QIWI

contactpay

Векторы атак: взгляд разработчика

2023

Содержание презентации

1. Что известно о безопасности
2. Жили-были уязвимости
3. Превентивные меры

contactpay

Что известно о безопасности

- Open Web Application Security Project (OWASP)
- Payment Card Industry Data Security Standard (PCI DSS)
- Security Operations Center (SOC)
- и другие...

contactpay

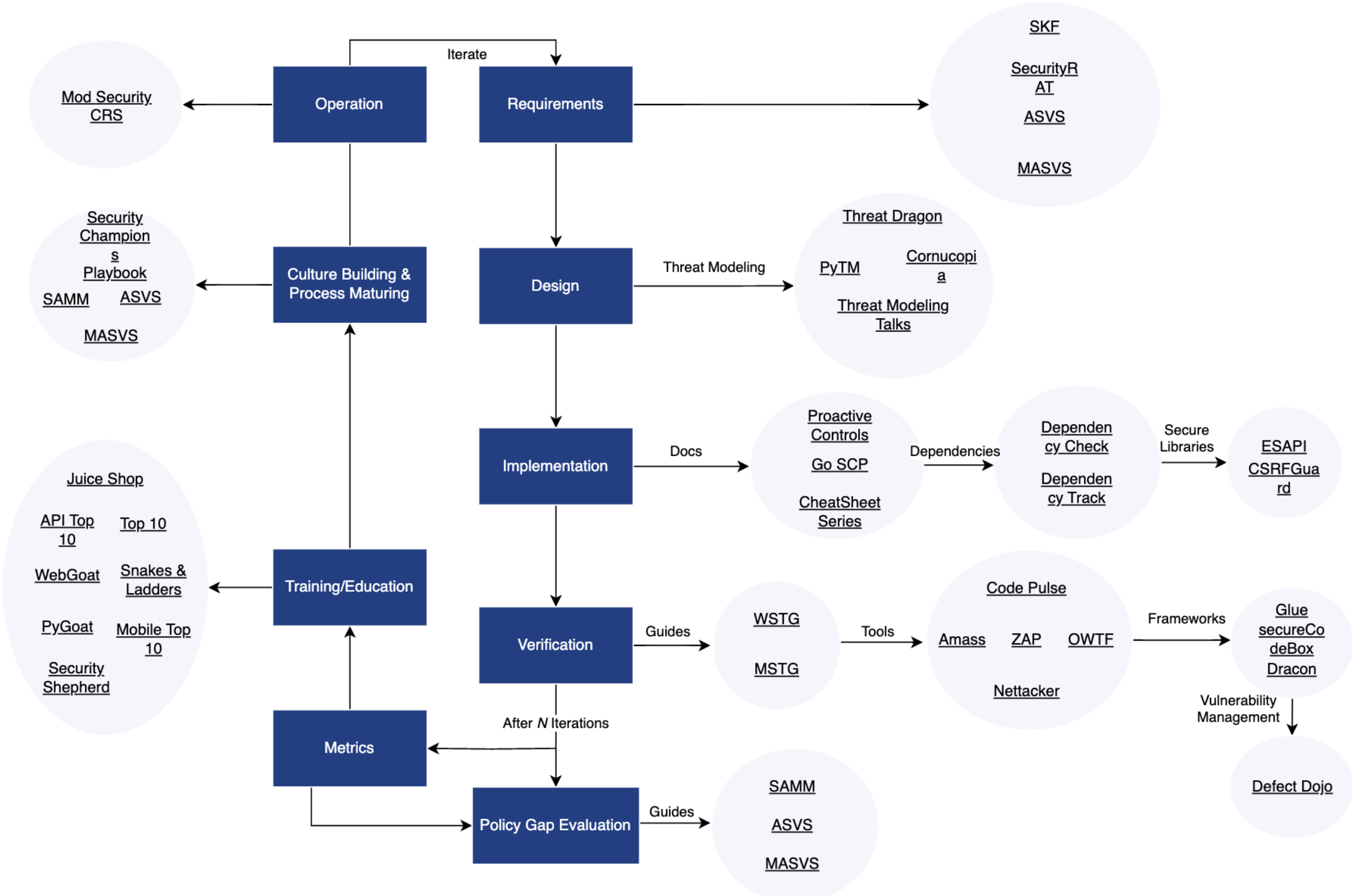
Open Web Application Security Project (OWASP)

OWASP

contactpay

293 проекта

contactpay


Iterate
Operation
Requirements
Design
Implementation
Verification
Metrics
Policy Gap Evaluation
Training/Education
Culture Building & Process Maturing
Mod Security
CRS
SKF
SecurityRAT
ASVS
MASVS
Threat Modeling
Threat Dragon
PyTM
Cornucopia
Threat Modeling Talks
Docs
Proactive Controls
Go SCP
CheatSheet Series
Dependencies
Dependency Check
Dependency Track
Secure Libraries
ESAPI
CSRFGuard
Guides
WSTG
MSTG
Tools
Code Pulse
Amass
ZAP
OWTF
Nettacker
Frameworks
Glue
secureCodeBox
Dracon
Vulnerability Management
Defect Dojo
After N Iterations
Guides
SAMM
ASVS
MASVS
Security Champions Playbook
SAMM
ASVS
MASVS
Juice Shop
API Top 10
Top 10
WebGoat
Snakes & Ladders
PyGoat
Mobile Top 10
Security Shepherd

https://owasp.org/projects/

OWASP Projects, the SDLC, and the Security Wayfinder

contactpay

Open CRE

Search... Search

CSRF protection

CRE: 028-727

Which is part of CREs:

- CRE : 546-564 : Cross-cutting concerns

Which is related to CREs:

- CRE : 060-472 : Use CSRF protection against authenticated functionality, add anti-automation controls for unauthenticated functionality

Which is linked to sources:

- Standard : ASVS : V4.2.2 : Verify that the application or framework enforces a strong anti-CSRF mechanism to protect authenticated functionality, and effective anti-automation or anti-CSRF protects unauthenticated functionality.
- Standard : CWE : 352
- Standard : OWASP Cheat Sheets : Authorization Testing Automation Cheat Sheet
- Standard : OWASP Cheat Sheets : Cross-Site Request Forgery Prevention Cheat Sheet
- Standard : OWASP Cheat Sheets : Insecure Direct Object Reference Prevention Cheat Sheet
- Standard : OWASP Web Security Testing Guide (WSTG) : WSTG-SESS-05
- Tool : ZAP Rule : 10202 : Absence of Anti-CSRF Tokens
- Tool : ZAP Rule : 20012 : Anti-CSRF Tokens Check

Which has been automatically mapped to:

- Standard : CAPEC : 3.9 : 111 : JSON Hijacking (aka JavaScript Hijacking)
- Standard : CAPEC : 3.9 : 462 : Cross-Domain Search Timing
- Standard : CAPEC : 3.9 : 467 : Cross Site Identification
- Standard : CAPEC : 3.9 : 62 : Cross Site Request Forgery

contactpay

https://www.opencre.org/

OpenCRE

contactpay

OWASP Top Ten

- OWASP API Security Top 10
- OWASP Cloud-Native Application Security Top 10
- OWASP Desktop App Security Top 10
- OWASP Docker Top 10
- OWASP Low-Code/No-Code Top 10
- OWASP Machine Learning Security Top Ten
- OWASP Mobile Top 10
- OWASP Top 10
- OWASP Top 10 CI/CD Security Risks
- OWASP Top 10 Client-Side Security Risks
- OWASP Top 10 Privacy Risks
- OWASP Serverless Top 10

contactpay

OWASP API Security Project

1. Broken Object Level Authorization
2. Broken Authentication
3. Broken Object Property Level Authorization NEW!
4. Unrestricted Resource Consumption NEW!
5. Broken Function Level Authorization
6. Unrestricted Access to Sensitive Business Flows NEW!
7. Server Side Request Forgery NEW!
8. Security Misconfiguration
9. Improper Inventory Management NEW!
10. Unsafe Consumption of APIs NEW!

contactpay

OWASP API Security Project

1. Broken Object Level Authorization
2. Broken Authentication
3. Broken Object Property Level Authorization 

4. Unrestricted Resource Consumption
5. Broken Function Level Authorization
6. Unrestricted Access to Sensitive Business Flows
7. Server Side Request Forgery
8. Security Misconfiguration
9. Improper Inventory Management
10. Unsafe Consumption of APIs

contactpay

OWASP API Security Project

1. Broken Object Level Authorization
2. Broken Authentication
3. Broken Object Property Level Authorization
4. Unrestricted Resource Consumption 

5. Broken Function Level Authorization
6. Unrestricted Access to Sensitive Business Flows
7. Server Side Request Forgery
8. Security Misconfiguration
9. Improper Inventory Management
10. Unsafe Consumption of APIs

OWASP API Security Project

1. Broken Object Level Authorization
2. Broken Authentication
3. Broken Object Property Level Authorization
4. Unrestricted Resource Consumption
5. Broken Function Level Authorization
6. Unrestricted Access to Sensitive Business Flows 

7. Server Side Request Forgery
8. Security Misconfiguration
9. Improper Inventory Management
10. Unsafe Consumption of APIs

contactpay

OWASP API Security Project

1. Broken Object Level Authorization
2. Broken Authentication
3. Broken Object Property Level Authorization
4. Unrestricted Resource Consumption
5. Broken Function Level Authorization
6. Unrestricted Access to Sensitive Business Flows
7. Server Side Request Forgery 

8. Security Misconfiguration
9. Improper Inventory Management
10. Unsafe Consumption of APIs

contactpay

OWASP API Security Project

1. Broken Object Level Authorization
2. Broken Authentication
3. Broken Object Property Level Authorization
4. Unrestricted Resource Consumption
5. Broken Function Level Authorization
6. Unrestricted Access to Sensitive Business Flows
7. Server Side Request Forgery
8. Security Misconfiguration
9. Improper Inventory Management
10. Unsafe Consumption of APIs

OWASP API Security Project

1. Broken Object Level Authorization
2. Broken Authentication
3. Broken Object Property Level Authorization
4. Unrestricted Resource Consumption
5. Broken Function Level Authorization
6. Unrestricted Access to Sensitive Business Flows
7. Server Side Request Forgery
8. Security Misconfiguration
9. Improper Inventory Management 

10. Unsafe Consumption of APIs

contactpay

OWASP API Security Project

1. Broken Object Level Authorization
2. Broken Authentication
3. Broken Object Property Level Authorization
4. Unrestricted Resource Consumption
5. Broken Function Level Authorization
6. Unrestricted Access to Sensitive Business Flows
7. Server Side Request Forgery
8. Security Misconfiguration
9. Improper Inventory Management
10. Unsafe Consumption of APIs 

contactpay

https://owasp.org/API-Security/editions/2023/en/0x00-header/

OWASP API Security Project

contactpay

Взять на вооружение

- OpenCRE
- OWASP Top Ten
- OWASP Cheat Sheet Series

https://cheatsheetseries.owasp.org/

OWASP Cheat Sheet Series

contactpay

Payment Card Industry Data Security Standard (PCI DSS)

PCI DSS COMPLIANT

contactpay

Чего хочет PCI DSS

- Блокирующий Code Review
- Аудит доступов
- Инвентаризация (доступов, сервис-провайдеров и т.п.)
- ASV (Approved Scanning Vendors) сканирование
- Шифрование чувствительных данных
- WAF (Web Application Firewall)
- Парольные политики
- Анализ уязвимостей инфраструктурных компонентов и приложений
- Аудит логов на наличие чувствительных данных
- Инцидентная система

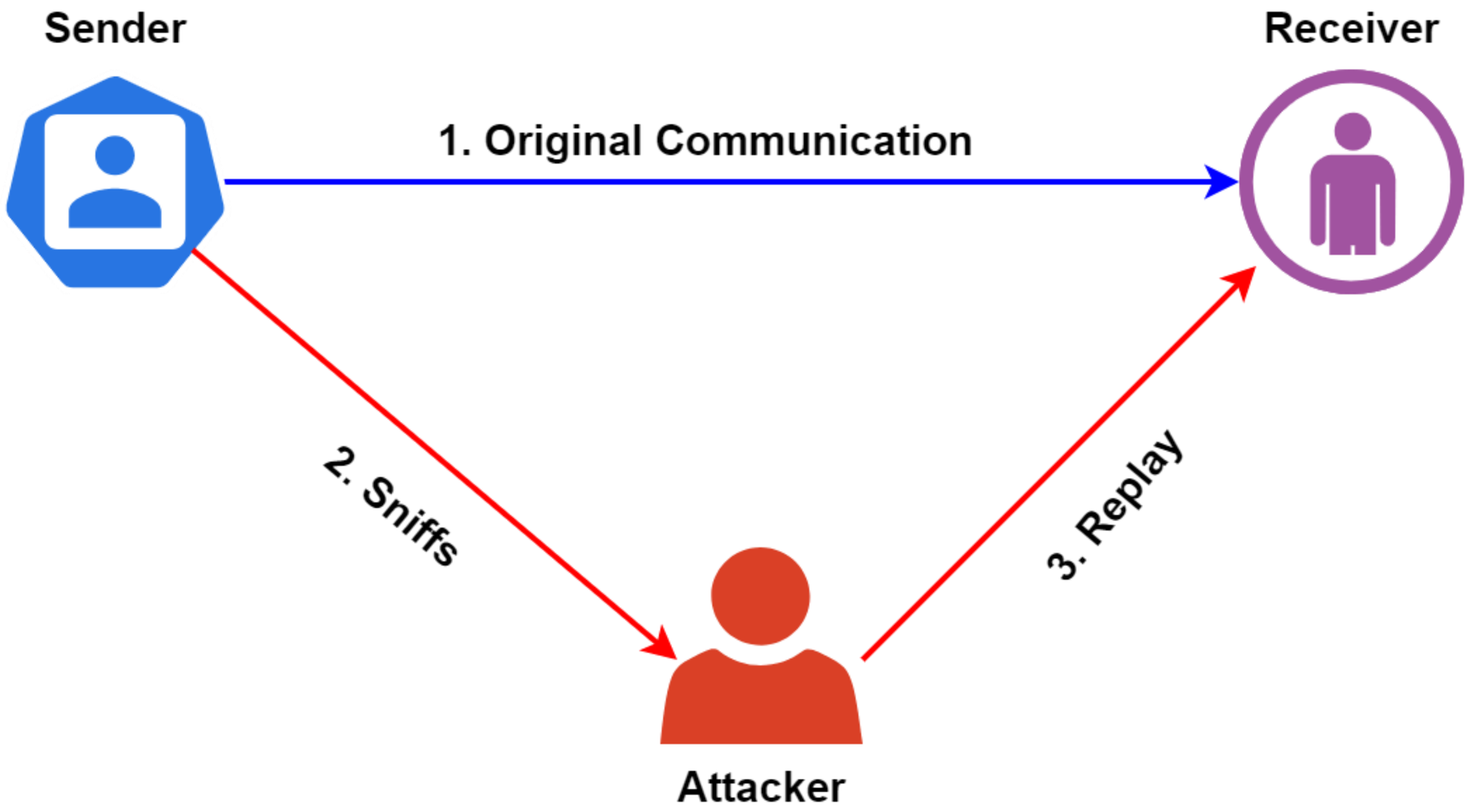
contactpay

Это хороший API?

```
curl --location 'https://api.example.com/payout/' \
--header 'Authorization: Bearer VERY_SECRET_TOKEN' \
--data '{
  "amount": "110.01",
  "currency": "EUR",
  "account": "79991112233"
}'
```


contactpay

Replay attack



contactpay

А так?

```
curl --location 'https://api.example.com/payout/' \
--header 'Authorization: Bearer VERY_SECRET_TOKEN' \
--data '{
  "order_id": "84b1b9a3-faac-4b3a-9ec8-2fdadc309242",
  "amount": "110.01",
  "currency": "EUR",
  "account": "79991112233"
}'
```


contactpay

```python
import hashlib

signature_keys = (
    'account',
    'amount',
    'currency',
)

string_to_sign = '79991112233:110.01:EUR:very_secret_token'

sign = hashlib.sha256(string_to_sign.encode()).hexdigest()
print(sign)

>> '87d0376e72fe71c76d888d2d51520c9adffad6410214a596b912e392f3dfd056'
```

```
curl --location 'http://api.example.com/payout/' \
--header 'X-Signature: 87d0376e72fe71c76d888d2d51520c9adffad6410214a596b912e392f3dfd056' \
--data '{
    "order_id": "84b1b9a3-faac-4b3a-9ec8-2fdadc309242",
    "amount": "110.01",
    "currency": "EUR",
    "account": "79991112233"
}'
```

```python
import hashlib
from datetime import datetime

signature_keys = (
    'account',
    'amount',
    'currency',
    'order_id',
)

timestamp = int(round(datetime.now().timestamp()))

string_to_sign = '79991112233:110.01:EUR:1694414344:84b1b9a3-faac-4b3a-9ec8-2fdadc309242very_secret_token'

sign = hashlib.sha256(string_to_sign.encode()).hexdigest()
print(sign)

>> '1afd677afc50cdc4d8f0cc8b2037c24930e1c035fd41553072cbd0da3a50c78d'
```

Сначала не понял, а потом как понял

contactpay

Certificate Transparency (RFC 9162)

https://datatracker.ietf.org/doc/html/rfc9162

Certificate Transparency Version 2.0 (RFC 9162)

contactpay

crt.sh Identity Search

Group by Issuer

Criteria	Type: Identity Match: ILIKE Search: 'apple.com'

Certificates

crt.sh ID	Logged At ⇧	Not Before	Not After	Common Name	Matching Identities	Issuer Name
2382386751	2020-01-27	2013-08-09	2015-08-09	Matt Martin-MPKI SSL-Premium	mattmartin@apple.com	C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, CN=VeriSign Class 3 Managed PKI Administrator CA - G3
2382089966	2020-01-27	2013-09-23	2015-09-24	food.apple.com	food.apple.com	C=US, O="Entrust, Inc.", OU=www.entrust.net/rpa is incorporated by reference, OU="(c) 2009 Entrust, Inc.", CN=Entrust Certification Authority - L1C
2381996484	2020-01-27	2014-07-22	2015-07-22	ecommerce-qa.apple.com	ecommerce-qa.apple.com	C=US, O="Entrust, Inc.", OU=www.entrust.net/rpa is incorporated by reference, OU="(c) 2009 Entrust, Inc.", CN=Entrust Certification Authority - L1C
2381996432	2020-01-27	2014-05-05	2015-05-05	b2b-test.apple.com	b2b-test.apple.com	C=US, O="Entrust, Inc.", OU=www.entrust.net/rpa is incorporated by reference, OU="(c) 2009 Entrust, Inc.", CN=Entrust Certification Authority - L1C
2381422119	2020-01-27	2014-09-11	2015-09-12	afsportal2.euro.apple.com	afsportal2.euro.apple.com	C=US, O="Entrust, Inc.", OU=www.entrust.net/rpa is incorporated by reference, OU="(c) 2009 Entrust, Inc.", CN=Entrust Certification Authority - L1C
2381408260	2020-01-27	2011-04-11	2015-06-08	b2btest.apple.com	b2btest.apple.com	C=US, O="Entrust, Inc.", OU=www.entrust.net/rpa is incorporated by reference, OU="(c) 2009 Entrust, Inc.", CN=Entrust Certification Authority - L1C
2381374674	2020-01-27	2013-05-01	2015-05-02	hrweb-maint.apple.com	hrweb-maint.apple.com	C=US, O="Entrust, Inc.", OU=www.entrust.net/rpa is incorporated by reference, OU="(c) 2009 Entrust, Inc.", CN=Entrust Certification Authority - L1C
2381374708	2020-01-27	2014-07-25	2015-07-24	wdg02-uat.apple.com	sso-uat-nc.corp.apple.com wdg02-uat.apple.com	C=US, O="Entrust, Inc.", OU=www.entrust.net/rpa is incorporated by reference, OU="(c) 2009 Entrust, Inc.", CN=Entrust Certification Authority - L1C
2381374659	2020-01-27	2014-04-23	2015-04-24	hrweb-qa.apple.com	hrweb-qa.apple.com	C=US, O="Entrust, Inc.", OU=www.entrust.net/rpa is incorporated by reference, OU="(c) 2009 Entrust, Inc.", CN=Entrust Certification Authority - L1C
2380977932	2020-01-26	2014-03-04	2015-03-04	applechinawifi.apple.com	applechinawifi.apple.com	C=US, O="Entrust, Inc.", OU=www.entrust.net/rpa is incorporated by reference, OU="(c) 2009 Entrust, Inc.", CN=Entrust Certification Authority - L1C
2380927914	2020-01-26	2012-11-02	2014-11-02	ray.apple.com	ray.apple.com	C=US, O="Entrust, Inc.", OU=www.entrust.net/rpa is incorporated by reference, OU="(c) 2009 Entrust, Inc.", CN=Entrust Certification Authority - L1C
2380821719	2020-01-26	2012-05-10	2014-05-31	idesk.corp.apple.com	idesk.corp.apple.com	C=US, O="Entrust, Inc.", OU=www.entrust.net/rpa is incorporated by reference, OU="(c) 2009 Entrust, Inc.", CN=Entrust Certification Authority - L1C
2380318841	2020-01-26	2012-09-22	2014-10-09	courier.sandbox.push.apple.com	courier.sandbox.push.apple.com	C=US, O="Entrust, Inc.", OU=www.entrust.net/rpa is incorporated by reference, OU="(c) 2009 Entrust, Inc.", CN=Entrust Certification Authority - L1C
2380315969	2020-01-26	2010-04-13	2012-05-31	gateway.push.apple.com	gateway.push.apple.com	C=US, O="Entrust, Inc.", OU=www.entrust.net/rpa is incorporated by reference, OU="(c) 2009 Entrust, Inc.", CN=Entrust Certification Authority - L1C
2380315852	2020-01-26	2011-08-16	2012-10-17	webmail.euro.apple.com	webmail.euro.apple.com	C=US, O="Entrust, Inc.", OU=www.entrust.net/rpa is incorporated by reference, OU="(c) 2009 Entrust, Inc.", CN=Entrust Certification Authority -

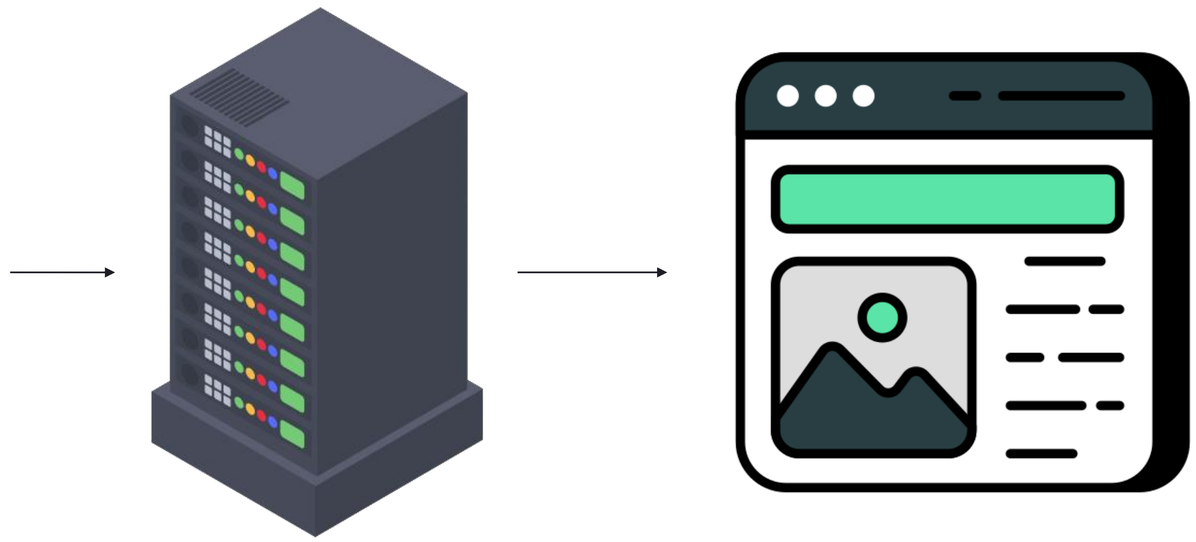
https://crt.sh

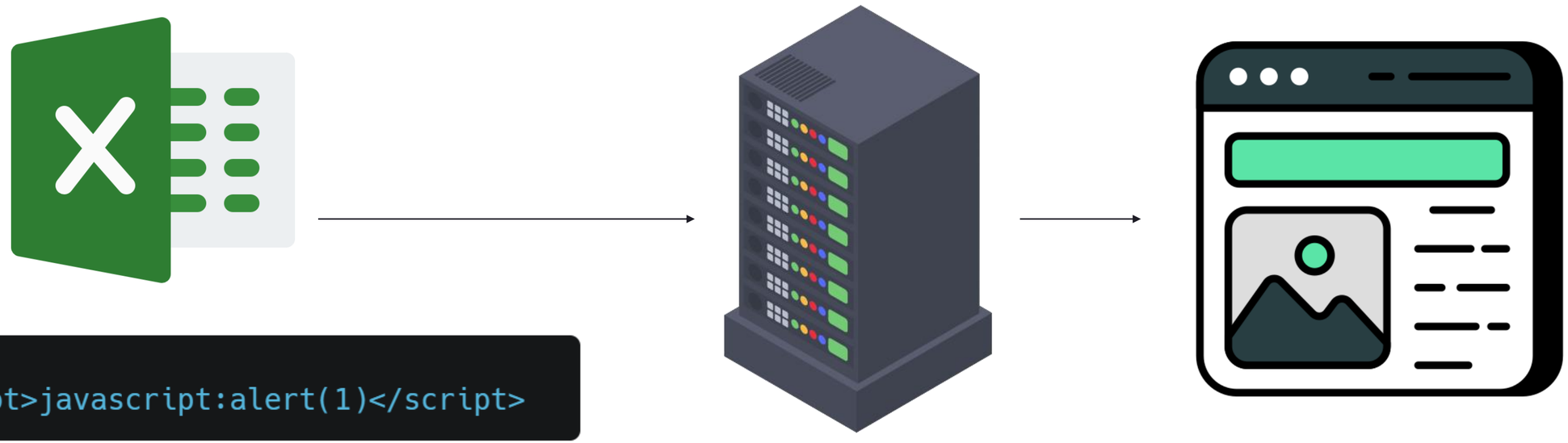
crt.sh
Certificate Search

contactpay

Cross Site Scripting (XSS)

```
'`"><\x3Cscript>javascript:alert(1)</script>
```





```
'`"><\x3Cscript>javascript:alert(1)</script>
```

Add advanced interaction controls to your HTML tables *the free & easy way*

Browser NPM

1 - Include these two files

```
CSS  //cdn.datatables.net/1.13.6/css/jquery.da
JS   //cdn.datatables.net/1.13.6/js/jquery.dat
```

2 - Initialise your DataTable:

```
1  let table = new DataTable('#myTable');
```

3 - You get a fully interactive table →

Full Getting Started Guide

Show 10 entries Search:

Name	Position	Office	Age	Start date
Airi Satou	Accountant	Tokyo	33	11/28/2008
Angelica Ramos	Chief Executive Officer (CEO)	London	47	10/9/2009
Ashton Cox	Junior Technical Author	San Francisco	66	1/12/2009
Bradley Greer	Software Engineer	London	41	10/13/2012
Brenden Wagner	Software Engineer	San Francisco	28	6/7/2011
Brielle Williamson	Integration Specialist	New York	61	12/2/2012
Bruno Nash	Software Engineer	London	38	5/3/2011
Caesar Vance	Pre-Sales Support	New York	21	12/12/2011
Cara Stevens	Sales Assistant	New York	46	12/6/2011
Cedric Kelly	Senior Javascript Developer	Edinburgh	22	3/29/2012
Name	**Position**	**Office**	**Age**	**Start date**

Showing 1 to 10 of 57 entries Previous 1 2 3 4 5 6 Next

```
{
    data: 'product',
    render: DataTable.render.text()
}
```

Content Security Policy (CSP)

```
Content-Security-Policy: default-src 'self'; form-action 'self'; object-src 'none'; frame-ancestors 'none'; upgrade-insecure-requests;
block-all-mixed-content

X-XSS-Protection: 0
```

Вывод

- Изучить подходящие вам проекты OWASP
- Взять на вооружение рекомендации стандартов
- Подходить к безопасности не только с точки зрения написания кода
- Важна регулярность и постоянство
- Помнить про особенности работы Интернета
- Никому не доверять

https://t.me/ozeranskii

QIWI

Сергей Озеранский

contactpay