NOTE: SCENARIO IS WHAT THE PUZZLE TAKER SEES.

SCENARIO

You are working on a RESTful web service that analyzes Python code against a SQL injection vulnerability. The service receives an HTTP request, in which the body is a YAML document containing a Python code snippet, analyzes it, and returns an HTTP response containing the analysis result. Here is an example of the body (a YAML document) of a valid HTTP request:

```
id: 0
code: "print('Hello, world!')"
```

And the below is an example of the body (a YAML document containing the analysis result) of a valid HTTP response:

```
id : 0
result: “No vulnerabilities were found.”
```

YAML, which is a data serialization format similar to JSON and XML. Python developers commonly use the PyYAML library for YAML serialization and deserialization. PyYAML is not included in the Python standard library. In the snippet of code below, the function `yaml.load` converts the YAML document (serialized form of data) coming from the client HTTP request into a Python object. Consider the snippet of code below and answer the following questions, assuming that the code has all required permissions to execute.

```
# Analysis Request Template (YAML document)
id: 0 # Request ID, to be sent back with the result
code: "Any python code to be analyzed"
```

```
# Analysis Response Template (YAML document)
id: 0 # Response ID, same as request ID
result: "The result of vulnerability analysis of the code"
```

```
# SQL injections vulnerability analyzer (Python code)
import yaml # and whatever else is needed

# This function accepts an analysis request in YAML format,
# deserializes it into a Python object, analyzes the Python code
# (sent by user to be analyzed), prepares the analysis result #
(string), and returns it.
def process(request):
  document = yaml.load(request)
  id = document['id']
  print('Analyzing the request with ID:', id)
  result = _analyze(id, document['code'])
```

```
13     result['id'] = id
14     return yaml.dump(result)
15
16   # The following function analyzes Python code for SQL injection
17   # vulnerability. It accepts Python code (as string), analysis
18   # the code and returns the analysis result (a Python object).
19   def _analyze(code):
20     # The implementation is omitted. You may assume that the
21     # implementation correctly does what is described above.
22     pass
```

Questions:

1. What will the program do when executed?

2. Which of the following could be true, if the user is allowed to send to the service any data.

a. The user will receive only the request ID and a string corresponding to the SQL injection vulnerability analysis result of the submitted code.

b. Line #11 prints the ID of the request made by user.

c. Both a and b.

d. Neither a nor b.

[Other statistical questions will be imported here while creating the survey.}

NOTE: ANSWER IS TO BE SHOWN TO THE PUZZLE TAKER AT THE END OF THE SESSION.

ANSWERS

1. The program deserializes the request made by the user to a Python object, analyzes the Python code (sent as request body in YAML format) against the SQL injection vulnerability and responds to the user with the analysis result.

2. d

While deserializing YAML documents using the `yaml.load` function, the function may call any Python function. For example if the YAML document sent by the user contained:

```
id: !!python/object/apply:subprocess.check_output ['ls']
code: "print('Hello, world!')"
```

The user would receive the list of files and directories in the current directory on the server. The safe way to deserializes a YAML document from an untrusted source is to use the `yaml.safe_load` function which limits the deserialization process to only simple Python objects like integers or lists.

TAGS:

Python, YAML, remote-code-execution

CATEGORIES:
Blindspot - YES
Type - Injection
Number of distinct functions - 4
Number of total functions - 4
Blindspot function - `yaml.load`
Function call omitted - YES
Blindspot type - dangerous method
Number of parameters in the blindspot function - 1
Cyclomatic complexity - 1

NAME:
yaml.load(document)

DESCRIPTION:
The function yaml.load converts a YAML document to a Python object. yaml.load accepts a byte string, a Unicode string, an open binary file object, or an open text file object. A byte string or a file must be encoded with utf-8, utf-16-be or utf-16-le encoding. yaml.load detects the encoding by checking the BOM (byte order mark) sequence at the beginning of the string/file. If no BOM is present, utf-8 is assumed. yaml.load returns a Python object.

BLINDSPOT:
It is not safe to call yaml.load with any data received from an untrusted source! yaml.load is as powerful as pickle.load and so may call any Python function. It enables users to craft input that executes arbitrary code on your machine i.e code injection.

YAML.load(params[:file]) for example

CORRECT USE EXAMPLE:
The correct way to deserialize a YAML document is to use the `yaml.safe_load` function.

MORE INFORMATION:
#N/A

REFERENCES:

1. http://kevinlondon.com/2015/08/15/dangerous-python-functions-pt2.html