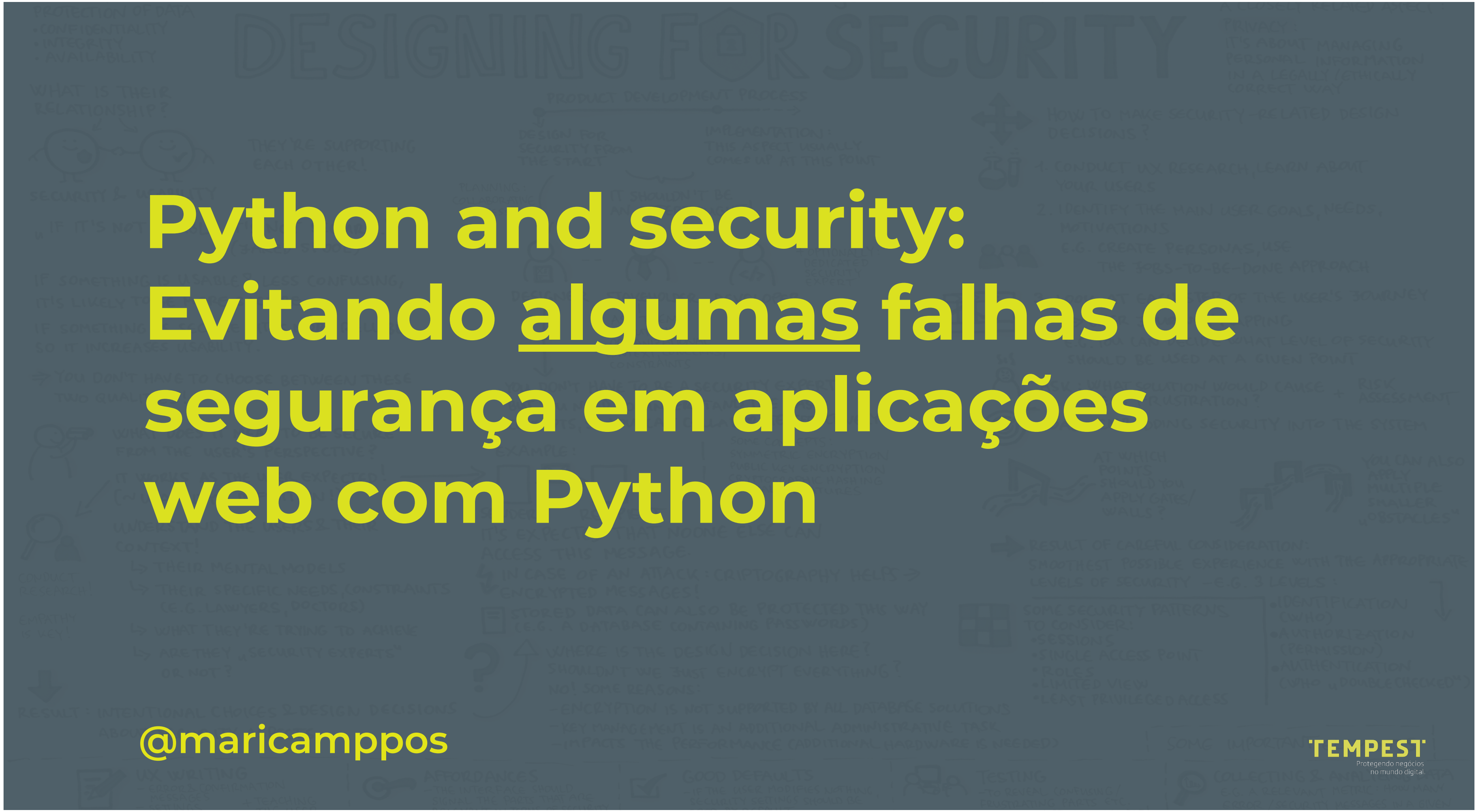
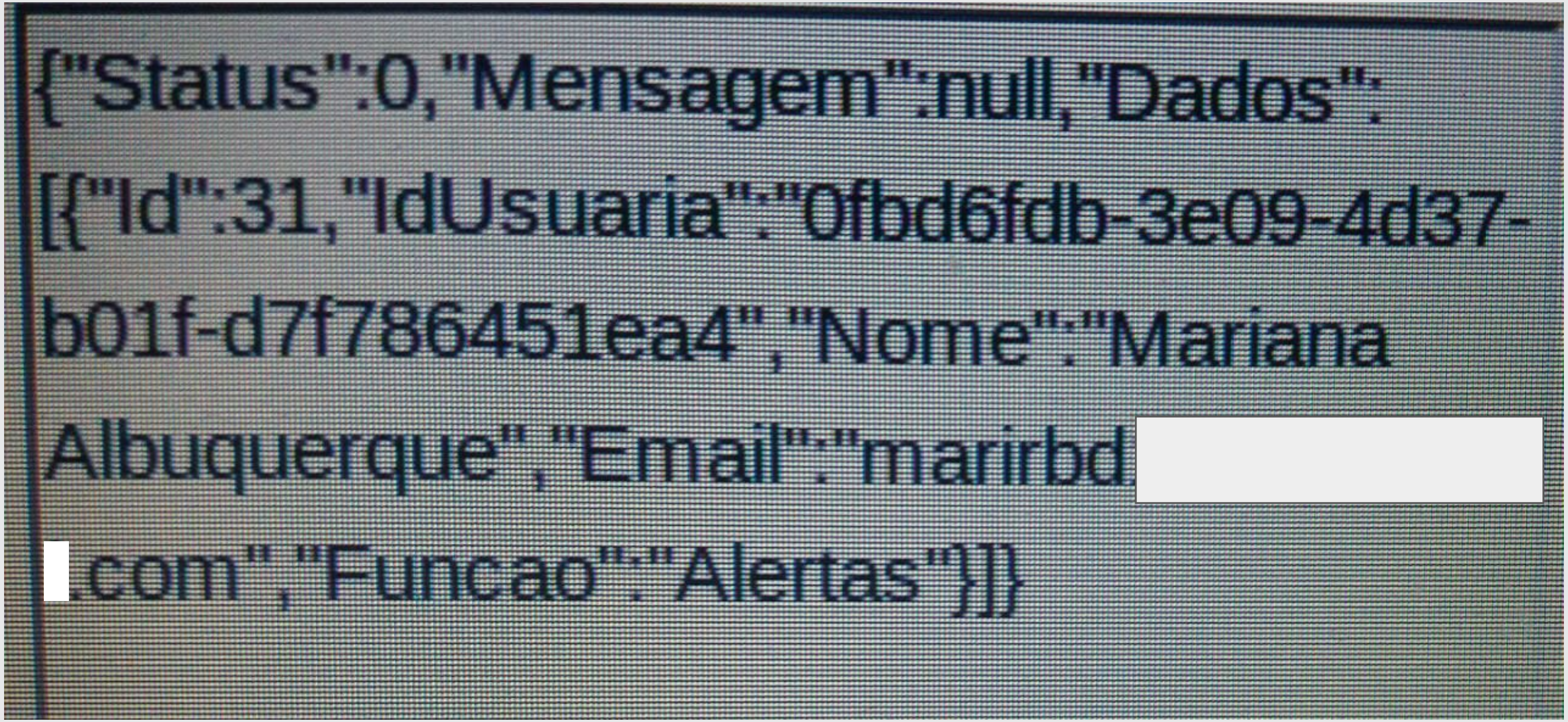


Python and security: Evitando algumas falhas de segurança em aplicações web com Python

@maricamppos

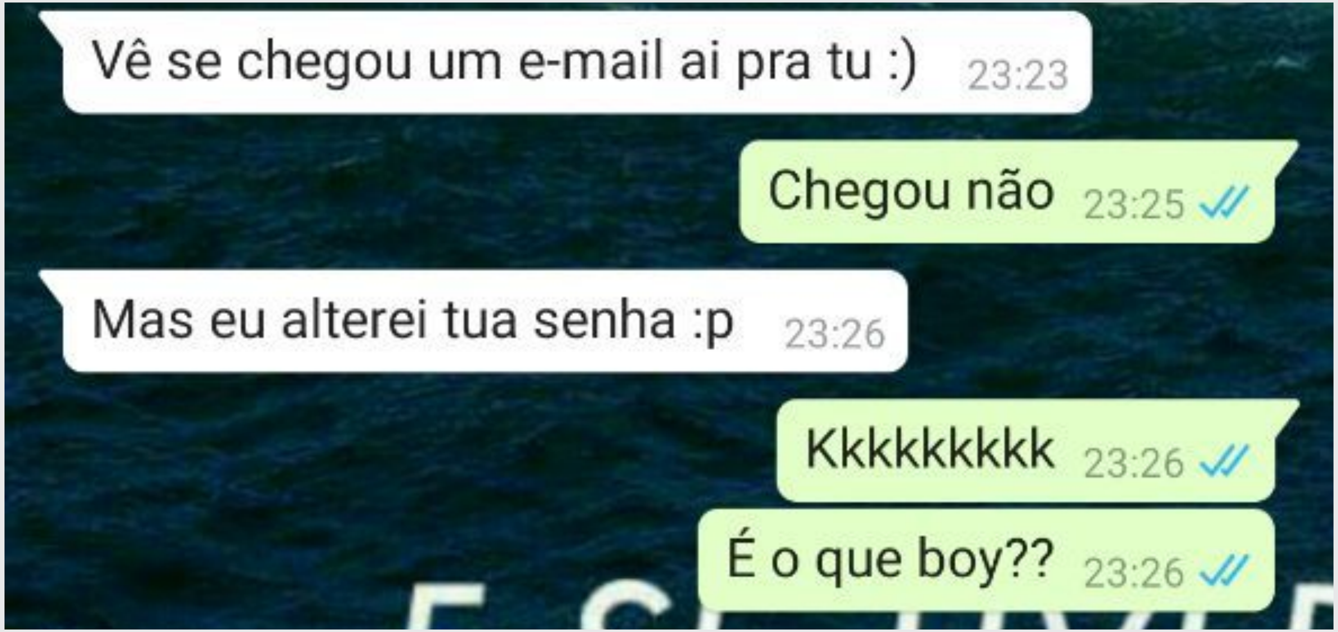
TEMPEST
Protegendo negócios no mundo digital.

FANTÁSTICO



```
{"Status":0,"Mensagem":null,"Dados":
[{"Id":31,"IdUsuaria":"0fbd6fdb-3e09-4d37-
b01f-d7f786451ea4","Nome":"Mariana
Albuquerque","Email":"marirbd[redacted]
.com","Funcao":"Alertas"}]}
```

FOX
NewGirl
hulu



Vê se chegou um e-mail ai pra tu :) 23:23

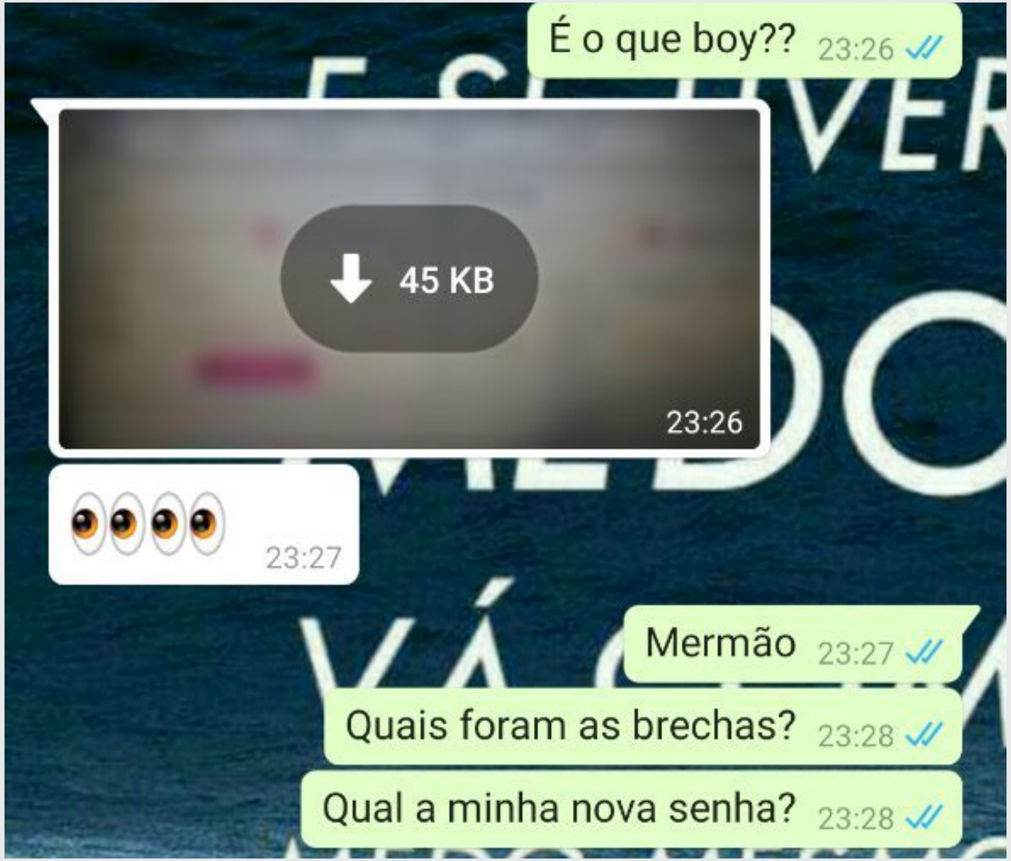
Chegou não 23:25

Mas eu alterei tua senha :p 23:26

Kkkkkkkkk 23:26

É o que boy?? 23:26

eOnMars



É o que boy?? 23:26

45 KB 23:26

👀👀 23:27

Mermão 23:27

Quais foram as brechas? 23:28

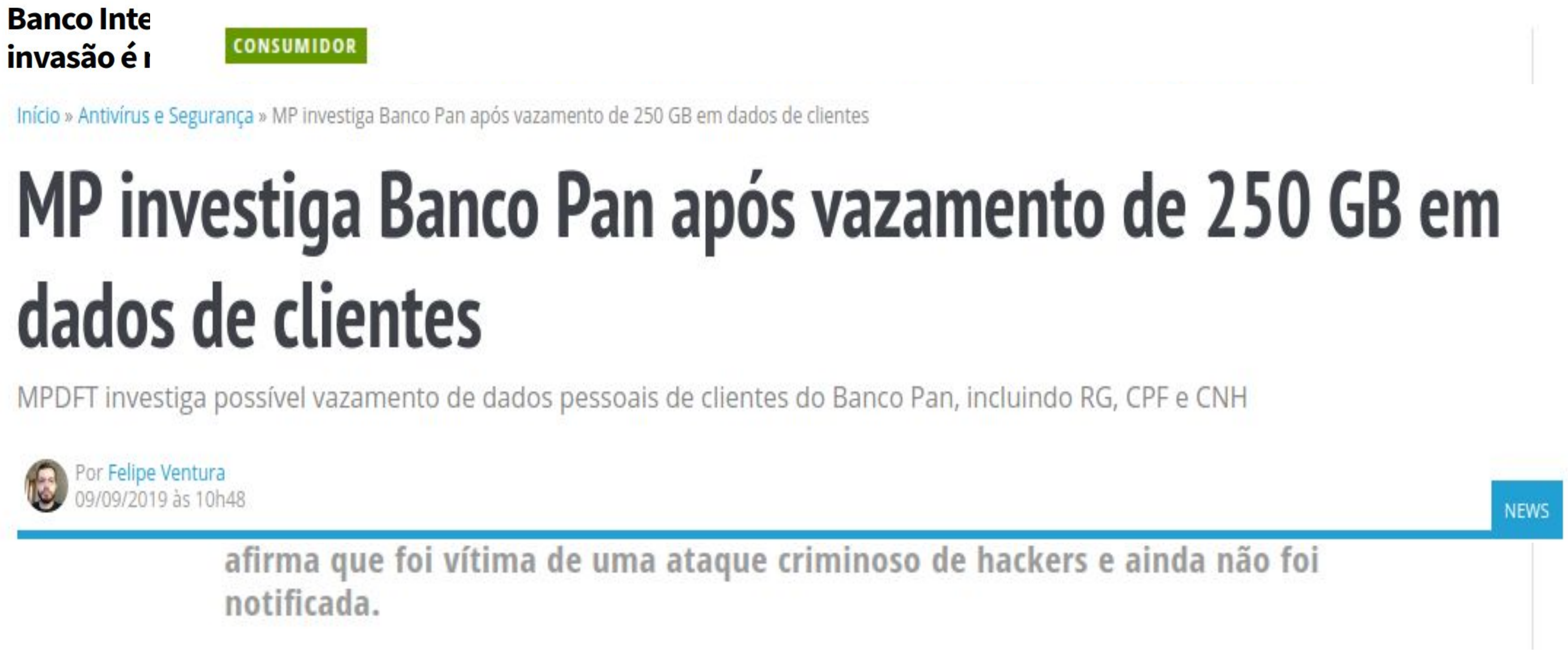
Qual a minha nova senha? 23:28

Ressalvas

.NET

Feito às pressas

Porque pensar em segurança?


Banco Inte
invasão é r

CONSUMIDOR

Início » Antivírus e Segurança » MP investiga Banco Pan após vazamento de 250 GB em dados de clientes

MP investiga Banco Pan após vazamento de 250 GB em dados de clientes

MPDFT investiga possível vazamento de dados pessoais de clientes do Banco Pan, incluindo RG, CPF e CNH

Por Felipe Ventura
09/09/2019 às 10h48

NEWS

afirma que foi vítima de uma ataque criminoso de hackers e ainda não foi notificada.

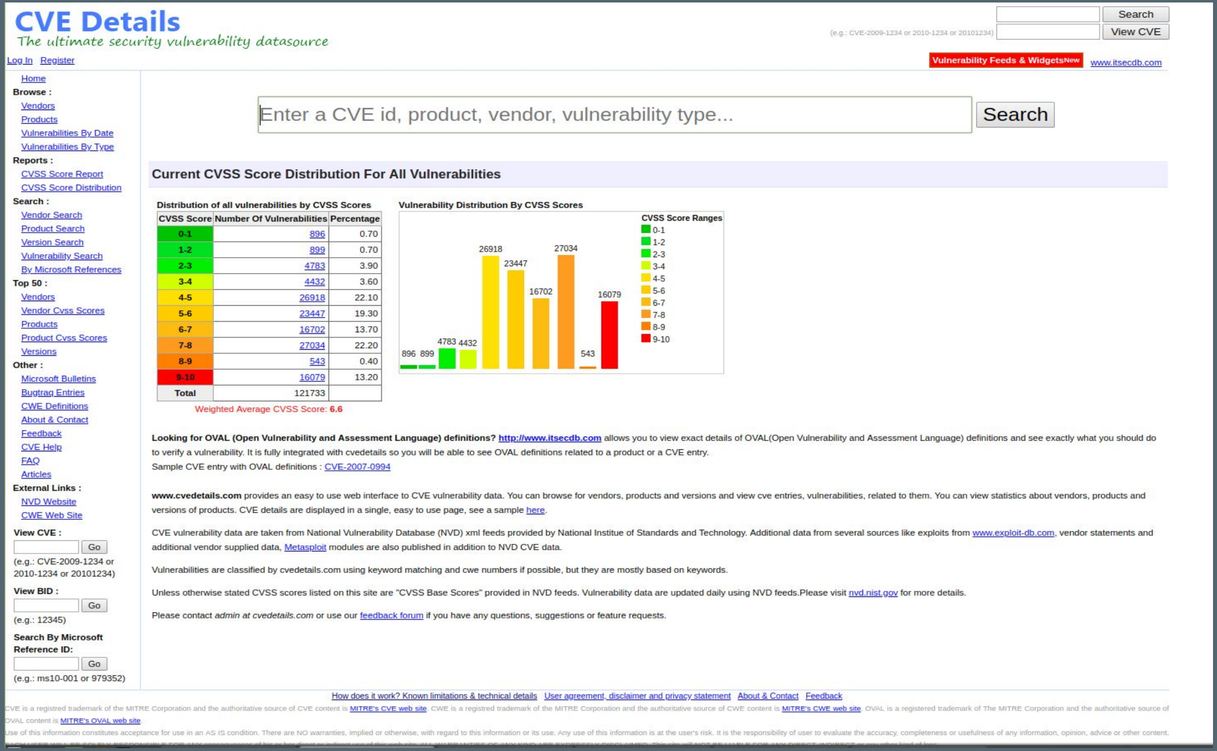
E nós como desenvolvedores web Python, devemos nos preocupar com isso?

Python é uma linguagem de programação segura?

O que devo fazer para evitar vulnerabilidades nas minhas aplicações?

Motivação

CVE Details (Commom Vulnerabilities and Exposures)


CVE Details
The ultimate security vulnerability datasource
Search
View CVE
(e.g.: CVE-2009-1234 or 2010-1234 or 20101234)
Log In
Register
Vulnerability Feeds & WidgetsNew
www.itsecdb.com
Enter a CVE id, product, vendor, vulnerability type...
Search

Home
Browse :
Vendors
Products
Vulnerabilities By Date
Vulnerabilities By Type
Reports :
CVSS Score Report
CVSS Score Distribution
Search :
Vendor Search
Product Search
Version Search
Vulnerability Search
By Microsoft References
Top 50 :
Vendors
Vendor Cvss Scores
Products
Product Cvss Scores
Versions
Other :
Microsoft Bulletins
Bugtraq Entries
CWE Definitions
About & Contact
Feedback
CVE Help
FAQ
Articles
External Links :
NVD Website
CWE Web Site

View CVE : Go
(e.g.: CVE-2009-1234 or 2010-1234 or 20101234)

View BID : Go
(e.g.: 12345)

Search By Microsoft Reference ID: Go
(e.g.: ms10-001 or 979352)

Current CVSS Score Distribution For All Vulnerabilities

Distribution of all vulnerabilities by CVSS Scores

CVSS Score	Number Of Vulnerabilities	Percentage
0-1	896	0.70
1-2	899	0.70
2-3	4783	3.90
3-4	4432	3.60
4-5	26918	22.10
5-6	23447	19.30
6-7	16702	13.70
7-8	27034	22.20
8-9	543	0.40
9-10	16079	13.20
Total	121733	

Weighted Average CVSS Score: **6.6**

Vulnerability Distribution By CVSS Scores

Looking for OVAL (Open Vulnerability and Assessment Language) definitions? http://www.itsecdb.com allows you to view exact details of OVAL(Open Vulnerability and Assessment Language) definitions and see exactly what you should do to verify a vulnerability. It is fully integrated with cvedetails so you will be able to see OVAL definitions related to a product or a CVE entry.
Sample CVE entry with OVAL definitions : CVE-2007-0994

www.cvedetails.com provides an easy to use web interface to CVE vulnerability data. You can browse for vendors, products and versions and view cve entries, vulnerabilities, related to them. You can view statistics about vendors, products and versions of products. CVE details are displayed in a single, easy to use page, see a sample here.

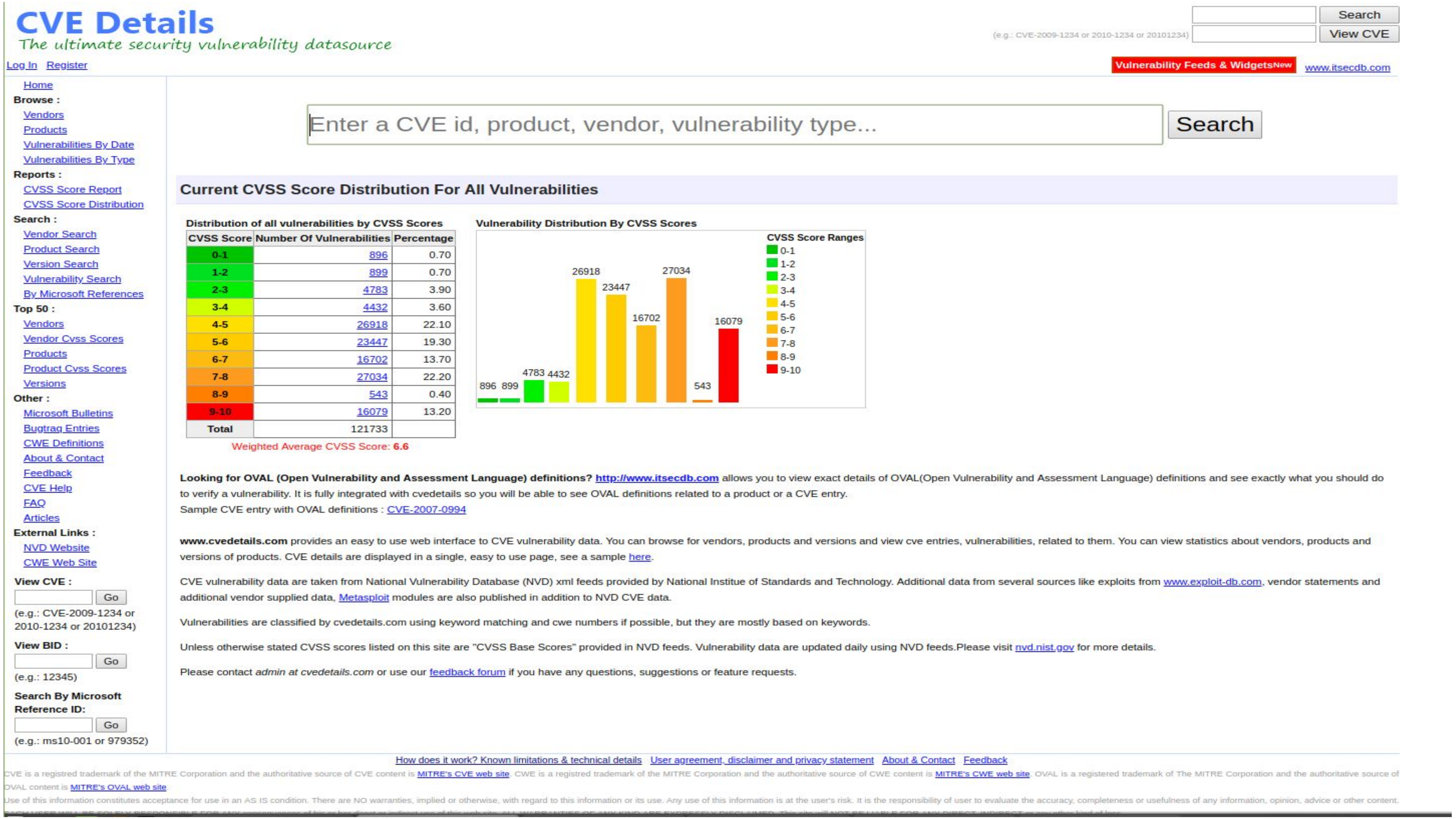
CVE vulnerability data are taken from National Vulnerability Database (NVD) xml feeds provided by National Institue of Standards and Technology. Additional data from several sources like exploits from www.exploit-db.com, vendor statements and additional vendor supplied data, Metasploit modules are also published in addition to NVD CVE data.

Vulnerabilities are classified by cvedetails.com using keyword matching and cwe numbers if possible, but they are mostly based on keywords.

Unless otherwise stated CVSS scores listed on this site are "CVSS Base Scores" provided in NVD feeds. Vulnerability data are updated daily using NVD feeds.Please visit nvd.nist.gov for more details.

Please contact *admin at cvedetails.com* or use our feedback forum if you have any questions, suggestions or feature requests.

How does it work? Known limitations & technical details   User agreement, disclaimer and privacy statement   About & Contact   Feedback



CVE Details

The ultimate security vulnerability datasource

Log In Register

Search

(e.g.: CVE-2009-1234 or 2010-1234 or 20101234) View CVE

Vulnerability Feeds & WidgetsNew www.itsecdb.com

Home

Browse :

- Vendors
- Products
- Vulnerabilities By Date
- Vulnerabilities By Type

Reports :

- CVSS Score Report
- CVSS Score Distribution

Search :

- Vendor Search
- Product Search
- Version Search
- Vulnerability Search
- By Microsoft References

Top 50 :

- Vendors
- Vendor Cvss Scores
- Products
- Product Cvss Scores
- Versions

Other :

- Microsoft Bulletins
- Bugtraq Entries
- CWE Definitions
- About & Contact
- Feedback
- CVE Help
- FAQ
- Articles

External Links :

- NVD Website
- CWE Web Site

View CVE :

Go

(e.g.: CVE-2009-1234 or 2010-1234 or 20101234)

View BID :

Go

(e.g.: 12345)

Search By Microsoft Reference ID:

Go

(e.g.: ms10-001 or 979352)

Enter a CVE id, product, vendor, vulnerability type... Search

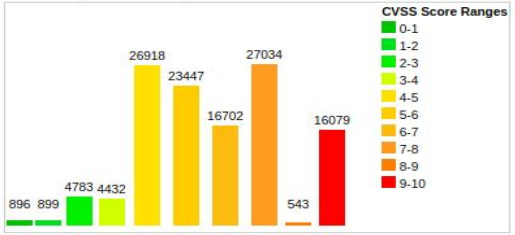
Current CVSS Score Distribution For All Vulnerabilities

Distribution of all vulnerabilities by CVSS Scores

CVSS Score	Number Of Vulnerabilities	Percentage
0-1	896	0.70
1-2	899	0.70
2-3	4783	3.90
3-4	4432	3.60
4-5	26918	22.10
5-6	23447	19.30
6-7	16702	13.70
7-8	27034	22.20
8-9	543	0.40
9-10	16079	13.20
Total	121733	

Weighted Average CVSS Score: **6.6**

Vulnerability Distribution By CVSS Scores



Looking for OVAL (Open Vulnerability and Assessment Language) definitions? http://www.itsecdb.com allows you to view exact details of OVAL(Open Vulnerability and Assessment Language) definitions and see exactly what you should do to verify a vulnerability. It is fully integrated with cvedetails so you will be able to see OVAL definitions related to a product or a CVE entry.
Sample CVE entry with OVAL definitions : CVE-2007-0994

www.cvedetails.com provides an easy to use web interface to CVE vulnerability data. You can browse for vendors, products and versions and view cve entries, vulnerabilities, related to them. You can view statistics about vendors, products and versions of products. CVE details are displayed in a single, easy to use page, see a sample here.

CVE vulnerability data are taken from National Vulnerability Database (NVD) xml feeds provided by National Institue of Standards and Technology. Additional data from several sources like exploits from www.exploit-db.com, vendor statements and additional vendor supplied data, Metasploit modules are also published in addition to NVD CVE data.

Vulnerabilities are classified by cvedetails.com using keyword matching and cwe numbers if possible, but they are mostly based on keywords.

Unless otherwise stated CVSS scores listed on this site are "CVSS Base Scores" provided in NVD feeds. Vulnerability data are updated daily using NVD feeds.Please visit nvd.nist.gov for more details.

Please contact *admin at cvedetails.com* or use our feedback forum if you have any questions, suggestions or feature requests.

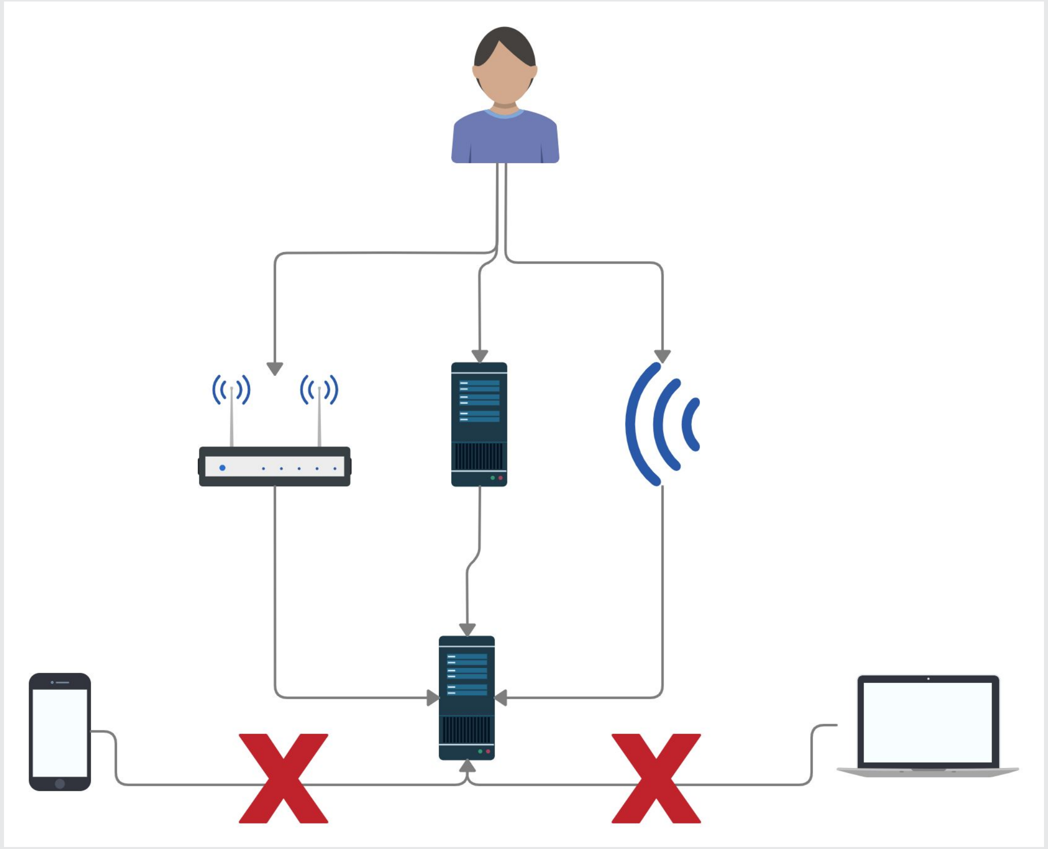
How does it work? Known limitations & technical details User agreement, disclaimer and privacy statement About & Contact Feedback

Vulnerabilidades por linguagens

Fonte: CVE Details

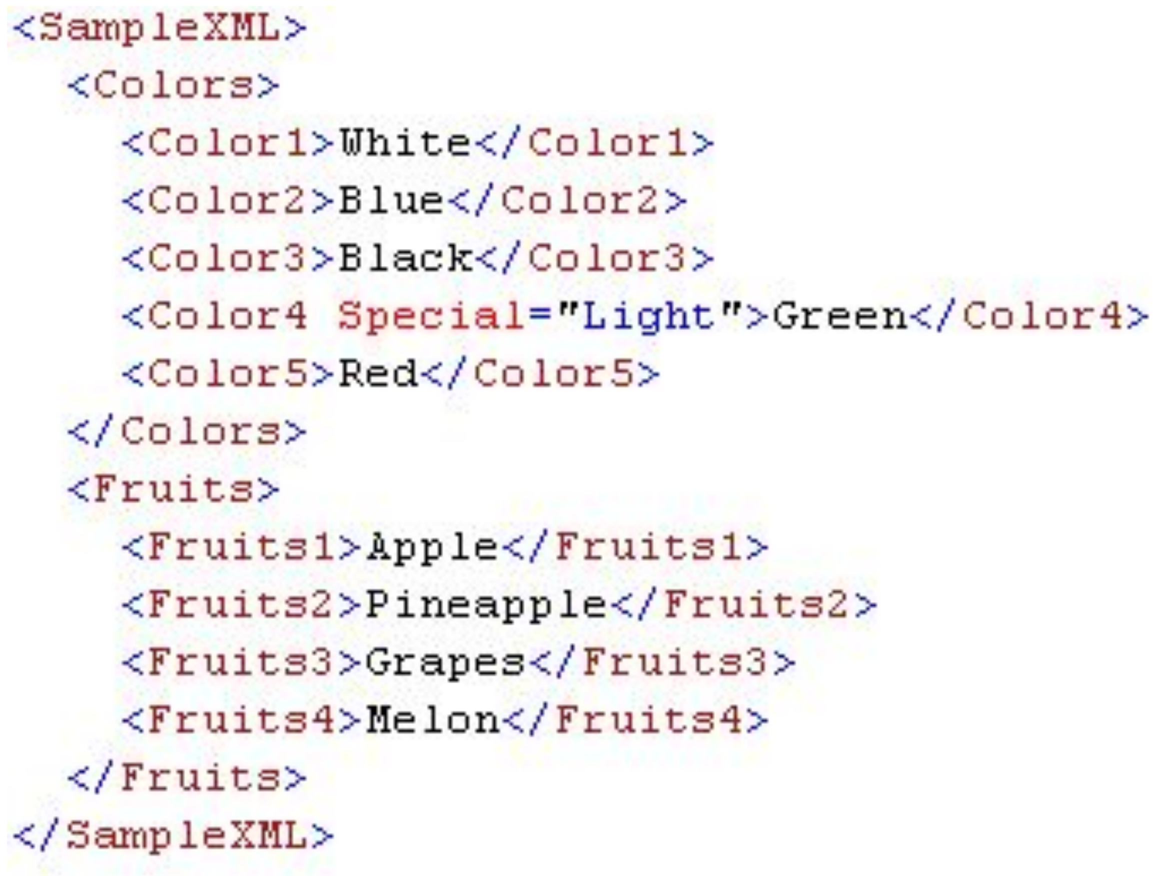
Linguagens/ Frameworks	2016	2017	2018	2019	Total
Python	5	3	8	6	22
Nodejs	9	19	18	2	48
Java	37	69	55	8	169
Golang	4	8	11	5	28

DoS

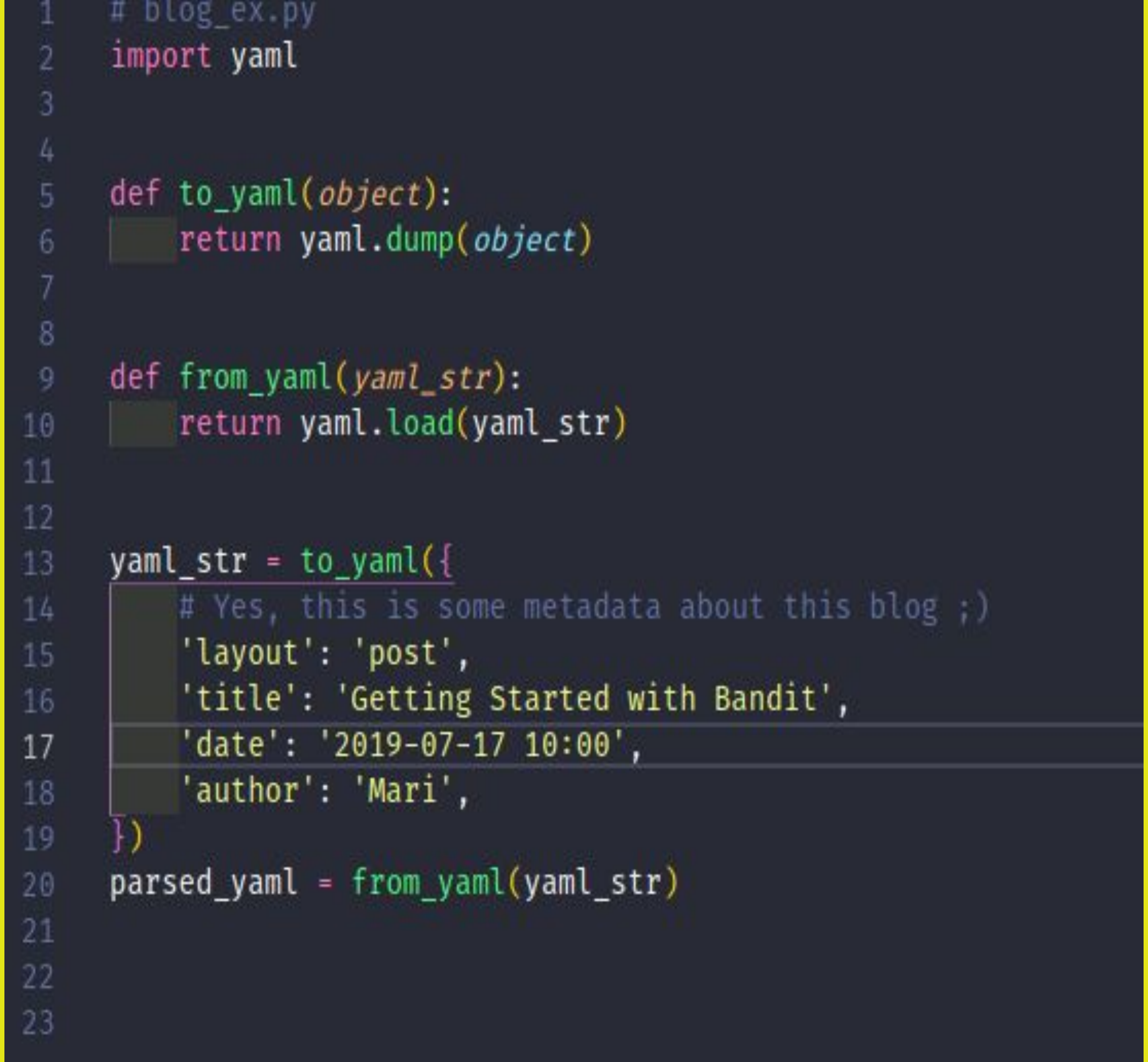


CVE-2018-14647

- ElementTree
- XML_SetHashSalt()
- Ataque: XML provocando uma colisão de hash na estrutura de dados

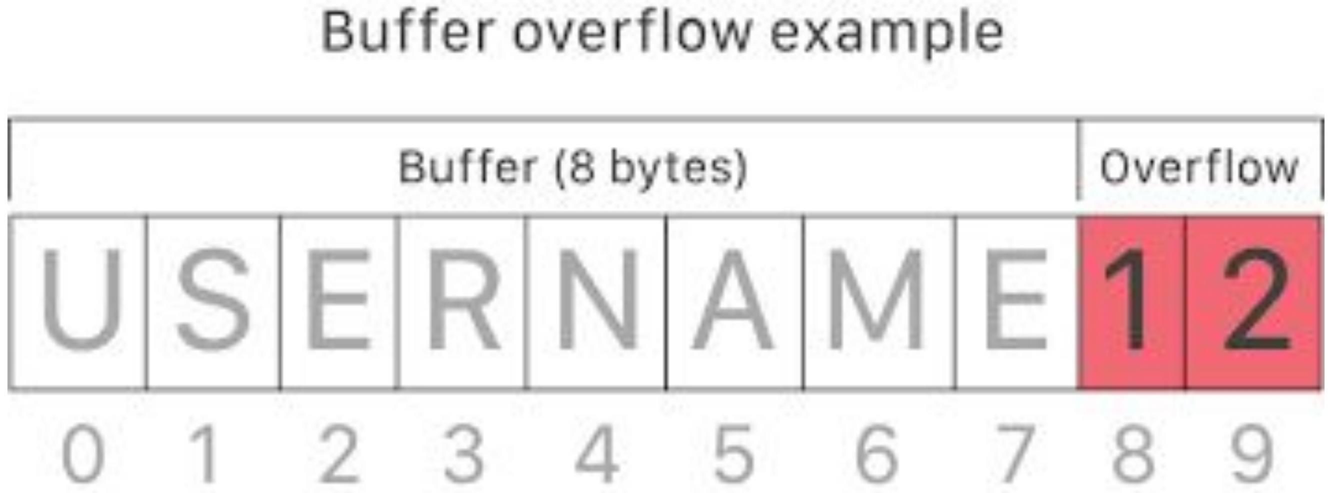


```
<SampleXML>
  <Colors>
    <Color1>White</Color1>
    <Color2>Blue</Color2>
    <Color3>Black</Color3>
    <Color4 Special="Light">Green</Color4>
    <Color5>Red</Color5>
  </Colors>
  <Fruits>
    <Fruits1>Apple</Fruits1>
    <Fruits2>Pineapple</Fruits2>
    <Fruits3>Grapes</Fruits3>
    <Fruits4>Melon</Fruits4>
  </Fruits>
</SampleXML>
```



```python
# blog_ex.py
import yaml


def to_yaml(object):
    return yaml.dump(object)


def from_yaml(yaml_str):
    return yaml.load(yaml_str)


yaml_str = to_yaml({
    # Yes, this is some metadata about this blog ;)
    'layout': 'post',
    'title': 'Getting Started with Bandit',
    'date': '2019-07-17 10:00',
    'author': 'Mari',
})
parsed_yaml = from_yaml(yaml_str)
```

Overflow


Buffer overflow example
Buffer (8 bytes)
Overflow
U S E R N A M E 1 2
0 1 2 3 4 5 6 7 8 9

CVE-2018-1000117

- Entre as versões 3.2 e 3.6.4 Python
- os.symlink()
- Ataque: Python Script onde o atacante gerencia o local de criação do link

XSS

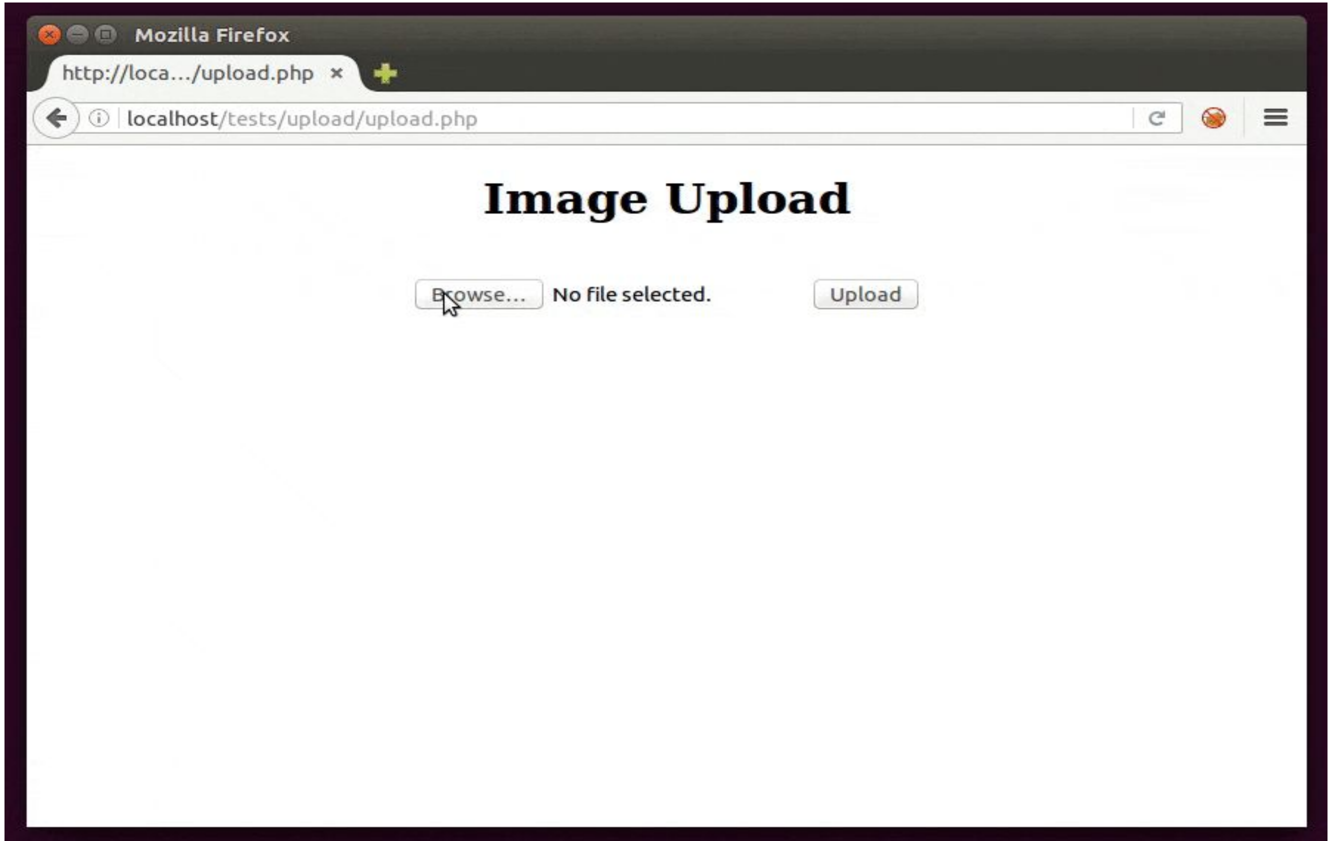
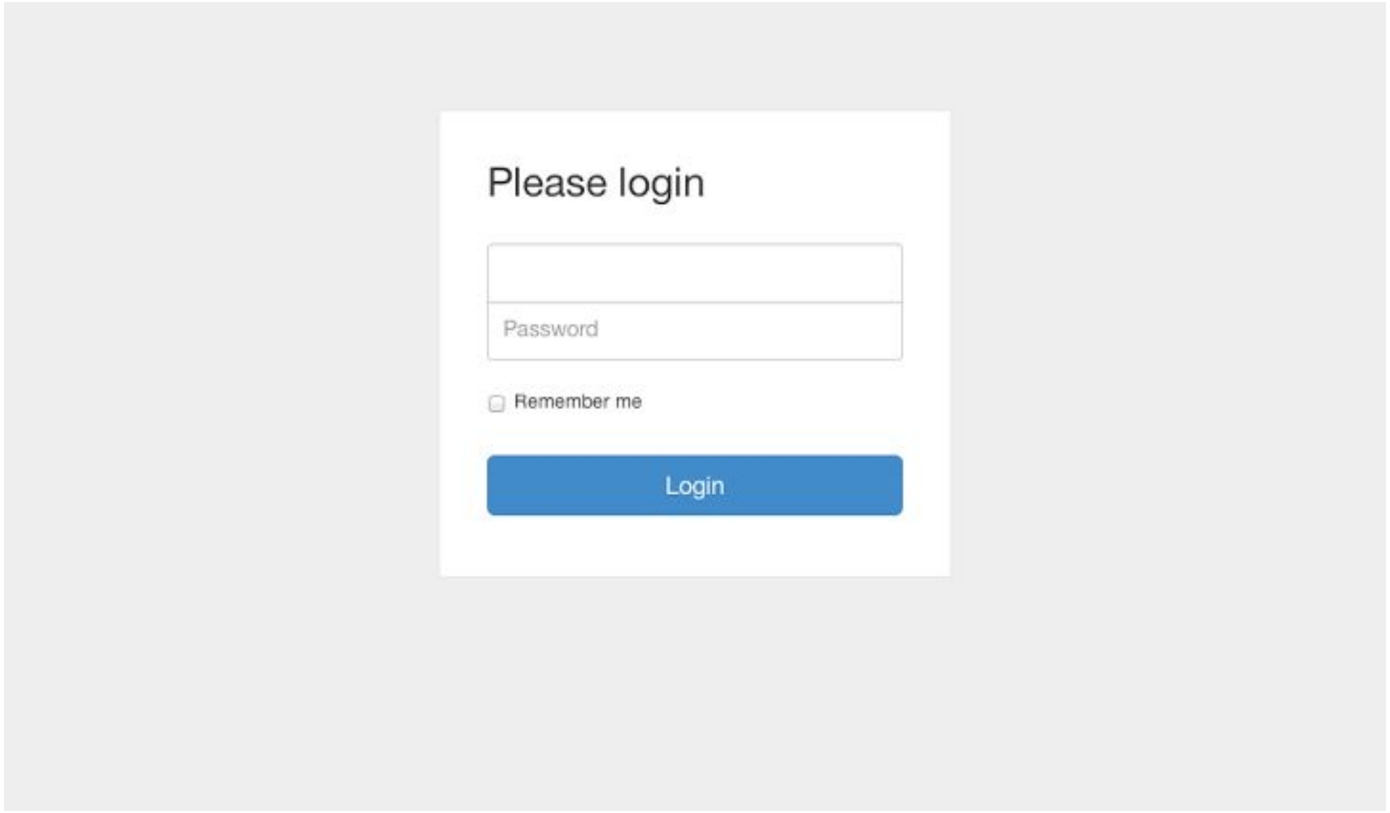


Image Upload

CVE-2019-12308

- django.contrib.admin.widgets
- AdminURLFieldWidget
- Versões 1.11 e 2.1
- Ataque: O usuário mal intencionado pode passar um valor diferente do esperado, gerando um link javascript clicável

Timing Attacks



P A S S W O R D

2,821,109,907,456 combinações

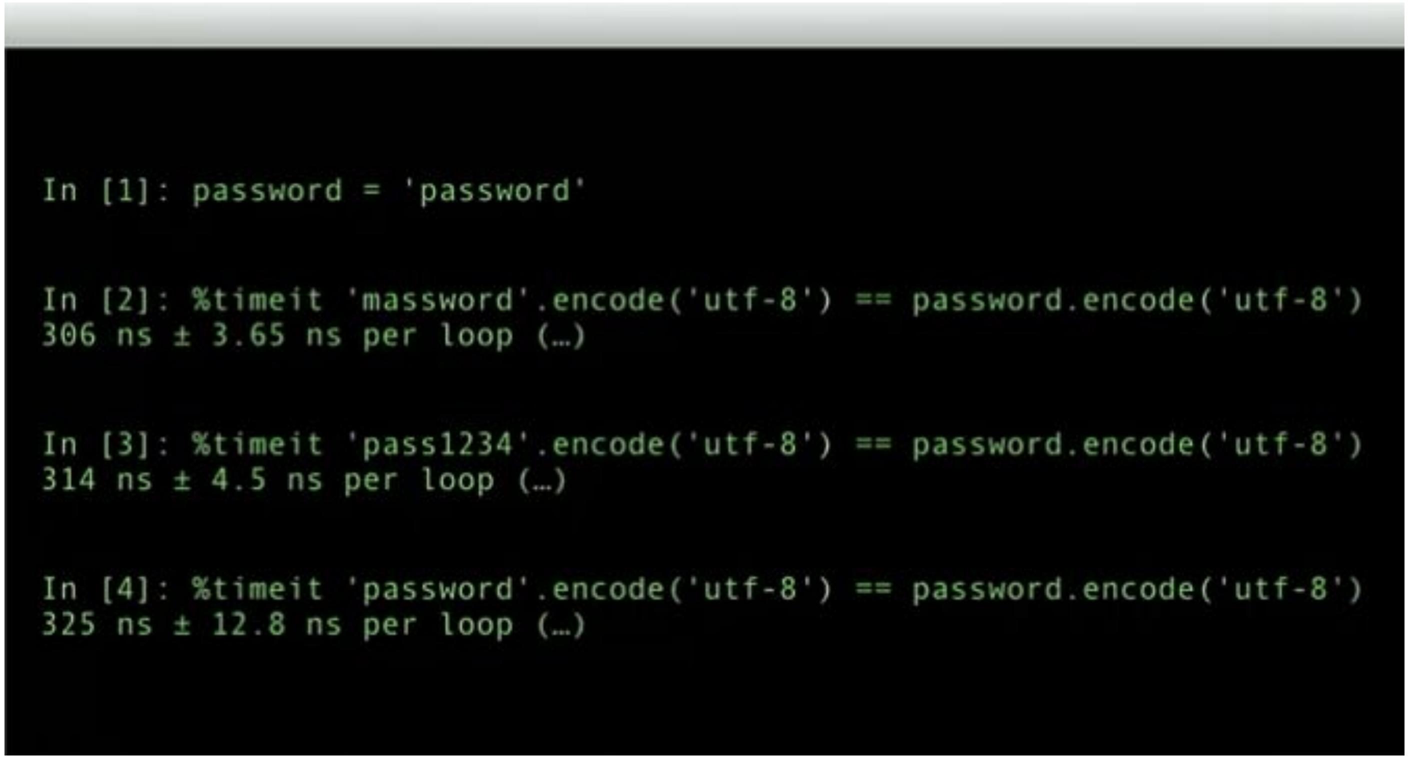
= ~ 89 anos

P A S S W O R D

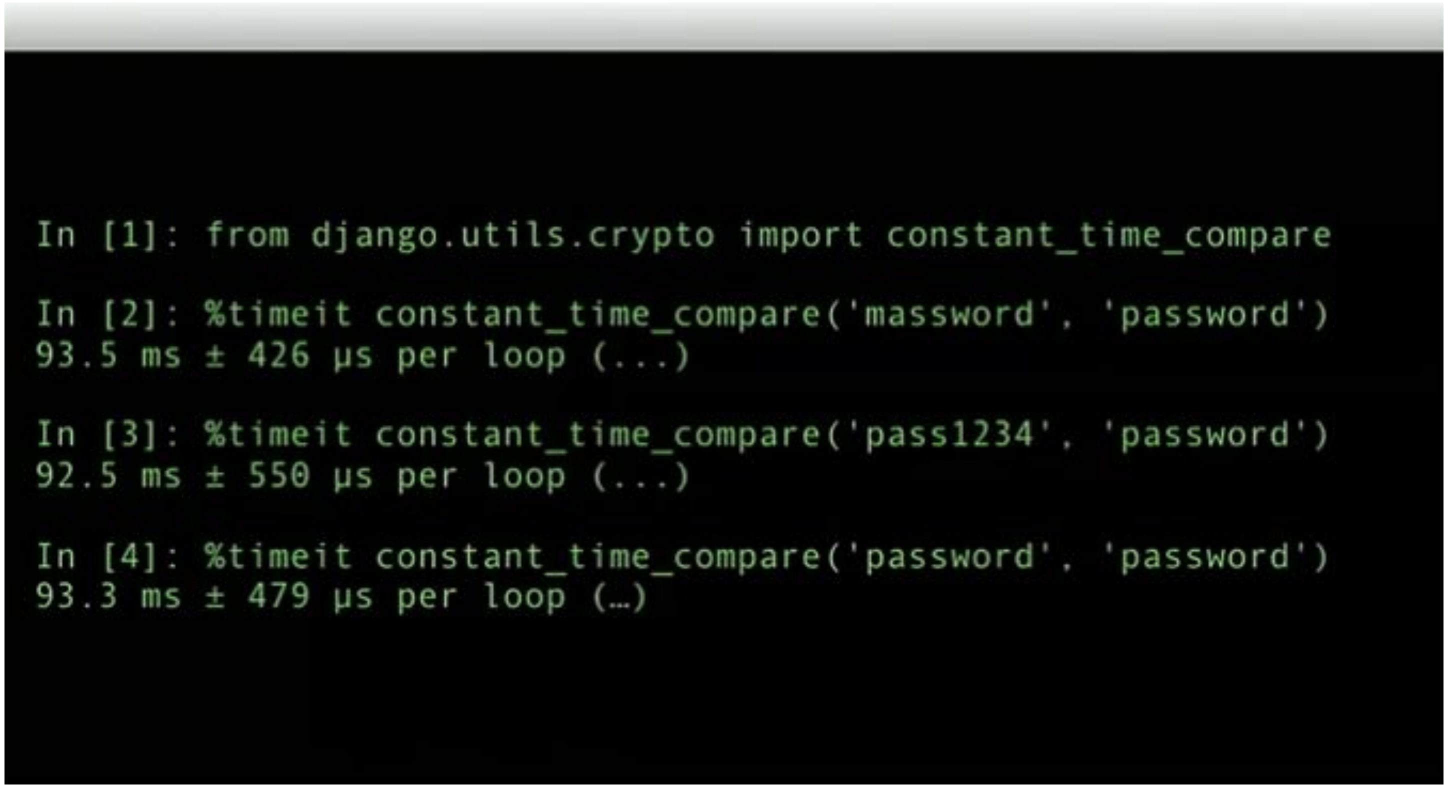
$\boxed{\cancel{\text{M}}}$ A S S W O R D

P A S S $\boxed{\cancel{1}}$ 2 3 4

P A S S W O R D ✓



```
In [1]: password = 'password'


In [2]: %timeit 'massword'.encode('utf-8') == password.encode('utf-8')
306 ns ± 3.65 ns per loop (…)


In [3]: %timeit 'pass1234'.encode('utf-8') == password.encode('utf-8')
314 ns ± 4.5 ns per loop (…)


In [4]: %timeit 'password'.encode('utf-8') == password.encode('utf-8')
325 ns ± 12.8 ns per loop (…)
```



```
In [1]: from django.utils.crypto import constant_time_compare

In [2]: %timeit constant_time_compare('massword', 'password')
93.5 ms ± 426 µs per loop (...)

In [3]: %timeit constant_time_compare('pass1234', 'password')
92.5 ms ± 550 µs per loop (...)

In [4]: %timeit constant_time_compare('password', 'password')
93.3 ms ± 479 µs per loop (…)
```

Python Hash Seed

```
/search?
q=bananas&
page=3&
country=br
```

```
request.GET = {
        ‘q’:’bananas’,
        ‘page’:3,
        ‘country’:’br’
}
```

Lists vs. Dicts

2s -> 6506s

```
hash(data)
hash(rand, data)
```

Antes de tudo: Pesquise

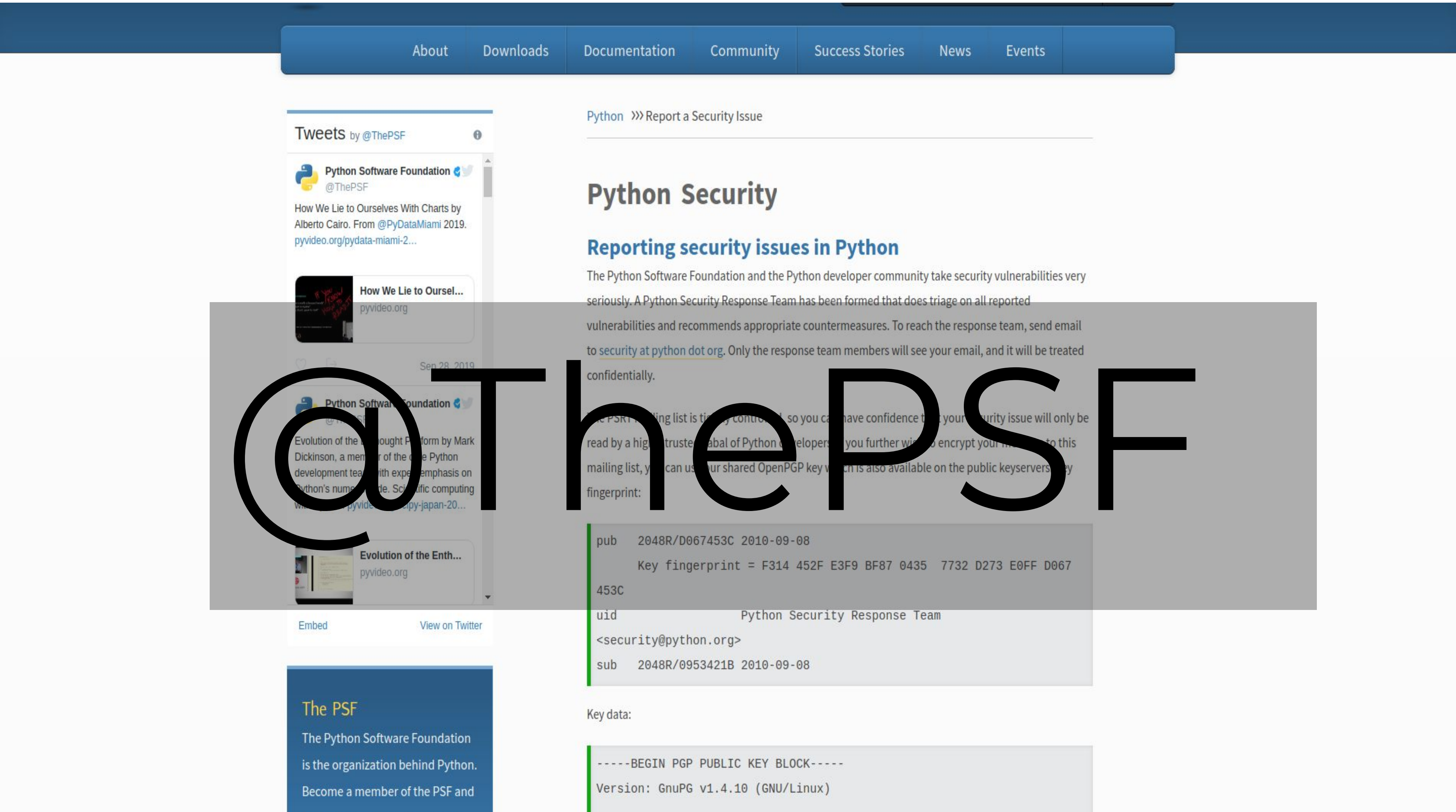
Antes de usar uma lib, verifique a última data de atualização

Faça atualizações

Manter as versões de frameworks e libs sempre atualizadas

Acompanhe o Python Security

Lista de vulnerabilidades e versões, assim como correções



Tweets by @ThePSF

Python Software Foundation @ThePSF

How We Lie to Ourselves With Charts by Alberto Cairo. From @PyDataMiami 2019. pyvideo.org/pydata-miami-2...

How We Lie to Oursel... pyvideo.org

Sep 28, 2019

Python Softwa[illegible]oundation

Evolution of the [illegible]ought P[illegible]form by Mark Dickinson, a mem[illegible]r of the c[illegible]e Python development tea[illegible]ith expe[illegible] emphasis on [illegible]ython's nume[illegible]de. Sci[illegible]fic computing w[illegible] pyvide[illegible]ipy-japan-20...

Evolution of the Enth... pyvideo.org

Embed View on Twitter

The PSF

The Python Software Foundation is the organization behind Python. Become a member of the PSF and

@ThePSF

Python >>> Report a Security Issue

Python Security

Reporting security issues in Python

The Python Software Foundation and the Python developer community take security vulnerabilities very seriously. A Python Security Response Team has been formed that does triage on all reported vulnerabilities and recommends appropriate countermeasures. To reach the response team, send email to security at python dot org. Only the response team members will see your email, and it will be treated confidentially.

[illegible]e PSRT [illegible]ing list is ti[illegible] contro[illegible] so you ca[illegible] have confidence t[illegible] your [illegible]urity issue will only be read by a hig[illegible] truste[illegible]abal of Python d[illegible]elopers [illegible] you further wi[illegible]o encrypt your [illegible] to this mailing list, y[illegible] can us[illegible]ur shared OpenPGP key w[illegible]ch is also available on the public keyservers [illegible]ey fingerprint:

```
pub   2048R/D067453C 2010-09-08
      Key fingerprint = F314 452F E3F9 BF87 0435  7732 D273 E0FF D067
453C
uid                   Python Security Response Team
<security@python.org>
sub   2048R/0953421B 2010-09-08
```

Key data:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.10 (GNU/Linux)
```

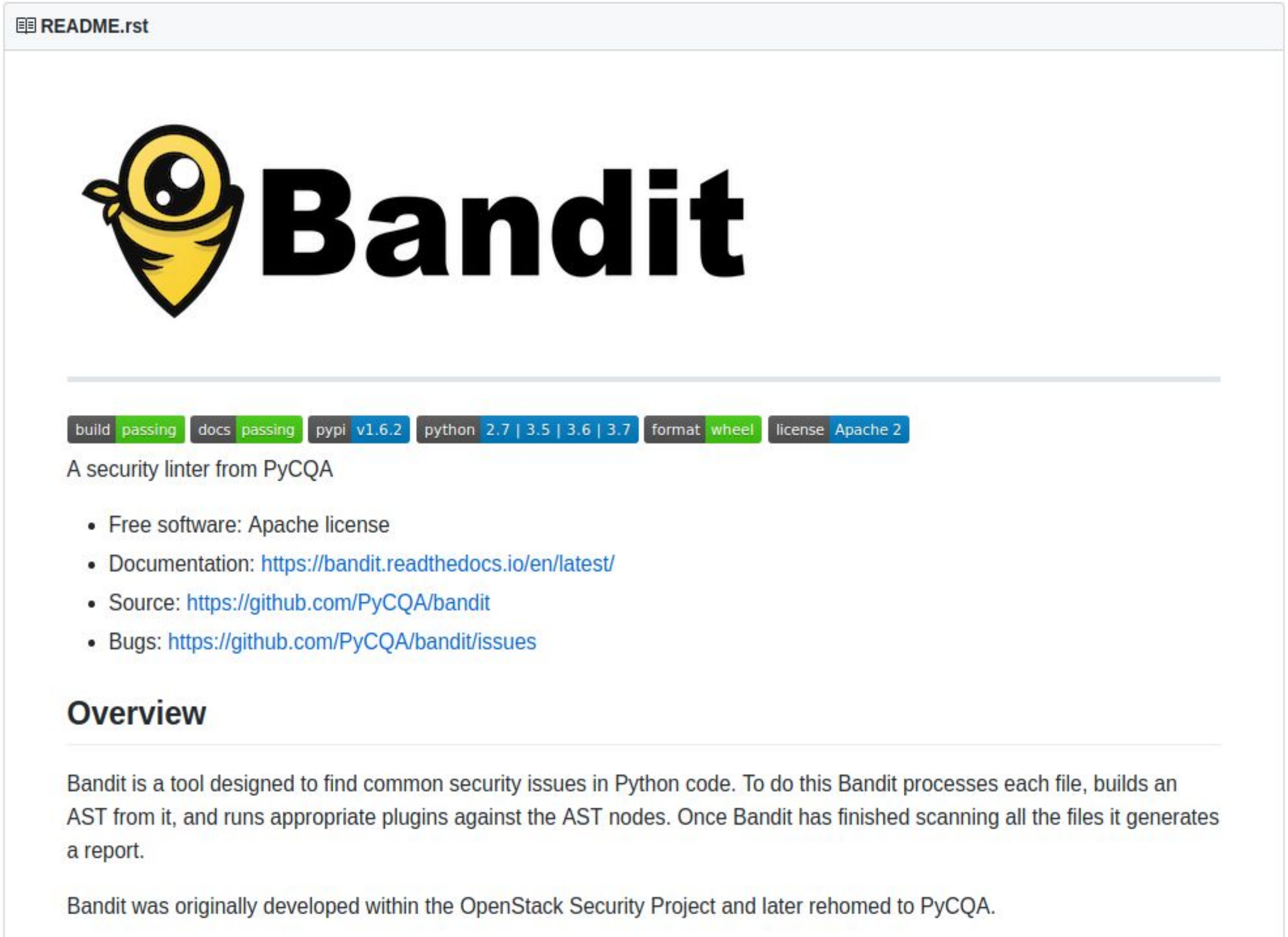
Configuração de DEBUG

Manter flag DEBUG de acordo com o ambiente

Atenção ao ambiente

Verificar logs e paths de configuração

Ferramentas



README.rst

Bandit

build passing | docs passing | pypi v1.6.2 | python 2.7 | 3.5 | 3.6 | 3.7 | format wheel | license Apache 2

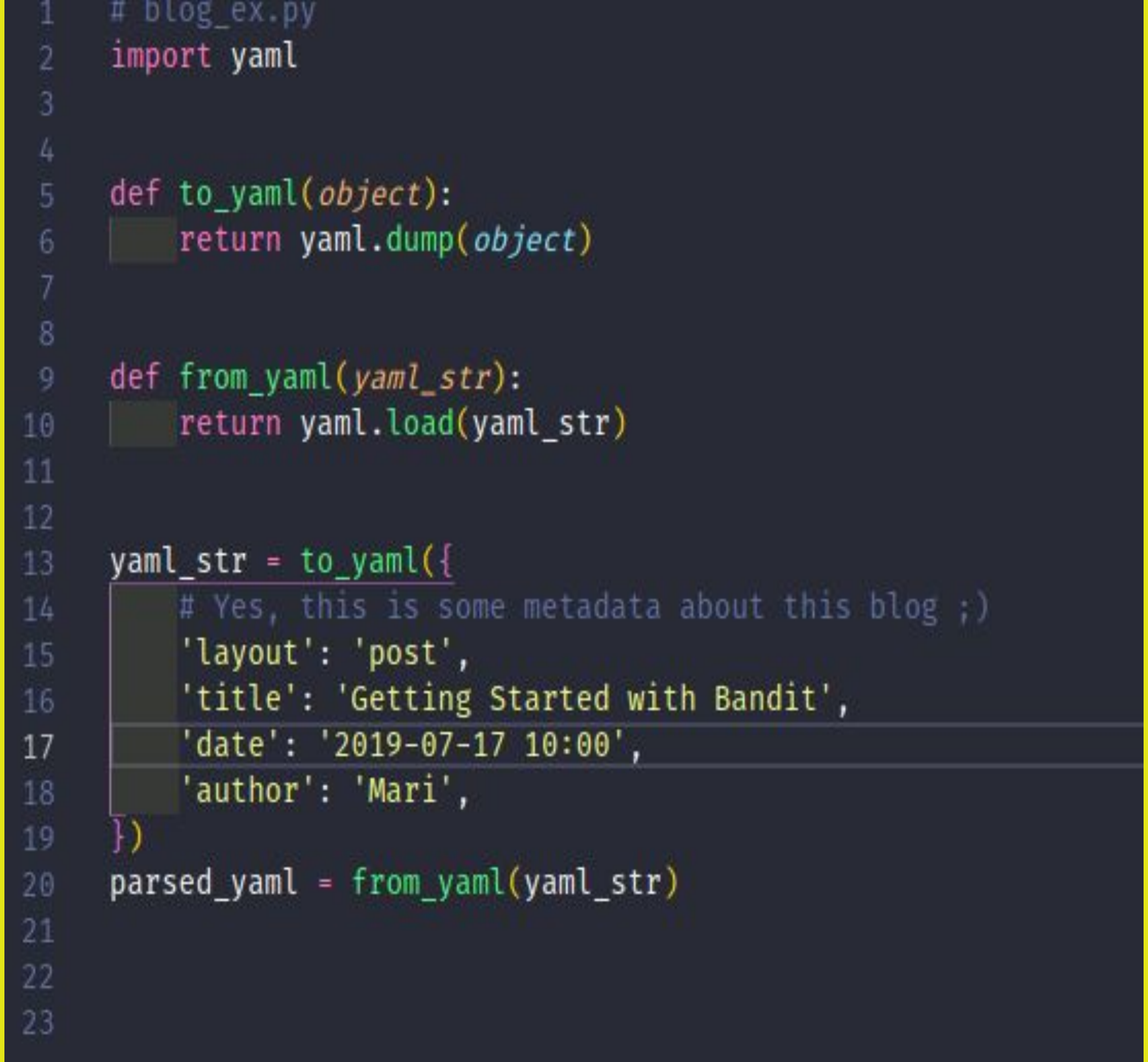
A security linter from PyCQA

- Free software: Apache license
- Documentation: https://bandit.readthedocs.io/en/latest/
- Source: https://github.com/PyCQA/bandit
- Bugs: https://github.com/PyCQA/bandit/issues

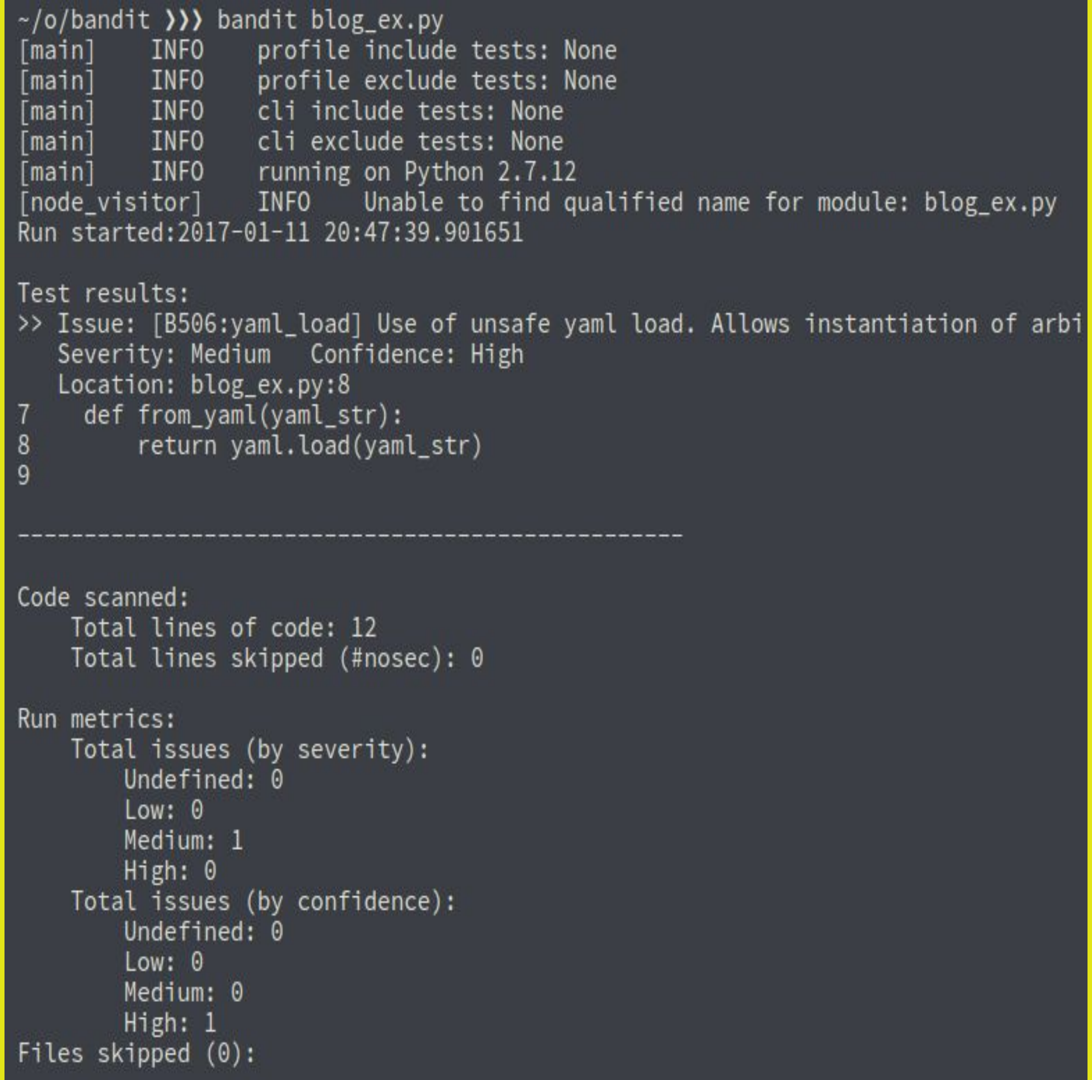
Overview

Bandit is a tool designed to find common security issues in Python code. To do this Bandit processes each file, builds an AST from it, and runs appropriate plugins against the AST nodes. Once Bandit has finished scanning all the files it generates a report.

Bandit was originally developed within the OpenStack Security Project and later rehomed to PyCQA.

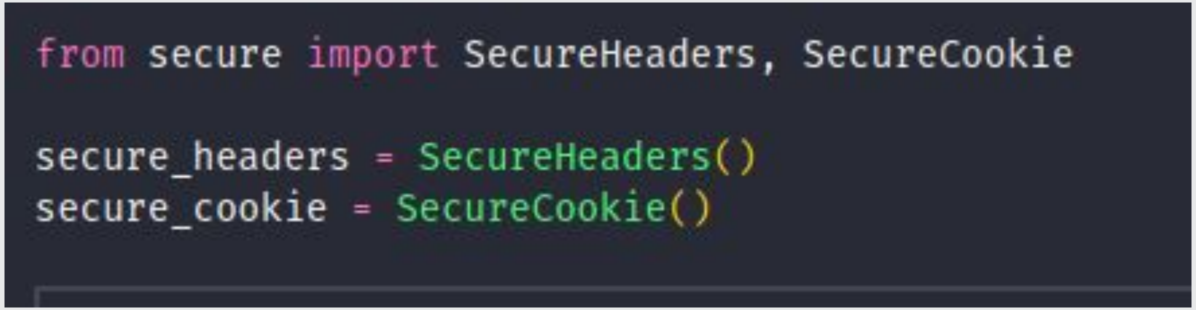


```python
# blog_ex.py
import yaml


def to_yaml(object):
    return yaml.dump(object)


def from_yaml(yaml_str):
    return yaml.load(yaml_str)


yaml_str = to_yaml({
    # Yes, this is some metadata about this blog ;)
    'layout': 'post',
    'title': 'Getting Started with Bandit',
    'date': '2019-07-17 10:00',
    'author': 'Mari',
})
parsed_yaml = from_yaml(yaml_str)
```

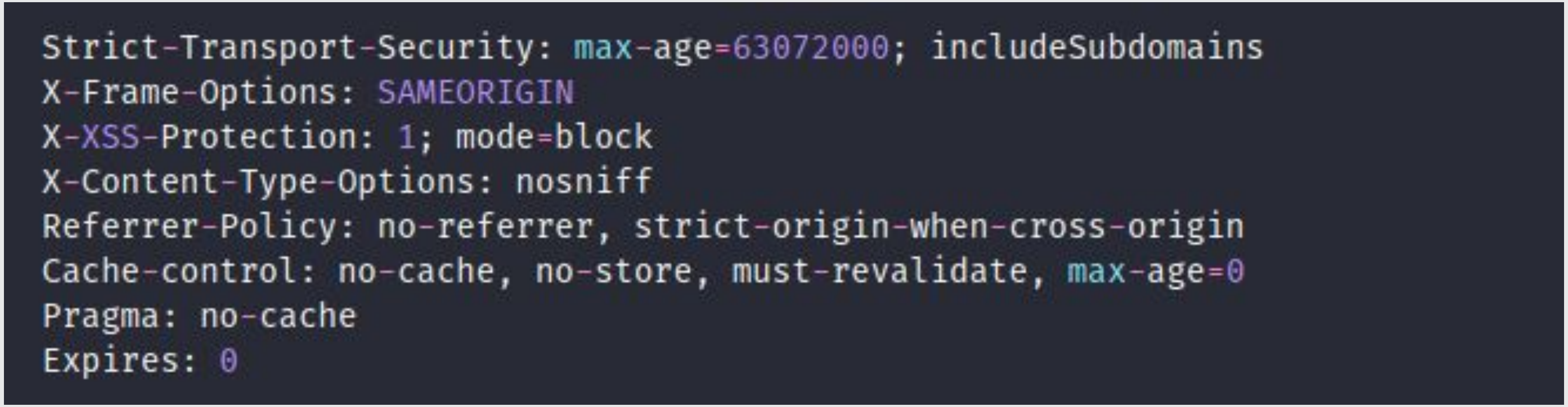


```
~/o/bandit ❯❯❯ bandit blog_ex.py
[main]    INFO    profile include tests: None
[main]    INFO    profile exclude tests: None
[main]    INFO    cli include tests: None
[main]    INFO    cli exclude tests: None
[main]    INFO    running on Python 2.7.12
[node_visitor]    INFO    Unable to find qualified name for module: blog_ex.py
Run started:2017-01-11 20:47:39.901651

Test results:
>> Issue: [B506:yaml_load] Use of unsafe yaml load. Allows instantiation of arbi
   Severity: Medium   Confidence: High
   Location: blog_ex.py:8
7    def from_yaml(yaml_str):
8        return yaml.load(yaml_str)
9

--------------------------------------------------


Code scanned:
    Total lines of code: 12
    Total lines skipped (#nosec): 0

Run metrics:
    Total issues (by severity):
        Undefined: 0
        Low: 0
        Medium: 1
        High: 0
    Total issues (by confidence):
        Undefined: 0
        Low: 0
        Medium: 0
        High: 1
Files skipped (0):
```

Secure.py



```python
from secure import SecureHeaders, SecureCookie

secure_headers = SecureHeaders()
secure_cookie = SecureCookie()
```



```
Strict-Transport-Security: max-age=63072000; includeSubdomains
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Referrer-Policy: no-referrer, strict-origin-when-cross-origin
Cache-control: no-cache, no-store, must-revalidate, max-age=0
Pragma: no-cache
Expires: 0
```

Outras ferramentas

Flask-HTTPAuth

Talisman

django-session-csrf

- Não garante 100% de segurança
- Minimizar superfície de ataque
- Grandes desafios, mas com grandes recompensas

Como você evita falhas de segurança em aplicações web com Python?

valeu galera!

pra quem quiser trocar uma ideia depois...

Github

@maricampos

Twitter & LinkedIn

@maricamppos

Software Developer at Tempest Security Intelligence

TEMPEST
Protegendo negócios
no mundo digital.