

Unique ways to Hack into a Python Web Service



About

- Senior Solutions Engineer at we45
- Developer of Open-Source Project called Orchestron, ThreatPlaybook
- Part of multiple CTF

<https://github.com/we45/orchestron-community>

<https://github.com/we45/ThreatPlaybook>



HMM



Agenda

- Intro to Web-Services
- Common Vulnerabilities
- Unique Vulnerabilities
- Some of Remediation Techniques
- Demo !

What is Web-Service?

- It designed to support interoperable machine to machine over the internet
- It is not tied to any one operating system or programming language



Types of Web-Service?

- SOAP Web Services
- RESTful web services

Python Rest Frameworks

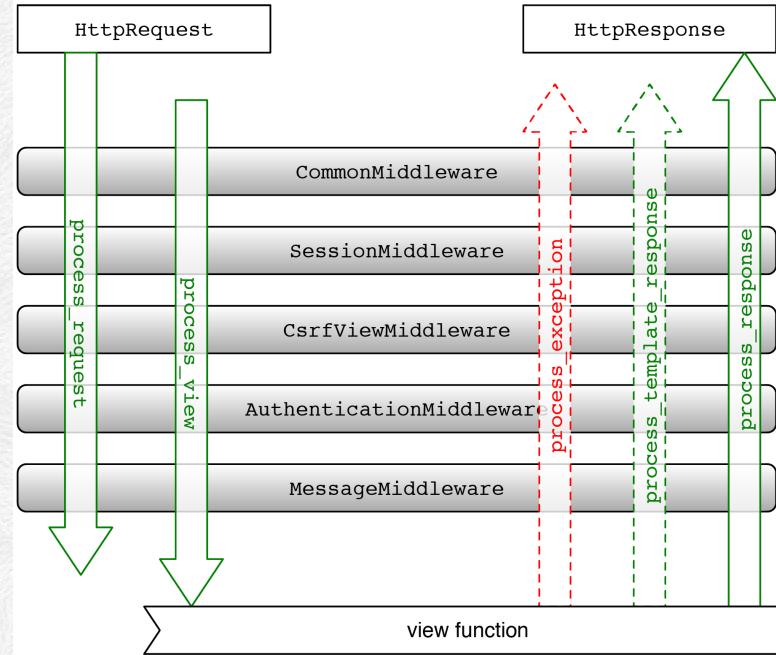
- DjangoRestFramework
- Flask
- Falcon
- Pyramid
- CherryPy
- Bottle

Common Security Threats

- SQL Injection
- Cross Site Scripting
- Broken Authentication
- Security Misconfiguration
- Cross-site Request Forgery
- Many More ..

Django prevents some of these attacks

- SQL Injection
- Cross-Site Request Forgery
- Cross-Site Scripting
- Session Hijacking



What about these

- JWT Manipulation
- XML External Entity
- InsecureDirectObjectReference
- Server-Side Template Injection
- Etc ...

JWT Manipulation

OWASP-2017 A5 Broken Access Control

Why JWT

- Stateless Application
- Authorization Mechanism
- Transfers information between server and client
- Scalable and decoupled



JSON Web Token(JWT)

- The process is relatively simple (typically):
 - Once a user authenticates, the server generates some JSON payload (with some info) and signs the JSON payload with a key
 - This can be a HMAC Based Key (HS256) or a Asymmetric System (RS256)
 - The token is sent by the client (like a session cookie)

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCIpXVCJ9.eyJzdW  
Ii0iIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaaG4gRG9lI  
iwiYWRtaW4iOnRydW9.TJVA950rM7E2cBab30RMhrH  
DcEfxyoYZgeFONFh7HgQ
```

Decoded EDIT THE PAYLOAD AND SECRET (ONLY HS256 SUPPORTED)

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

PAYOUT: DATA

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "admin": true  
}
```

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  secret  
) □ secret base64 encoded
```

⌚ Signature Verified

Lots of ways to get JWT wrong

- JWT allows for a “none” signature for a token
- Algo Confusion Attacks:
 - CVE-2017-11424
 - CVE-2015-9235
- JWT verification on non-unique private claims



Recent Attack



<https://thehackernews.com/2018/04/auth0-authentication-bypass.html>

month, Auth0 is one of the biggest identity platforms.

While pentesting an application back in September 2017, researchers from security firm Cinta Infinita discovered a flaw ([CVE-2018-6873](#)) in Auth0's **Legacy Lock API**, which resides due to improper validation of the **JSON Web Tokens (JWT)** audience parameter.

SHARE



Researchers successfully exploited this issue to bypass login authentication using a simple cross-site request forgery (CSRF/XSRF) attack against the applications running over Auth0 authentication.



Auth0's CSRF vulnerability ([CVE-2018-6874](#)) allows an attacker to reuse a valid signed JWT generated for a separate account to access the targeted victim's account.



For this, all an attacker needs is the victim's user ID or email address, which can be obtained using simple social engineering tricks.

DEMO GODS



**PLEASE LET
THESE DEMO WORKS**

imgflip.com

Mitigation

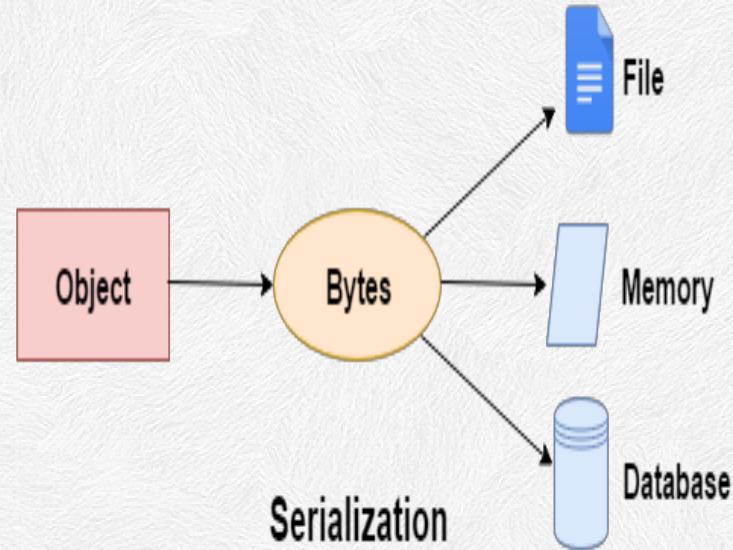
- Validate using Unique ID
- Ensure that JWT implementation doesn't support 'None' signature
- JWT lifetime relatively short
- Check library flaw

Insecure Deserialization

OWASP-2017 A8 Insecure Deserialization

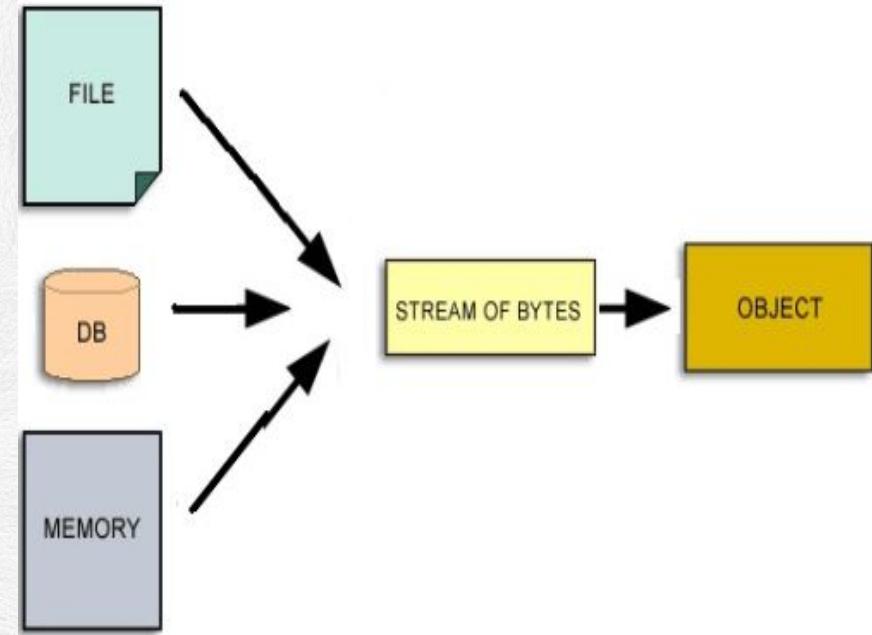
What is Serialization

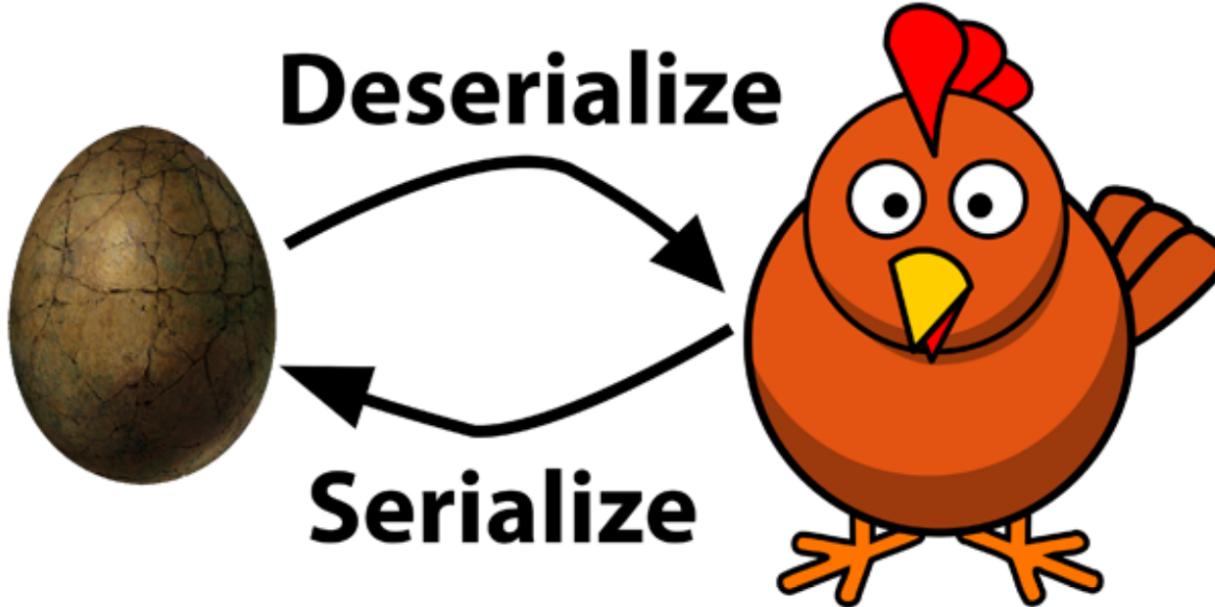
- Serialization means converting object into a binary stream.



What is Deserialization

- Deserialization means converting binary stream into an object.





Deserialization vulnerability

Security Gotcha !!!

- Malformed data deserialisation
- Abusive of an application logic
- Denial-of-Service
- Remote code execution

Recent Attack

SHARE
f
t
in
h
s

Exploiting PHP Deserialization Attack Against WordPress Sites

New TYPO3 site 8.7.16

Path: f3admin/ (auto-created) / user_upload 4 Files, 1.83 MB

File Name Type Last Modified Size RW Re

Temporary files (_temp_) Folder 28-06-18 1 File RW -

guzzle.jpg JPG 25-07-18 1.83 MB RW -

index.html HTML 28-06-18 0 B RW -

typo3.jpg JPG 28-06-18 352 B RW -

New TYPO3 site 8.7.16

Alignment Default

Date

Link

phar%3a//.../htdocs/f3admin/user_upload/guzzle.. Subheader

404

Popular News

New KickAss Torrent Best Torrent Sites of

The Pirate Bay Altern Best Torrent Sites | F Download

New Cold Boot Attac Disk Encryption On N Modern PCs

In a detailed paper released at Black Hat conference last week, Thomas demonstrated how this attack can be executed against Wordpress sites using an author account to take full control over the web

威胁

Cloud Security / Malware / Vulnerabilities / Privacy

Java Serialization Bug Crops Up At PayPal



The image shows a black tablet and a silver smartphone lying on a light-colored wooden surface. Both devices have the "PayPal™" logo displayed on their screens. Behind them, several credit cards are fanned out, including Visa, Mastercard, and American Express cards.

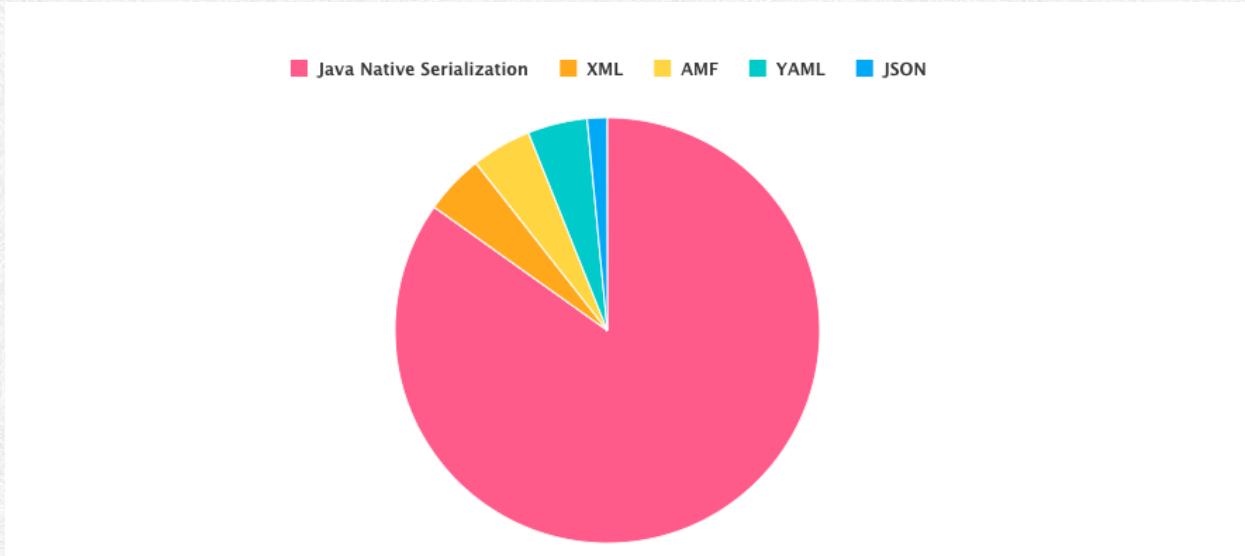
HEY HEY !!



DJANGO IS SECURE

imgflip.com

Vulnerabilities Serialization formats





Tavern

Automated RESTful API testing

[Documentation](#)[Plugins](#)[Examples](#)[View on GitHub](#)[Changelog](#)[Chat on Gitter](#)

Easier API testing

[build](#) [passing](#) [pypi](#) [v0.18.3](#) [gitter](#) [join chat](#)

Tavern is a pytest plugin, command-line tool and Python library for automated testing of APIs, with a simple, concise and flexible YAML-based syntax. It's very simple to get started, and highly customisable for complex tests. Tavern supports testing RESTful APIs as well as MQTT based APIs.

The best way to use Tavern is with [pytest](#). Tavern comes with a pytest plugin so that literally all you have to do is install pytest and Tavern, write your tests in `.tavern.yaml` files and run pytest. This means you get access to all of the [pytest ecosystem](#) and allows you to do all sorts of things like regularly run your tests against a test



Features

Business

Explore

Marketplace

Pricing

yaml.load

Sign in or Sign up

Code 14

Commits 1

Issues

Wikis

Languages 13

Python

Markdown 1

[Advanced search](#) [Cheat sheet](#)

14 code results in [taverntesting/tavern](#) or view all results on GitHub

Sort: Best match ▾

[tavern/schemas/files.py](#)

Python

Showing the top six matches Last indexed on Aug 7

```
16     from tavern.util.loader import IncludeLoader
17     core.yaml.safe_load = functools.partial(yaml.load, Loader=IncludeLoader)
18
19     logger = logging.getLogger(__name__)
...
32         with open(schema_filename, "r") as sfile:
33             self._loaded[schema_filename] = yaml.load(sfile.read())
```

[tavern/util/general.py](#)

Python

Showing the top three matches Last indexed on Jun 27

```
1     import logging
2     import yaml
3     from .dict_util import deep_dict_merge
4
5
6     logger = logging.getLogger(__name__)
...
25        for filename in global_cfg_paths:
26            with open(filename, "r") as gfileobj:
27                contents = yaml.load(gfileobj)
```



imgflip.com

DEMO

Mitigations

- Integrate integrity check such as digital signature
- Isolate then deserialise the data
- Monitor incoming and outgoing network connectivity
- Instead of ‘yaml.load’ use yaml.safe_load

Insecure Direct Object Reference

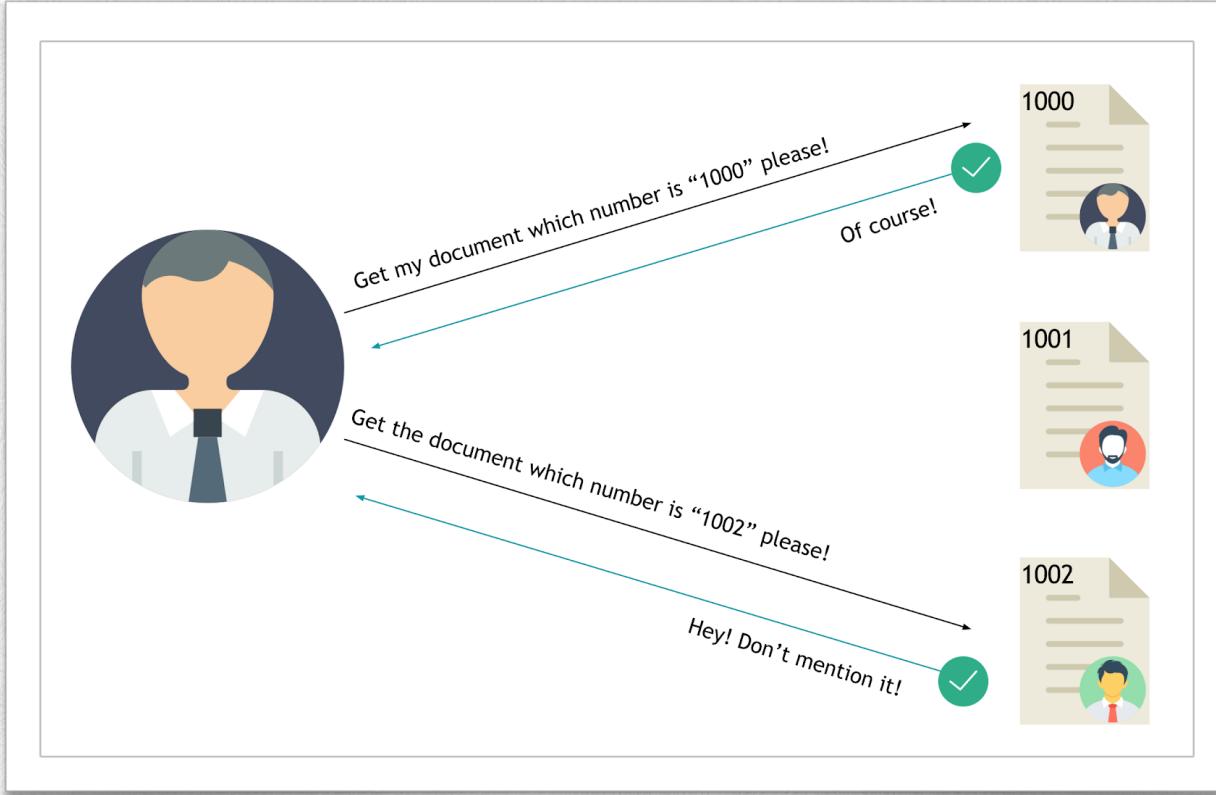
OWASP-2017 A8 Broken Access Control

Insecure Direct Object reference

- “id”, “pid”, “uid” are often seen in HTTP parameter
- Accessing other user privilege
- Backend not properly validated users

```
re_path(r'api/get_user/(?P<pk>[0-9]+)$', IndividualUserView.as_view({'get': 'list', 'post': 'update'}))
```

How it works



Bad Code

```
class IndividualUserView(viewsets.ModelViewSet):
    serializer_class = UserSerializer
    model_class = User

    def get_queryset(self, pk):
        object_list = self.model_class.objects.get(pk=pk)
        return object_list

    def list(self, request, pk=None):
        queryset = self.get_queryset(pk)
        serializer = self.serializer_class(queryset)
        return Response(serializer.data)

    def update(self, request, pk=None):
        queryset = self.get_queryset(pk)
        serializer = self.serializer_class(queryset, data=request.data)
        if serializer.is_valid(raise_exception=True):
            serializer.save()
            return Response(serializer.data, status=status.HTTP_200_OK)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

Good Code

```
class IndividualUserView(viewsets.ModelViewSet):
    serializer_class = UserSerializer
    model_class = User

    def get_queryset(self, pk):
        object_list = self.model_class.objects.get(pk=pk)
        return object_list

    def list(self, request):
        queryset = self.get_queryset(request.user.id)
        serializer = self.serializer_class(queryset)
        return Response(serializer.data)

    def update(self, request):
        queryset = self.get_queryset(request.user.id)
        serializer = self.serializer_class(queryset, data=request.data)
        if serializer.is_valid(raise_exception=True):
            serializer.save()
            return Response(serializer.data, status=status.HTTP_200_OK)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

Yahoo Breach

The screenshot shows a web browser window with the URL https://thehackernews.com/2014/03/yahoo-vulnerability-allows-hacker-to_1.html. The main content is a news article titled "Yahoo vulnerability allows Hacker to delete 1.5 million records from Database". Below the title, it says "March 01, 2014" and "Anonymous". The article discusses a security vulnerability found on the Yahoo Answers platform.

Screenshot of the Yahoo! Answers Suggestion Board:

- Header:** YAHOO! ANSWERS
- Title:** Yahoo! Answers Suggestion Board
- Text:** We are listening - we want to hear from you!
- Details:** Created 5 years ago by vtarrani. Category: Suggestions.
- Comments:** 1 vote
- Description:** Encryption on yahoo for secure login would not require a great many changes -- it could be implemented as a selection to test code before it became part of the entire procedure. Yahoo has links to a lot of places for purchases, and this would show how much you help get rid of phishers and spammers. Thanks.
- Buttons:** Have an idea? make a suggestion
- Help:** NEED HELP? What's Yahoo Answers? Community Guidelines Team Blog HelpFAQs About this Board User-moderated Groups Report Abuse
- Comments:** 0 Comments. No comments yet...
- Sign In:** Please sign in to add a comment.

Bottom of the page: Yahoo! The 4th most visited website on the Internet has been found vulnerable multiple times, and this

Right sidebar (Sponsored):

- Ad:** Secure and manage all the endpoints from a single console. ManageEngine Desktop Central. Free Trial.
- Text:** SPONSORED

Popular News:

- New KickAss Torrents Site — 6 Best Torrent Sites of 2018**
- The Pirate Bay Alternatives | Best Torrent Sites | Free Movie Download**

TIME FOR

DEMO



imgflip.com

Mitigation

- Validate user using requested query
- Check database is that user is genuine or not
- Custom validation in server side as well as client side
- JWT should be invalidated once the user is logout

Some Tips



To Prevent some of threats

- Run SCA,
 - <https://github.com/pyupio/safety>
- Run SAST
 - <https://github.com/PyCQA/bandit>
- Run DAST
 - https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
- Include security testing in DevOps pipeline



Basic Pipeline Demo

Download Examples and Slides

- You can download it from
 - [http://github.com/we45/
djangocon-2018](http://github.com/we45/djangocon-2018)

<https://github.com/we45/orchestron-community>

<https://github.com/we45/ThreatPlaybook>

Thank you

