Peach Fuzzer™

Converting Peach Pits from Version 2.3

This document outlines the conversion details to update Peach fuzzing definitions (Peach Pits) from the Peach Platform version 2.3 to version 3.x. The changes include a few global issues; advancements in methodology that translate to changes in the description language; and product improvements, bug fixes, and enhancements. Once finished, you will be able to run the Peach Fuzzer Platform with the converted Pit using the command-line interface.

Global Changes

The global changes consist of the following items that you need to change for all version 2.3 Pits.

- The <Peach> element *xmlsns* and *xsi:schemaLocation* attribute values have changed. The simplest fix is to replace the v2.3 `<Peach ….>` element with the following:

```
<!-- Peach v3.x  -->
<Peach xmlns="http://peachfuzzer.com/2012/Peach"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://peachfuzzer.com/2012/Peach peach.xsd">
```

- The <include> element for `defaults.xml` is no longer used. Remove the include statement from the Pit. An example of the statement to remove follows:

```
<!-- Peach v2.3  -->
<Include ns="default" src="file:defaults.xml"/>
```

- The relationship "*from*" is no longer used. Remove all "*from*" relations. An example statement follows:

```
<!-- Peach v2.3  -->
<Relation type="size" from="Length" />
```

Peach v2.3 provided "*from*" relations to boost performance by specifying both sides of a relationship using "of" and "from" parameters, as in the following example.

```
<!-- Peach v2.3  -->
<Number name="Length" size="32" endian="network" signed="false">
 <Relation type="size" of="Data" />
</Number>
<Blob name="Data">
 <Relation type="size" from="Length" />
</Blob>
```

Peach v3.x does not use the "from" parameter. The following example, written for peach v3.x, provides identical functionality to the previous example.

```
<!-- Peach v3.x  -->
<Number name="Length" size="32" endian="network" signed="false">
 <Relation type="size" of="Data" />
</Number>
<Blob name="Data"/>
```

- The <Logger> element is now part of the <Test> element. Move the Logger element block into the Test element block. For more information, see the next item.
- The functionality of the <Run> element is now part of the <Test> element. Upon moving the Logger element block into the Test element block, remove the Run element block from the Pit. An example follows.

```
<!--  Peach v2.3 Test and Run elements sample    -->
<Test name="UdpResp">
    <Agent ref="LocalAgent" />
    <StateModel ref="UdpTransaction"/>
    <Publisher class="Udp">
        <Param name="host" value="192.168.1.3"/>
        <Param name="port" value="53"/>
    </Publisher>
</Test>
<Run name="DefaultRun">
    <Logger class="logger.Filesystem">
        <Param name="Path" value="logs"/>
    </Logger>

    <Test ref="UdpResp"/>
</Run>
```

In Peach v3.x, the Test element block identifies the Agent, StateModel (and, by implication, the DataModel), Publisher, and Logger for a fuzzing session. A v3.x Test element block follows. The block is functionally identical to the previous example.

```
<!--  Peach v3.x Test element sample    -->
<Test name="Default">
    <Agent ref="LocalAgent" />
    <StateModel ref="UdpTransaction"/>
    <Publisher class="Udp">
        <Param name="Host" value="192.168.1.3"/>
        <Param name="Port" value="53"/>
    </Publisher>
                                                        <Logger
class="logger.Filesystem">
        <Param name="Path" value="logs"/>
    </Logger>
</Test>
```

NOTE: In the example, the name for the Test element is “Default”. This is the default name for the Test element. At runtime, Peach v3.x automatically looks for and runs the default Test, unless you specify a Test name on the command line.

- The parameter names for Monitors and Publishers now use CamelCasing. In the previous example, the param names “Host” and “Port” for the publisher have changed slightly due to CamelCasing. You can use the Peach DOM reference to check parameters that fail validation.
  Run “peach --showenv” to generate the DOM reference, or refer to the documentation.

TIP: You can place the data and state model definitions in separate xml files to improve re-use of these definitions. Once defined, you can the pull these definitions into different Pits to perform different tests on the same data and state models. Specify the file containing the models using the include xml element.

The following example shows file fragments of a Pit and definition files that contain the state and data models. Two include elements are used: 1) the main Pit file includes the state model, and 2) the state model file includes the data model.

```
<!--  Peach v3.x Pit file   -->
<Peach xmlns=http://peachfuzzer.com/2012/Peach
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=http://peachfuzzer.com/2012/Peach peach.xsd
author="Deja Vu Security, LLC"     description="Fuzz an FTP server in
active mode"     version="0.0.2">


    <!--  Pull the StateModel into the Pit.  -->    ❶
    <Include ns="FTP"
src="file:##PitLibraryPath##/_Common/Models/Net/ftp_State.xml"/>


    <!--  et cetera   -->


<!--  Peach v3.x StateModel definitions (ftp_State.xml)  -->
<Peach xmlns="http://peachfuzzer.com/2012/Peach"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://peachfuzzer.com/2012/Peach peach.xsd"
author="Deja Vu Security, LLC"  description="File Transfer Protocol PIT
StateModels" version="0.0.2">     <!--
Pull the DataModel into the StateModel.  -->    ❷
    <Include ns="FTP"
src="file:##PitLibraryPath##/_Common/Models/Net/ftp_Data.xml"/>
    <Import import="ftp" />


    <!-- FTP Passive Mode-->
    <StateModel name="ClientPassive" initialState="InputUserName">


    <!--  et cetera   -->
```

Changes to Individual xml Elements

The following Peach Platform v2.3 xml elements require changes when used with Peach Platform v3.x.

<Defaults>

The Defaults element contains default values for parameter definitions. If an individual element does not specify an optional parameter, the Platform uses the value specified in this element block. Values for optional attributes and parameters are defined in this element.

Note that the "Size" attribute of the number element is a required attribute that must be specified with each number instance.

<Import>

This xml element has one attribute, *import*, that names the python file containing code. Note the ".py" postfix is not used.

In Platform v3.x, you must specify each python file you want to use. Wildcard characters (*) are not supported.

The *from* attribute is now a top-level element named <PythonPath> that specifies the search path for all python modules. Note that a trailing "\" or "/" for the path is not used. Use multiple <PythonPath> elements to tell Peach to search in more than one place.

<DataModel>

Remove all "from" relation statements from all data model elements.

<Flag>

A multiple-bit Flag that uses the value parameter accepts a value expressed as a hexadecimal integer or a sequence of hexadecimal digits.

- A value expressed as a hexadecimal integer must fit into the bits allocated for the flag.
- A value expressed as a sequence of hexadecimal digits must have sufficient length to span the number of bits in the flag.

<Number>

When specifying a value for a number, you can use an integer value, a hexadecimal integer value or a sequence of hexadecimal digits.

A value expressed as a sequence of hexadecimal digits (where valueType="hex") must match lengthwise with the allocated size of the number or a validation error occurs. For example, initially setting a 64-bit number to one can be specified as seven digits of zeroes and one digit of one:

```
<!-- Peach v3.x  -->
<Number size="64" valueType="hex" value="00 00 00 00 00 00 00 01" />
```

For values expressed as hex integers prefix the value with “0x”.

```
<!-- Peach v3.x  -->
<Number size="64" value="0x01" />
```

Size is a required attribute. You cannot use a default size specified in the Defaults element block for number elements.

<StateModel>

No changes.

<Test>

Now includes logger definitions, and performs the functionality of the Platform v2.3 Run xml element.

<Run>

This section is obsolete. Move the Logger into the test section.

<Publisher>

<raw.RawIp> RawIpv4

Host and *Protocol* are required parameters

The *Protocol* parameter is required and must have a valid value. “17” is the value for UDP. “6” is the value for TCP.

The *Host* parameter is required and must have a valid value, specified as a hostname or an IP address.

The *Interface* parameter now uses CamelCase with a capitalized first letter. This parameter is optional.

NOTE: While the old publisher name is valid, the current name RawIpv4 is the name used in the documentation and in log messages from the Peach Platform.