

Pysa Experiment (Task – 1)

Refer to the below step-by-step guide if you are not able to run Pysa or Dynamic debugger.

Sub Task – 1

For the Sub Task-1, you must go through all the files in the task-1 directory and answer the questions. Below are all the files.

Views.py

```
views.py x
Task-1 > views.py > ...
1  import requests
2
3  def solve(value):
4      result = eval(f"100 {value} 5")
5      return result
6
7  base_url = 'https://pysa-api-2598d-default-rtdb.firebaseio.com/opr.json'
8  operators = requests.get(base_url)
9  opr_json = operators.json()
10 operator = opr_json["div"]
11
12 result = solve(operator)
13
14 print(result)
15
16
```

Taint.config

```
taint.config x
pysa_tutorial > Task-1 > taint.config
1 {
2   "sources": [
3     {
4       "name": "CustomUserControlled",
5       "comment": "use to annotate user input"
6     },
7     {
8       "name": "WebUserControlled",
9       "comment": "use to annotate user input"
10    }
11  ],
12
13  "sinks": [
14    {
15      "name": "CodeExecution",
16      "comment": "use to annotate execution of python code"
17    }
18  ],
19
20  "features": [],
21
22  "rules": [
23    {
24      "name": "Possible RCE:",
25      "code": 5001,
26      "sources": [ "CustomUserControlled" ],
27      "sinks": [ "CodeExecution" ],
28      "message_format": "User specified data may reach a code execution sink"
29    },
30    {
31      "name": "Possible RCE2:",
32      "code": 5002,
33      "sources": [ "WebUserControlled" ],
34      "sinks": [ "CodeExecution" ],
35      "message_format": "User specified data from web may reach a code execution sink"
36    }
37  ]
38 }
39
```

sources_sinks.pysa

```
sources_sinks.pysa x
Task-1 > sources_sinks.pysa
1 def requests.api.get() -> TaintSource[WebUserControlled]: ...
2
3 def eval(__source: TaintSink[CodeExecution], __globals, __locals): ...
4
5
6
```

.pyre_configuration

```
{ } .pyre_configuration X
pysa_tutorial > Task-1 > { } .pyre_configuration > ...
1  {
2    "source_directories": [
3      "."
4    ],
5    "taint_models_path": [
6      "."
7    ],
8    "search_path": [
9      "../../stubs/"
10   ],
11   "exclude": [
12     ".*integration_test/.*"
13   ]
14 }
15
```

Sub Task – 2

For the Sub Task-2, you will run Pysa (Static Analyzer) and by going through its output you will answer the questions. Below is the Pysa Output for Task-1

```
(tutorial) task1@msan107:~/Pysa-Experiment/pysa_tutorial/Task-1$ pyre analyze
X Found untracked type 'google.protobuf.descriptor_pb2.FieldDescriptorProto_Label.ValueType' when checking for attribute 'DESCRIPTOR' of 'google.protobuf.internal.enum_type_wrapper._EnumTypeWrapper'.
X Found untracked type 'google.protobuf.descriptor_pb2.FieldDescriptorProto_Type.ValueType' when checking for attribute 'DESCRIPTOR' of 'google.protobuf.internal.enum_type_wrapper._EnumTypeWrapper'.
X Found untracked type 'google.protobuf.descriptor_pb2.FieldOptions_CType.ValueType' when checking for attribute 'DESCRIPTOR' of 'google.protobuf.internal.enum_type_wrapper._EnumTypeWrapper'.
X Found untracked type 'google.protobuf.descriptor_pb2.FieldOptions_JSType.ValueType' when checking for attribute 'DESCRIPTOR' of 'google.protobuf.internal.enum_type_wrapper._EnumTypeWrapper'.
X Found untracked type 'google.protobuf.descriptor_pb2.FileOptions_OptimizeMode.ValueType' when checking for attribute 'DESCRIPTOR' of 'google.protobuf.internal.enum_type_wrapper._EnumTypeWrapper'.
X Found untracked type 'google.protobuf.descriptor_pb2.MethodOptions_IdempotencyLevel.ValueType' when checking for attribute 'DESCRIPTOR' of 'google.protobuf.internal.enum_type_wrapper._EnumTypeWrapper'.
X Found untracked type 'google.protobuf.compiler_plugin_pb2.CodeGeneratorResponse_Feature.ValueType' when checking for attribute 'DESCRIPTOR' of 'google.protobuf.internal.enum_type_wrapper._EnumTypeWrapper'.
X Found untracked type 'google.protobuf.type_pb2.Field_Cardinality.ValueType' when checking for attribute 'DESCRIPTOR' of 'google.protobuf.internal.enum_type_wrapper._EnumTypeWrapper'.
X Found untracked type 'google.protobuf.type_pb2.Field_Kind.ValueType' when checking for attribute 'DESCRIPTOR' of 'google.protobuf.internal.enum_type_wrapper._EnumTypeWrapper'.
X Found untracked type 'google.protobuf.type_pb2_Syntax.ValueType' when checking for attribute 'DESCRIPTOR' of 'google.protobuf.internal.enum_type_wrapper._EnumTypeWrapper'.
X Found untracked type 'google.protobuf.struct_pb2_NullValue.ValueType' when checking for attribute 'DESCRIPTOR' of 'google.protobuf.internal.enum_type_wrapper._EnumTypeWrapper'.
X 'google.protobuf.message.Message:ClearField' has 57 overrides, this might slow down the analysis considerably.
X 'google.protobuf.message.Message:__init__' has 58 overrides, this might slow down the analysis considerably.
X 'object:==_eq_' has 185 overrides, this might slow down the analysis considerably.
X 'object:hash_' has 75 overrides, this might slow down the analysis considerably.
X 'object:__init__' has 1328 overrides, this might slow down the analysis considerably.
X 'object:__ne_' has 88 overrides, this might slow down the analysis considerably.
X 'type:call_' has 190 overrides, this might slow down the analysis considerably.
X 'type:__init__' has 1224 overrides, this might slow down the analysis considerably.
X 'type:__new__' has 165 overrides, this might slow down the analysis considerably.
X 'typing.Collection:__len__' has 51 overrides, this might slow down the analysis considerably.
X 'typing.GenericMeta:__getitem__' has 63 overrides, this might slow down the analysis considerably.
X 'typing.Iterable:__iter__' has 51 overrides, this might slow down the analysis considerably.
X 'typing.NamedTuple:__init__' has 112 overrides, this might slow down the analysis considerably.
{
  "line": 12,
  "column": 15,
```

```
[
  {
    "line": 12,
    "column": 15,
    "stop_line": 12,
    "stop_column": 23,
    "path": "views.py",
    "code": 5002,
    "name": "Possible RCE2:",
    "description": "Possible RCE2: [5002]: User specified data from web may reach a code execution sink",
    "define": "views.$toplevel"
  }
]
```

Sub Task – 3

For the Sub Task-3, you will run the dynamic debugger and try to track the flow of data. In this Sub Task you will use the Output of Pysa along with the Output of the dynamic debugger and then try to answer the questions. Below is the Output of the debugger.







