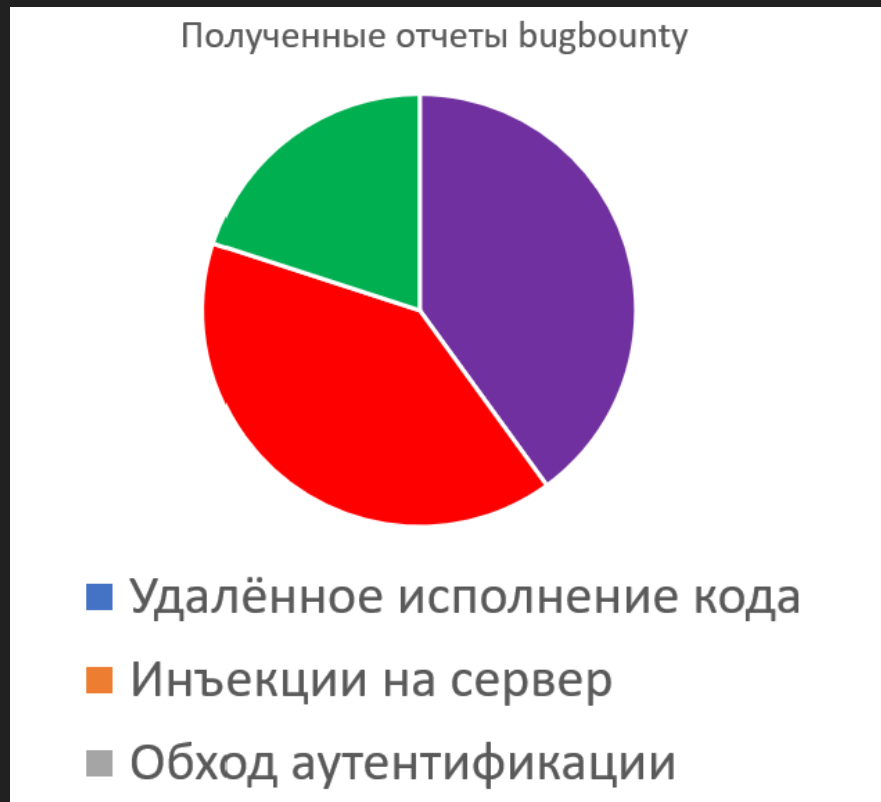
Proof Of Work алгоритм

- Хеш функция

$$F(x) = 9F291846...47A32C$$

- Клиент получает Nonce (набор случайных символов)
- Клиент должен подобрать такой Postfix, чтобы хеш функция давала значение с N нулей с начале
$$F(\text{Nonce} + \text{Postfix}) = 0000000... ..47A32C$$
- Подобранный Postfix является доказательством выполненной работы, то есть решённой капчи

Bounty программы



Спасибо за внимание!

Ваши вопросы?