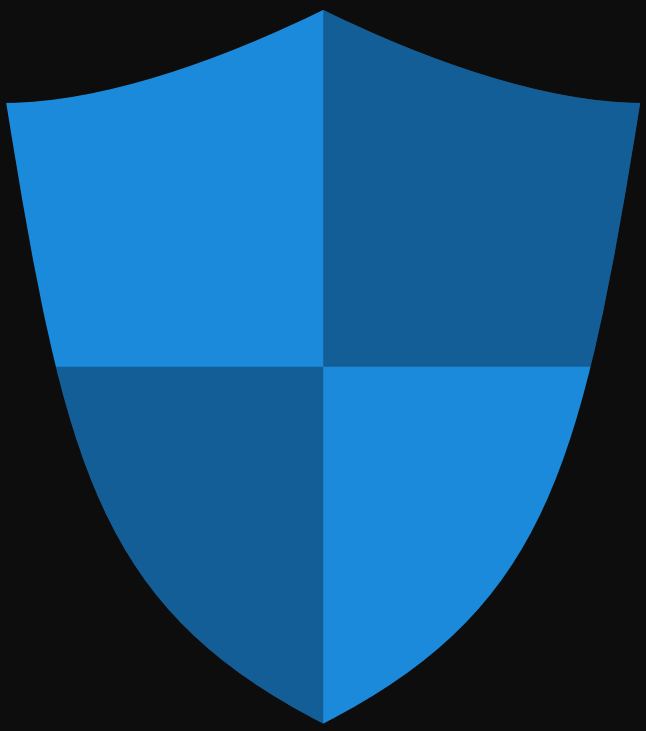
Insecure Deserialization Explained

Understanding the Security Risk



What is Deserialization?

Converting data from a serialized format back into an object that an application can use.

Deserialization in Web Applications

Deserialization in web applications refers to the process of converting a data stream, often received from a client or another external source, back into its original form as an object that the application can use.

In web applications, data serialization formats like JSON, XML, and Protocol Buffers are commonly used to transfer data efficiently and reliably.

The Risk of Insecure Deserialization

Insecure deserialization is a security risk where a web application deserializes data from an untrusted source without proper validation or sanitization.

This can allow an attacker to manipulate serialized objects and potentially execute arbitrary code, gain unauthorized access to sensitive information, or cause a denial of service.

Javascript insecure example

```
const serializedData = '{"name":"user","role":"admin"}';
const parsedData = JSON.parse(serializedData);
console.log(parsedData);
```

Javascript secure example

```
const serializedData = '{"name":"user","role":"admin"}';
// Validate the data using a library like joi
const joi = require('@hapi/joi');
const schema = joi.object({
  name: joi.string(),
  role: joi.string(),
});
const { error, value } = schema.validate(serializedData);
if (error) {
  throw new Error('Invalid data');
}
// Now you can safely deserialize the data
const parsedData = JSON.parse(value);
console.log(parsedData);
```

Python insecure example

```python
import pickle

serialized_data =
b'\x80\x03c__main__\nUser\nq\x00)\x81q\x01}q\x02X\x05\x00\x00\x00nameq\x03X\x03\x00\x00\x00ageq\x04
deserialized_data = pickle.loads(serialized_data)
print(deserialized_data)
```

Python secure example

```python
import pickle

# Allowlist of classes that can be deserialized
allowed_classes = [User]

serialized_data =
b'\x80\x03c__main__\nUser\nq\x00)\x81q\x01}q\x02X\x05\x00\x00\x00nameq\x03X\x03\x00\x00\x00ageq\x04K
def safe_deserialize(serialized_data):
    try:
        unpickled = pickle.Unpickler(serialized_data)
        class_ref = unpickled.find_class(allowed_classes, None)
        unpickled.find_class = class_ref
        return unpickled.load()
    except pickle.UnpicklingError:
        raise Exception("Invalid data")

deserialized_data = safe_deserialize(serialized_data)
print(deserialized_data)
```

PHP insecure example

```php
$serializedData = 'O:8:"stdClass":2:{s:5:"login";s:5:"admin";s:8:"password";s:5:"12345";}';
$deserializedData = unserialize($serializedData);
var_dump($deserializedData);
```

PHP secure example

```php
$serializedData = 'O:8:"stdClass":2:{s:5:"login";s:5:"admin";s:8:"password";s:5:"12345";}';
$pattern = '/^(O\:\d+\:".+"\:\d+\:')\{(.*)\}/';
if (preg_match($pattern, $serializedData)) {
    $deserializedData = unserialize($serializedData);
    var_dump($deserializedData);
} else {
    throw new Exception('Invalid data');
}
```

Learn More:

A08:2021-Software and Data Integrity Failures

https://owasp.org