Insecure Deserialization

/b1twis3

Overview

- Process of converting an object into a format that can be stored then restored later or transmitted over the network
- Deserialization refers to the opposite
- Why? To store, restore or transmit
- Nowadays JSON, before that, it was XML.

Overview

- Binary Formats: Java, Python (pickle) and C++
- Readable Formats: Yaml, XML, JSON, PHP and SOAP

PHP

1. serialize(): takes a PHP object and convert it to a specific string format [demo0.php]
2. unserialize(): takes specific string format and convert it into a PHP object [demo1.php]
3. phar:// -> PHAR is deserialized

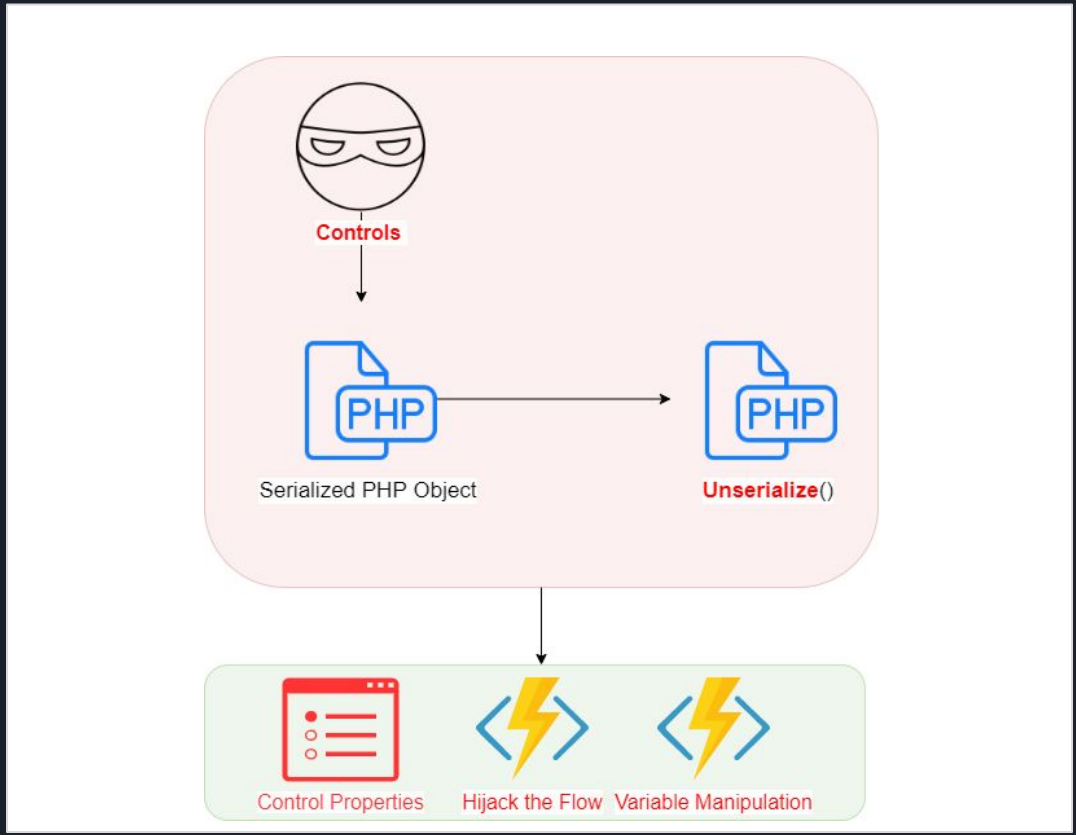
PHP (Insecure)

Magic Methods

The function names __construct(), __destruct(), __call(), __callStatic(), __get(), __set(), __isset(), __unset(), __sleep(), __wakeup(), __serialize(), __unserialize(), __toString(), __invoke(), __set_state(), __clone() and __debugInfo() are magical in PHP classes. You cannot have functions with these names in any of your classes unless you want the magic functionality associated with them.

unserialize(): calls __wakup() & __destruct()

PHP (Exploitation)



PHP (Variable Manipulation)

[demo2.php]

PHP (RCE)

1. demo3.php
2. demo4.php
3. https://hackerone.com/reports/415501

PHP (PHAR)

Global Phar manifest format

Size in bytes	Description
4 bytes	Length of manifest in bytes (1 MB limit)
4 bytes	Number of files in the Phar
2 bytes	API version of the Phar manifest (currently 1.0.0)
4 bytes	Global Phar bitmapped flags
4 bytes	Length of Phar alias
??	Phar alias (length based on previous)
4 bytes	Length of Phar metadata (*0* for none)
??	Serialized Phar Meta-data, stored in serialize() format
at least 24 * number of entries bytes	entries for each file

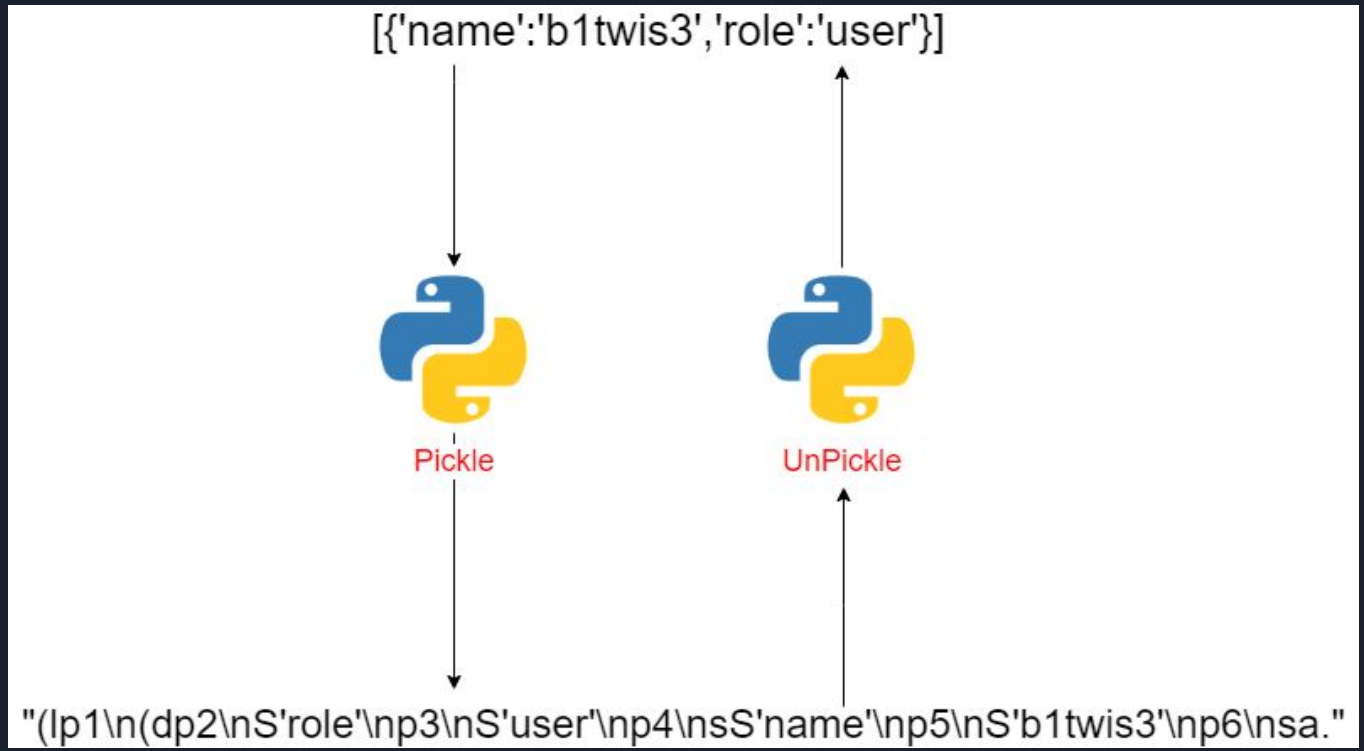
PHP (PHAR)

1. demo_phar1.php
2. Walking through CVE-2018-19274 write-up by Daniel Timofte

   https://www.ixiacom.com/company/blog/exploiting-php-phar-deserialization-vulnerabilities-part-2

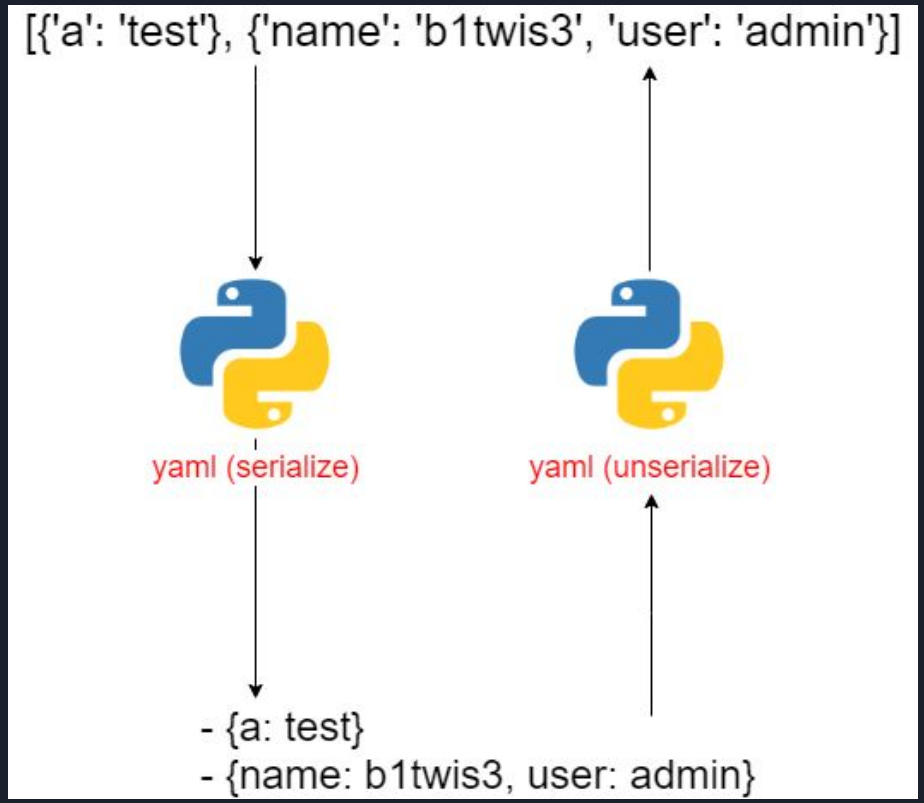
Note: PHP filesystem functions deserialize metadata through phar://.

Python



import pickle/c_pickle/_pickle/marshal [demo5.py]

Python



import yaml #PyYAML [demo6.py]

Python (__reduce__)

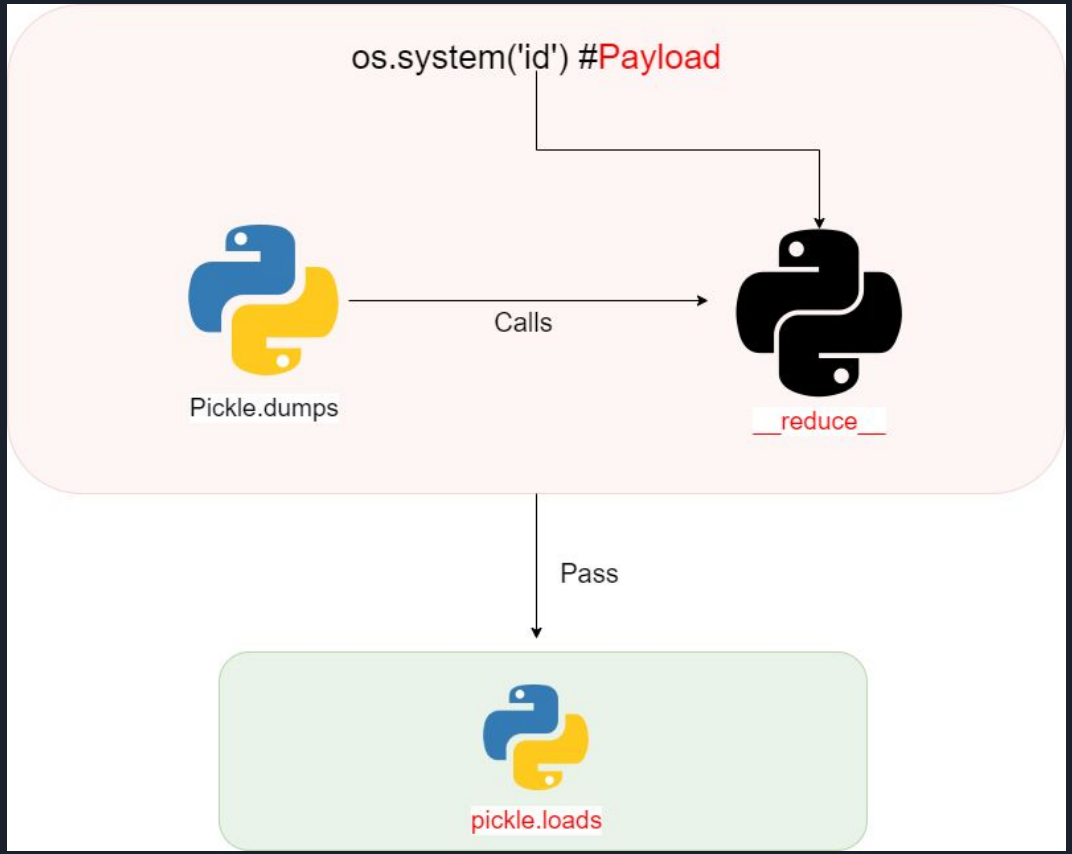
Intended for how to reconstruct (unpickle) objects

`object.__reduce__()`

The interface is currently defined as follows. The `__reduce__()` method takes no argument and shall return either a string or preferably a tuple (the returned object is often referred to as the “reduce value”).

If a string is returned, the string should be interpreted as the name of a global variable. It should be the object’s local name relative to its module; the pickle module searches the module namespace to determine the object’s module. This behaviour is typically useful for singletons.

Python (Exploitation)



-demo7.py

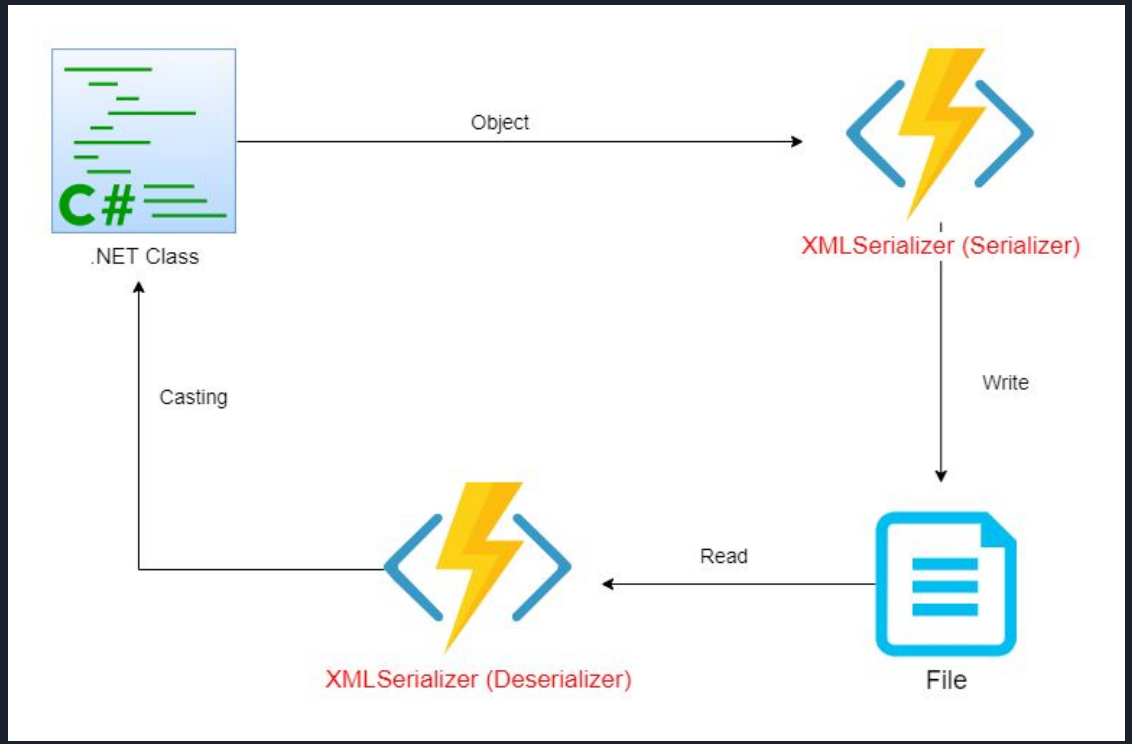
- app

.NET

1. .NET XML Serializer
2. .NET Formatters
3. .NET JSON Serializer

.NET

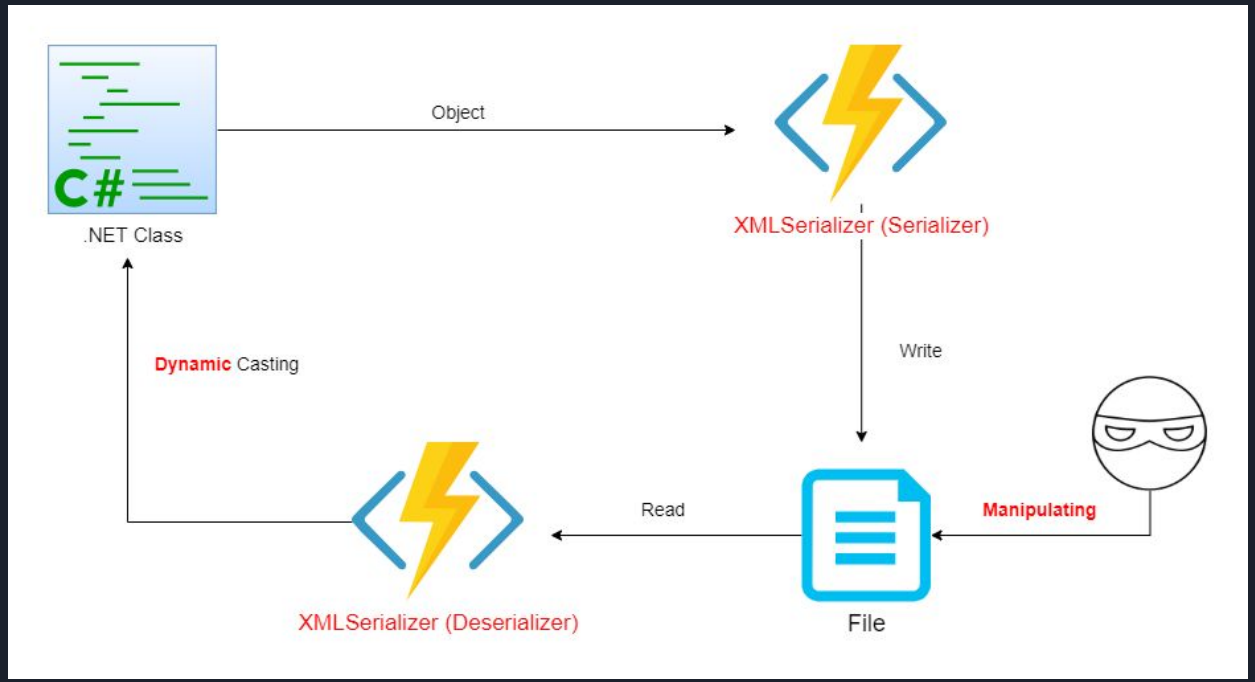
.NET XML Serializer



- Project1.cs
- Project2.cs

.NET

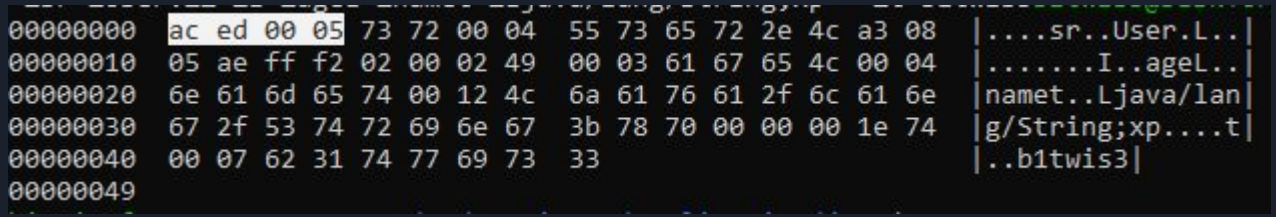
.NET XML Serializer



-Project3.cs
-Project4.cs

Java

1. java.io.Serializable Interface [writeObject() and readObject()]
2. Starts with base64(r00) or ACED0005
3. Content-type: application/x-java-serialized-object



JavaDemo1.java & JavaDemo2.java

Java

How it goes wrong?

```
bitwis3@DESKTOP-F1MQQJF:~/CodeReviewCS/serlization/java$ java JavaDemo3 hamid
::Session genererated::
bitwis3@DESKTOP-F1MQQJF:~/CodeReviewCS/serlization/java$ java JavaDemo4 session.txt
name = hamid
role = user
[+] Login Successful: hamid
bitwis3@DESKTOP-F1MQQJF:~/CodeReviewCS/serlization/java$ java JavaDemo3 b1twis3
::Session genererated::
bitwis3@DESKTOP-F1MQQJF:~/CodeReviewCS/serlization/java$ java JavaDemo4 session.txt
name = b1twis3
role = admin
[+] [Admin] Login Successful: b1twis3
bitwis3@DESKTOP-F1MQQJF:~/CodeReviewCS/serlization/java$
```

JavaDemo3.java & JavaDemo4.java