HSASec - IT Security and Forensics
University of Applied Sciences Augsburg

Extending models for numbers and binary data in fuzzing-frameworks to improve convertibility

Florian Schmidt
Research Group IT Security and Forensics
Augsburg University of Applied Sciences
Augsburg, Germany

What is fuzzing?

Fuzzing or fuzz-testing is the automated input of semi-valid data to an application interface or a protocol stack.

Fuzzing frameworks

Fuzzing frameworks or APIs (Advanced Programming Interfaces) facilitate modeling of data structures and workflows. These techniques are used to create an abstract description of the tested system, called protocol or interface definition (PoID).

Data models

Sulley

```
s_initialize('dataModel1')
s_string('value')
s_byte(5, signed=False)
```

Peach

```
<DataModel name="dataModel1">
  <String value="value"/>
  <Number value="5" size="8"
          signed="false"/>
</DataModel>
```

Number and binary data elements

API	object	size/width	endianness	sign	encoding
Sulley	byte word dword qword	8 bit 16 bit 32 bit 64 bit	*endian* *little* *big*	signed	*format* *ascii* *binary*
Peach	Number	1 - 64 bit	*endian* *little* *big* *network*	signed	*valueType* *string* *hex* *literal*

API	object	endianness	sign	encoding
Sulley	bit_field	*endian* *little* *big*	signed	*format* *ascii* *binary*
Peach	Blob	X	X	*valueType* *string* *hex* *literal*

Data models

Data models are a collection of so called data elements.

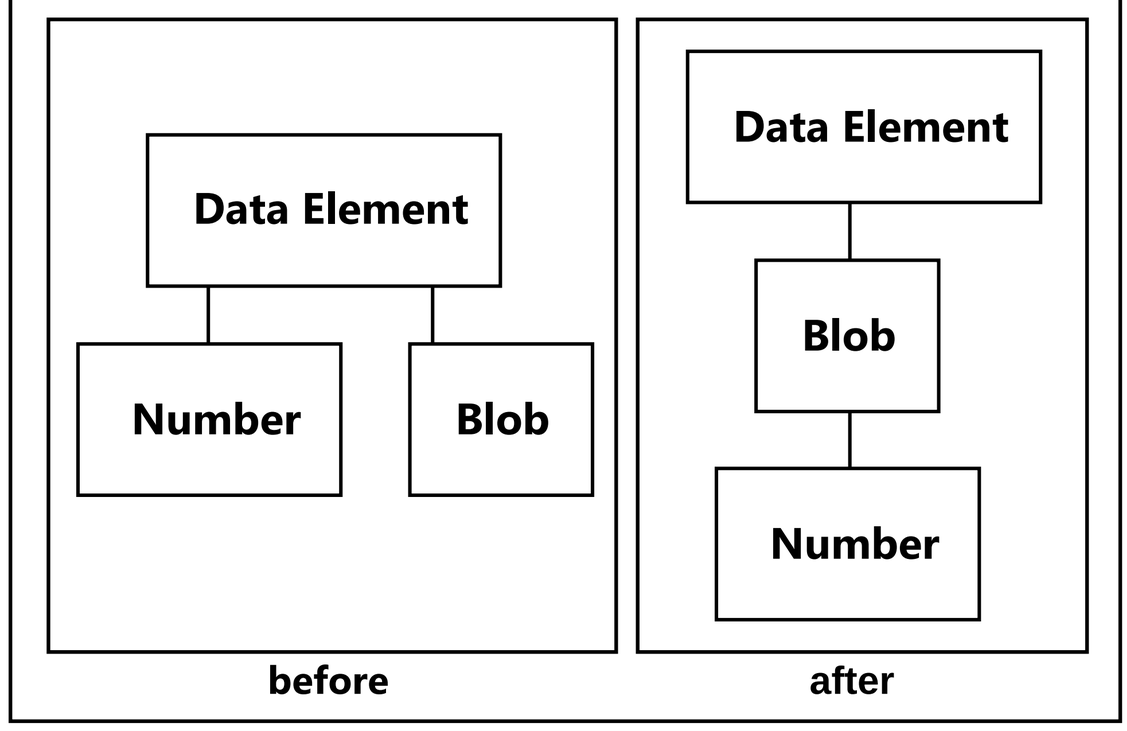
DataElements

Data elements are used to store values of a specific type and additional information about its binary representation.

Modifications

The incompatibilities of binary and number element types between Sulley and Peach, could be resolved by applying modifications to the Peach-API. The Blob data element was supplemented with the attributes endian and sign and now serves as base type of the *Number* data element.

Framework modifications



→ using the same base type for binary and number elements improves convertibility and expressiveness of the data model