```
1 """
2 preprocessing: Implements the transformation from o
                                                              2 preprocessing: Implements the transformation from o
   bjdump output to ROMEO text representation.
                                                                bjdump output to ROMEO text representation.
  11 11 11
                                                                11 11 11
3
4
                                                              4
5 import enum
                                                              5 import enum
6 import logging
                                                              6 import logging
7 import random
                                                              7 import random
8 import re
                                                              8 import re
  import typing
                                                              9 import typing
10
                                                             10
11 import disassembler
                                                             11 import disassembler
12
                                                             12
13
                                                             13
14 class Granularity(enum.Enum):
                                                             14 class Granularity(enum.Enum):
      INSTRUCTION = enum.auto()
                                                             15
                                                                    INSTRUCTION = enum.auto()
16
       FUNCTION = enum.auto()
                                                             16
                                                                    FUNCTION = enum.auto()
17
      TRANSLATION_UNIT = enum.auto()
                                                             17
                                                                    TRANSLATION_UNIT = enum.auto()
       OBJECT = enum.auto()
                                                             18
                                                                    OBJECT = enum.auto()
18
19
                                                             19
20
                                                             20
21 class LabelStrategy(enum.Enum):
                                                             21 class LabelStrategy(enum.Enum):
       BINARYCLASSIFICATION = enum.auto()
                                                                    BINARYCLASSIFICATION = enum.auto()
22
                                                             22
23
       MULTICLASSCLASSIFICATION = enum.auto()
                                                             23
                                                                    MULTICLASSCLASSIFICATION = enum.auto()
24
                                                             24
26 def validate_testcase_files(testcase) -> bool:
                                                             26 def validate_testcase_files(testcase) -> bool:
       """This hook can be used in the future to add c
                                                                    """This hook can be used in the future to add c
   riteria to testcases and stop the pipeline if they
                                                                riteria to testcases and stop the pipeline if they
    are not met."""
                                                                 are not met."""
       return True
                                                             28
                                                                    return True
28
29
                                                             29
31 def _build_symbol_translation_table(disassembly: li
                                                             31 def _build_symbol_translation_table(disassembly: li
   st[dict]) -> dict:
                                                                st[dict]) -> dict:
      """Constructs a symbol translation table for th
                                                                    """Constructs a symbol translation table for th
   e given disassembly, assigning random names to non-
                                                                e given disassembly, assigning random names to non-
   dynamic functions."""
                                                                dynamic functions."""
       symbol_translation_table = {}
                                                                    symbol_translation_table = {}
34
       scrambled_symbol_count = 0
                                                                    scrambled_symbol_count = 0
                                                                    for name, symbol in disassembly["symbols"].item
35
       for name, symbol in disassembly["symbols"].item
36
           # Local function names should be hidden!
                                                             36
                                                                        # Local function names should be hidden!
                                                                        if "F" in symbol["Flags"] and "D" not in sy
           if "F" in symbol["Flags"] and "D" not in sy
                                                             37
   mbol["Flags"]:
                                                                mbol["Flags"]:
              while True:
38
                                                                            while True:
39
                   proposed_translation = f"lc{random.
                                                             39
                                                                                proposed_translation = f"lc{random.
   randint(1,999):03d}"
                                                                 randint(1,999):03d}"
40
                  if proposed_translation not in symb
                                                                                if proposed_translation not in symb
   ol_translation_table.values():
                                                                ol_translation_table.values():
41
                       symbol_translation_table[name]
                                                             41
                                                                                     symbol_translation_table[name]
    = proposed_translation
                                                                 = proposed_translation
42
                                                             42
               scrambled_symbol_count += 1
                                                             43
                                                                            scrambled_symbol_count += 1
43
44
               if scrambled_symbol_count > 900:
                                                             44
                                                                            if scrambled_symbol_count > 900:
45
                   raise ValueError(
                                                             45
                                                                                 raise ValueError(
                       f"Too many symbols to scramble
                                                                                     f"Too many symbols to scramble
    for disassembly {disassembly['name']}"
                                                                 for disassembly {disassembly['name']}"
47
                                                             47
                   )
                                                                                )
           else:
48
                                                             48
                                                                         else:
49
               symbol_translation_table[name] = name
                                                             49
                                                                             symbol_translation_table[name] = name
                                                             50
50
       return symbol_translation_table
                                                                     return symbol_translation_table
51
                                                             51
52
                                                             52
```

```
53
                                                             53
54 def _sanitize_symbol(symbol: str):
                                                             54 def _sanitize_symbol(symbol: str):
       """Remove PLT artifacts etc. from symbols for \ensuremath{\text{e}}
                                                                     """Remove PLT artifacts etc. from symbols for e
                                                             55
   asier matching."""
                                                                 asier matching."""
       sanitized_symbol = str(symbol)
                                                              56
                                                                     sanitized_symbol = str(symbol)
56
       # 1 Remove multiple @ signs
                                                              58
                                                                     # 1 Remove multiple @ signs
       while "@@" in sanitized_symbol:
59
                                                                     while "@@" in sanitized_symbol:
           sanitized_symbol = sanitized_symbol.replace
                                                                         sanitized_symbol = sanitized_symbol.replace
60
                                                             60
   ("@@", "@")
                                                                 ("@@", "@")
                                                              61
61
62
       # 2 Remove things after @ sign
                                                              62
                                                                     # 2 Remove things after @ sign
       sanitized_symbol = sanitized_symbol.rsplit("@",
                                                                     sanitized_symbol = sanitized_symbol.rsplit("@",
63
                                                              63
   1)[0]
                                                                 1)[0]
64
                                                              64
65
       return sanitized_symbol
                                                              65
                                                                     return sanitized_symbol
66
                                                              66
67
                                                              67
   def _maybe_translate_symbol(
                                                              68 def _maybe_translate_symbol(
68
       symbol: str, disassembly: list[dict], logger: l
                                                                     symbol: str, disassembly: list[dict], logger: l
   ogging.Logger
                                                                 ogging.Logger
70
                                                              70 ):
       """Attemt to use the symbol translation table t
                                                                     """Attemt to use the symbol translation table t
   o translate a symbol, otherwise return the original
                                                                 o translate a symbol, otherwise return the original
   svmbol."""
                                                                 symbol."""
72
       symbol = _sanitize_symbol(symbol)
                                                              72
                                                                     symbol = _sanitize_symbol(symbol)
       if symbol in disassembly["symbol_translation_ta
                                                                    if symbol in disassembly["symbol_translation_ta
   ble"1.kevs():
                                                                 ble"1.kevs():
           return disassembly["symbol_translation_tabl
                                                                         return disassembly["symbol_translation_tabl
   e"][symbol]
                                                                 e"][symbol]
75
       else:
                                                              75
76
                                                              76
                                                                         logger.warning(
           logger.warning(
               f"Could not find symbol {symbol} in sym
                                                                             f"Could not find symbol {symbol} in sym
   bol translation table for disassembly {disassembly
                                                                 bol translation table for disassembly {disassembly
   ['name']}"
                                                                 ['name']}"
78
                                                              78
           )
                                                                         )
79
           return symbol
                                                              79
                                                                         return symbol
                                                              80
80
81
                                                              81
82 def _preprocess_disassembly(
                                                              82 def _preprocess_disassembly(
83
       disassembly: list[dict], logger: logging.Logger
                                                             83
                                                                     disassembly: list[dict], logger: logging.Logger
84 ) -> list[dict]:
                                                              84 ) -> list[dict]:
       """Preprocesses a disassembly, replacing operan
                                                                     """Preprocesses a disassembly, replacing operan
85
   d addresses with symbols and adding labels to funct
                                                                 d addresses with symbols and adding labels to funct
                                                                 ions."""
86
       disassembly["symbol_translation_table"] = _buil
                                                                     disassembly["symbol_translation_table"] = _buil
   d_symbol_translation_table(
                                                                 d_symbol_translation_table(
87
           disassembly
                                                              87
                                                                         disassembly
88
       )
                                                              88
89
                                                              89
       for _, section in disassembly["sections"].items
                                                                     for _, section in disassembly["sections"].items
90
                                                              90
   ():
                                                                 ():
           for function_name, function in dict(section
                                                                         for function_name, function in dict(section
   ["functions"]).items():
                                                                 ["functions"]).items():
92
               instruction_text_representations = []
                                                             92
                                                                             instruction_text_representations = []
93
               dependencies = []
                                                              93
                                                                             dependencies = []
               for instruction in function["instructio
                                                                             for instruction in function["instructio
   ns"]:
                                                                 ns"]:
95
                    instruction\_text\_representation: st
                                                              95
                                                                                  instruction_text_representation: st
                                                                 r = ""
                    # 1 Operation
                                                              96
                                                                                 # 1 Operation
96
                   instruction_text_representation +=
                                                                                 instruction_text_representation +=
97
                                                             97
    instruction["od_operation"]
                                                                  instruction["od_operation"]
98
                                                              98
                    # 2 Comment
                                                                                  # 2 Comment
99
                                                             99
                    \hbox{if instruction["od\_comment"] is not}\\
                                                                                  if instruction["od_comment"] is not
   None:
                                                                 None:
```

```
101
                         square_brackets = re.findall(
                                                               101
                                                                                        square_brackets = re.findall(
                             r"(?:QWORD PTR )?\[.+\]", i
                                                                                             r"(?:QWORD PTR )?\[.+\]", i
                                                               102
    nstruction["od_operand"]
                                                                    nstruction["od_operand"]
103
                                                               103
104
                         if len(square brackets) == 1:
                                                               104
                                                                                        if len(square brackets) == 1:
105
                             # Exactly one operand with
                                                               105
                                                                                             # Exactly one operand with
      an address, let's replace it with a comment
                                                                     an address, let's replace it with a comment
                             operand_to_replace = square
                                                               106
106
                                                                                             operand to replace = square
     brackets[0]
                                                                    brackets[0]
107
                             instruction["od_operand"] =
                                                               107
                                                                                             instruction["od_operand"] =
    instruction["od_operand"].replace(
                                                                    instruction["od_operand"].replace(
108
                                                               108
                                  operand to replace, ins
                                                                                                 operand_to_replace, ins
    truction["od_comment"]
                                                                    truction["od_comment"]
109
                                                               109
110
                             instruction["od_comment"] =
                                                               110
                                                                                             instruction["od_comment"] =
    None
                                                                    None
111
                         else:
                                                               111
                                                                                        else:
112
                             logger.warning(
                                                               112
                                                                                             logger.warning(
113
                                  f"Could not uniquely ma
                                                               113
                                                                                                 f"Could not uniquely ma
    tch operand with comment for instruction: {str(inst
                                                                    tch operand with comment for instruction: {str(inst
    ruction)}"
                                                                    ruction)}"
                                                               114
115
                                                               115
116
                     # 3 Operand
                                                               116
                                                                                    # 3 Operand
                     if instruction["od_operand"] is not
                                                                                    if instruction["od_operand"] is not
                                                               117
    None:
                                                                    None:
118
                         instruction text representation
                                                               118
                                                                                        instruction_text_representation
    += " " + instruction["od_operand"]
                                                                    += " " + instruction["od_operand"]
119
                                                               119
120
                         if (
                                                               120
                                                                                        if (
                              len(
                                                                                             len(
121
                                                               121
122
                                  matches := re.findall(
                                                               122
                                                                                                 matches := re.findall(
                                      r''[0-9a-f]+ <.+>''
                                                                                                     r''[0-9a-f]+ <.+>''
123
                                                               123
     instruction text representation
                                                                     instruction text representation
                                                               124
124
125
                             )
                                                               125
                                                                                             )
126
                               0
                                                                                               0
                                                               127
127
                         ):
                                                                                        ):
128
                             if len(matches) != 1:
                                                               128
                                                                                             if len(matches) != 1:
129
                                  logger.warning(
                                                               129
                                                                                                 logger.warning(
130
                                      f"Found multiple ma
                                                                                                     f"Found multiple ma
    tches for instruction {instruction_text_representat
                                                                    tches for instruction {instruction_text_representat
    ion} in function {function['name']} in section {sec
                                                                    ion} in function {function['name']} in section {sec
    tion['name']} in disassembly {disassembly['name']}"
                                                                    tion['name']} in disassembly {disassembly['name']}"
                                                               131
131
                                  )
132
                             operand_in_question = match
                                                               132
                                                                                             operand_in_question = match
    es[0]
                                                                    es[0]
133
                                                               133
                                                                                             start_index = operand_in_qu
                             start index = operand in qu
    estion.index("<")
                                                                    estion.index("<")
134
                             end_index = operand_in_ques
                                                               134
                                                                                             end_index = operand_in_ques
    tion.rfind(">")
                                                                    tion.rfind(">")
135
                             expression = operand in que
                                                               135
                                                                                             expression = operand in que
    stion[start_index + 1 : end_index]
                                                                    stion[start_index + 1 : end_index]
136
                                                               136
137
                              expression\_offset\_index = m
                                                               137
                                                                                             expression\_offset\_index = m
    ax(
                                                                   ax(
138
                                  expression.rfind("+"),
                                                               138
                                                                                                 expression.rfind("+"),
      expression.rfind("-")
                                                                     expression.rfind("-")
139
                                                               139
                             )
                                                                                             )
                             if (
140
                                                               140
                                                                                             if (
                                  expression_offset_index
                                                               141
                                                                                                 expression_offset_index
                                                               142
142
                                 and re.match(
                                                                                                 and re.match(
                                      r"[\+\-][0-9a-fx]",
                                                                                                     r"[\+\-][0-9a-fx]",
143
                                                               143
    expression[expression_offset_index:]
                                                                    expression[expression_offset_index:]
                                  )
                                                               144
                                                                                                 )
```

```
145
                                                                145
                                  is not None
                                                                                                  is not None
                                                                                              ):
146
                                                                146
                              ):
147
                                                                147
                                  offset = expression[exp
                                                                                                  offset = expression[exp
     ression_offset_index:]
                                                                     ression_offset_index:]
148
                              else:
                                                                148
                                                                                              else:
149
                                  expression_offset_index
                                                                149
                                                                                                  expression_offset_index
     = -1
150
                                  offset = ""
                                                                150
                                                                                                  offset = ""
152
                              symbol\_end\_index = (
                                                                152
                                                                                              symbol\_end\_index = (
                                                                                                  expression_offset_index
                                  expression_offset_index
153
                                                                                                  if expression_offset_in
154
                                  if expression offset in
                                                                154
    dex = -1
                                                                    dex != -1
155
                                  else len(expression)
                                                                155
                                                                                                  else len(expression)
156
                                                                156
157
                                                                157
                              symbol = expression[:symbol
                                                                                              symbol = expression[:symbol
     _end_index]
                                                                     _end_index]
159
                              dependencies += [_sanitize_
                                                                159
                                                                                              dependencies += [_sanitize_
     symbol(symbol)]
                                                                     symbol(symbol)]
160
                                                                160
                              symbol = _maybe_translate_s
                                                                                              symbol = _maybe_translate_s
161
                                                                161
    ymbol(symbol, disassembly, logger)
                                                                     ymbol(symbol, disassembly, logger)
162
                                                                162
163
                              \verb"instruction_text_representa"
                                                                163
                                                                                              instruction_text_representa
     tion = (
                                                                     tion = (
164
                                                                164
                                  instruction_text_repres
                                                                                                  instruction_text_repres
    entation.replace(
                                                                    entation.replace(
165
                                      operand_in_questio
                                                                165
                                                                                                      operand_in_questio
    n, f"{symbol}{offset}"
                                                                     n, f"{symbol}{offset}"
166
                                                                166
                                                                                                  )
167
                              )
                                                                167
                                                                                              )
168
                                                                168
169
                     instruction_text_representations +=
                                                                                      instruction_text_representations +=
170
                          instruction_text_representatio
                                                                170
                                                                                          instruction_text_representatio
    n.strip()
                                                                    n.strip()
171
                     ]
                                                                171
172
                                                                172
                                                                                 \hbox{if len(instruction\_text\_representation}\\
173
                 \hbox{if len(instruction\_text\_representation}\\
                                                                173
     s) > 0:
                                                                     s) > 0:
174
                     function_text_representation = (
                                                                174
                                                                                      function_text_representation = (
175
                         f"!{_maybe_translate_symbol(fun
                                                                175
                                                                                          f"\,!\,\{\_maybe\_translate\_symbol(fun
     ction_name, disassembly, logger)}:\n"
                                                                     ction_name, disassembly, logger)}:\n"
176
                          + "\n".join(instruction_text_re
                                                                                          + "\n".join(instruction_text_re
                                                                    presentations)
    presentations)
177
                     )
                                                                177
                                                                                      )
                                                                178
178
179
                     function["TextRepresentation"] = fu
                                                                179
                                                                                      function["TextRepresentation"] = fu
    nction text representation
                                                                    nction_text_representation
180
                     function["Dependencies"] = dependen
                                                                180
                                                                                      function["Dependencies"] = dependen
181
                 else:
                                                                181
                                                                                 else:
                     del section["functions"][function_n
                                                                                     del section["functions"][function_n
182
                                                                182
     ame]
                                                                     ame]
183
                                                                183
184
         return disassembly
                                                                184
                                                                         return disassembly
185
                                                                185
                                                                186
                                                                187
                                                                     def extract_examples_malware(testcase: dict,
                                                                                                    label_granularity: typ
                                                                188
                                                                     ing.Optional[Granularity],
                                                                189
                                                                                                   context_granularity: t
                                                                     yping.Optional[Granularity],
                                                                190
                                                                                                   emit_primary_good_func
                                                                     tion: bool,
```

```
191
                                  logger: logging.Logge
192
                                  ) -> list[dict]:
193
        """Transform a testcase into a set of examples
     by disassembling and processing them."""
        # First, disassemble all objects
194
195
        object_files = [
196
197
            for file in testcase["Files"]
198
            if file["Path"].name.endswith(".o")
               and not file["Path"].name.startswith("li
199
    nked-")
201
        if not len(object_files) == 0:
202
            object_file = object_files[0]
203
            elf_path = object_file["Path"]
204
205
206
            disassembled_code = disassembler.disassembl
    e_elf([elf_path], logger)
207
            disassembly = _preprocess_disassembly(disas
    sembled_code, logger)
208
209
            examples = []
            if label_granularity == Granularity.FUNCTIO
210
211
                text_section_functions = disassembly["s
    ections"][".text"]["functions"]
212
213
                # Decide which functions to use as exam
    ples
                primary_good_functions = []
214
215
                primary_bad_functions = []
216
                secondary_good_functions = []
217
218
                if str(disassembly['name']).endswith("_
    good.o"):
219
                    primary_good_functions.append(disas
    sembly['name'])
220
221
                    primary_bad_functions.append(disass
    embly['name'])
222
223
                # Emit labeled functions
                emitted_functions = secondary_good_func
    tions + primary bad functions
                if emit_primary_good_function:
226
                     emitted_functions += primary_good_f
    unctions
227
                 for function_name in emitted_functions:
229
                     function = text_section_functions
    [" start"]
                     text_representation = ""
230
231
                     for text_section in text_section_fu
    nctions:
232
                        text_representation += text_sec
    tion_functions[text_section]["TextRepresentation"]
233
                     example = {
234
                         "Example": text_representation,
235
                         "Testcase": testcase,
                         "GoodOrBad": "Good"
                         if function_name not in primary
     bad functions
238
                        else "Bad",
239
                    }
                     examples += [example]
```

```
242
                                                                            else:
                                                              243
                                                                                raise ValueError("Granularity not suppo
                                                                    rted")
                                                              244
                                                                            return examples
                                                              245
                                                              246
                                                                       else:
                                                                            print("No Object Files")
                                                              248
                                                                            exit(1)
                                                               249
                                                              250
    def extract_examples(
                                                              251
                                                                   def extract_examples(
                                                              252
        testcase: dict.
                                                                       testcase: dict,
189
        label_granularity: typing.Optional[Granularit
                                                              253
                                                                        label_granularity: typing.Optional[Granularit
    у],
                                                                   у],
190
        context_granularity: typing.Optional[Granularit
                                                              254
                                                                       context_granularity: typing.Optional[Granularit
    у],
                                                                   у],
191
        emit_primary_good_function: bool,
                                                              255
                                                                       emit_primary_good_function: bool,
192
        logger: logging.Logger,
                                                              256
                                                                       logger: logging.Logger,
193
    ) -> list[dict]:
                                                              257
                                                                   ) -> list[dict]:
        """Transform a testcase into a set of examples
                                                                        """Transform a testcase into a set of examples
                                                              258
                                                                    by disassembling and processing them."""
     by disassembling and processing them."""
                                                                       # First, disassemble all objects
195
        # First, disassemble all objects
                                                               259
196
        object files = Γ
                                                              260
                                                                       object files = [
197
            file
                                                              261
                                                                            file
            for file in testcase["Files"]
                                                                            for file in testcase["Files"]
198
                                                              262
199
            if file["Path"].name.endswith(".o")
                                                              263
                                                                            if file["Path"].name.endswith(".o")
            and not file["Path"].name.startswith("linke
                                                                            and not file["Path"].name.startswith("linke
200
                                                              264
    d-")
                                                                   d-")
201
                                                               265
202
                                                              266
                                                                       disassembly = _preprocess_disassembly(
        disassembly = _preprocess_disassembly(
                                                              267
203
204
            disassembler.disassemble_elf(
                                                              268
                                                                            disassembler.disassemble_elf(
205
                 [object_file["Path"] for object_file in
                                                                                [object_file["Path"] for object_file in
    object_files], logger
                                                                   object_files], logger
            ),
                                                                           ),
207
             logger,
                                                              271
                                                                            logger
208
        )
                                                              272
209
                                                              273
210
        examples = []
                                                              274
                                                                       examples = []
        if label_granularity == Granularity.FUNCTION:
                                                              275
                                                                       if label_granularity == Granularity.FUNCTION:
211
            text_section_functions = disassembly["secti
                                                                            text_section_functions = disassembly["secti
    ons"][".text"]["functions"]
                                                                   ons"][".text"]["functions"]
213
                                                              277
             # Decide which functions to use as examples
                                                                            # Decide which functions to use as examples
214
                                                              278
                                                              279
215
            primary good functions = []
                                                                            primary good functions = []
            primary_bad_functions = []
                                                                            primary_bad_functions = []
217
             secondary_good_functions = []
                                                              281
                                                                            secondary_good_functions = []
                                                              282
                                                                            for function name in text section function
219
            for function name in text section function
                                                              283
    s'
                                                                   s.
                 if re.match(r"^(CWE.*(_|::))?bad(^(\setminus))?
220
                                                              284
                                                                                if re.match(r''^(CWE.^*(\_|::))?bad((()))?
    $", function_name) is not None:
                                                                   $", function_name) is not None:
                     primary bad functions += [function
                                                              285
                                                                                    primary bad functions += [function
    name]
                                                                   name]
                 elif re.match(r"^(CWE.*(_|::))?good(\
                                                              286
                                                                                elif re.match(r"^(CWE.^*(\_|::))?good(\
    (\))?$", function_name) is not None:
                                                                   (\))?$", function_name) is not None:
                     primary_good_functions += [function
                                                                                    primary_good_functions += [function
223
                                                              287
     _name]
                                                                    _name]
224
                 elif (
                                                              288
                                                                                elif (
225
                     re.match(
                                                              289
                                                                                    re.match(
                         r"^(CWE.*(_|::))?good(\d+|G2B\d
                                                                                        r"^(CWE.*(_|::))?good(\d+|G2B\d
     *|B2G\d*)(\(\))?$", function_name
                                                                   *|B2G\d^*)(\(\))?$", function_name
227
                     )
                                                              291
                     is not None
                                                              292
                                                                                    is not None
229
                 ):
                                                              293
                                                                                ):
```

241

```
230
                    secondary_good_functions += [functi
                                                                                  secondary good functions += [functi
    on namel
                                                                  on namel
231
                                                              295
            # See if all functions are there
                                                                          # See if all functions are there
232
                                                              297
                                                                          \# Ignore missing good functions for "Bad-On
            # Ignore missing good functions for "Bad-On
    ly Test Cases" (see Appendix D in Juliet 1.2 doc)
                                                                   ly Test Cases" (see Appendix D in Juliet 1.2 doc)
235
                                                              299
236
                testcase["Weakness"]["WeaknessID"] != 5
                                                              300
                                                                               testcase["Weakness"]["WeaknessID"] != 5
    06
                                                                  06
237
                or testcase["FunctionalVariant"]
                                                                               or testcase["FunctionalVariant"]
                                                              301
238
                not in [
                                                              302
                                                                               not in [
                     "email",
                                                                                   "email",
239
                                                                                   "file_transfer_connect_socket",
240
                     "file_transfer_connect_socket",
                                                              304
241
                     "file_transfer_listen_socket",
                                                                                   "file_transfer_listen_socket",
242
                     "screen capture",
                                                              306
                                                                                   "screen capture",
243
                1
                                                              307
                                                                               1
            ) and (
                                                              308
                                                                           ) and (
244
                testcase["Weakness"]["WeaknessID"] != 5
                                                                               testcase["Weakness"]["WeaknessID"] != 5
                or testcase["FunctionalVariant"]
                                                                               or testcase["FunctionalVariant"]
246
                                                              310
247
                not in ["network_connection", "network_
                                                                               not in ["network_connection", "network_
    listen"]
                                                                   listen"]
                                                              312
248
            ):
                                                                          ):
                if not (len(primary_good_functions) >
                                                                               if not (len(primary_good_functions) >
                     len(secondary\_good\_functions) > 0
                                                                                   len(secondary_good_functions) > 0
250
                                                              314
251
                ):
                                                              315
                                                                               ):
252
                     logger.warning(
                                                              316
                                                                                   logger.warning(
                        f"Number of primary or secondar
                                                                                       f"Number of primary or secondar
    y good functions is zero for testcase \{str(testcas
                                                                  y good functions is zero for testcase {str(testcas
    e)}!"
                                                                  e)}!"
254
                                                              318
255
                                                              319
            if not (len(primary_bad_functions) > 0):
256
                                                              320
                                                                          if not (len(primary_bad_functions) > 0):
257
                logger.warning(
                                                              321
                                                                               logger.warning(
                     f"Number of primary bad functions i
                                                                                   f"Number of primary bad functions i
    s zero for testcase {str(testcase)}!"
                                                                  s zero for testcase {str(testcase)}!"
                                                              323
260
                                                              324
            # Emit labeled functions
                                                                           # Emit labeled functions
                                                              325
            emitted_functions = secondary_good_function
                                                                          emitted_functions = secondary_good_function
262
    s + primary_bad_functions
                                                                   s + primary_bad_functions
263
            if emit_primary_good_function:
                                                              327
                                                                           if emit_primary_good_function:
264
                emitted functions += primary good funct
                                                              328
                                                                               emitted functions += primary good funct
    ions
                                                                  ions
265
                                                              329
266
            for function_name in emitted_functions:
                                                              330
                                                                           for function_name in emitted_functions:
267
                function = text_section_functions[funct
                                                              331
                                                                               function = text section functions[funct
    ion name]
                                                                  ion namel
268
                                                              332
269
                example = {
                                                              333
                                                                               example = {
270
                     "Example": function["TextRepresenta
                                                              334
                                                                                   "Example": function["TextRepresenta
    tion"],
                                                                  tion"],
271
                     "Testcase": testcase,
                                                              335
                                                                                   "Testcase": testcase,
272
                     "GoodOrBad": "Good"
                                                              336
                                                                                   "GoodOrBad": "Good"
                     if function_name not in primary_bad
                                                              337
                                                                                   if function_name not in primary_bad
     functions
                                                                   functions
                     else "Bad",
                                                                                   else "Bad",
274
275
                                                              339
                }
                                                                               }
                if context_granularity == Granularity.F
                                                              340
                                                                               if context_granularity == Granularity.F
276
    UNCTION:
                                                                  UNCTION:
277
                                                              341
                elif context_granularity == Granularit
                                                              342
                                                                               elif context_granularity == Granularit
    y.OBJECT:
                                                                  y.OBJECT:
```

```
# Ignore functions that are not hel
                                                               343
                                                                                    # Ignore functions that are not hel
                                                                    pful or leak labels
    pful or leak labels
280
                     ignored_functions = [
                                                               344
                                                                                     ignored_functions = [
                         "_start",
                                                               345
                                                                                         "_start",
281
282
                         "__libc_csu_init",
                                                               346
                                                                                         "__libc_csu_init",
                         "__libc_csu_fini",
                                                                                         "__libc_csu_fini",
283
                                                               347
                         "_dl_relocate_static_pie",
                                                                                         "_dl_relocate_static_pie",
284
                                                               348
285
                         "deregister_tm_clones",
                                                               349
                                                                                         "deregister_tm_clones",
                         "register_tm_clones",
                                                               350
                                                                                         "register_tm_clones",
287
                         "__do_global_dtors_aux",
                                                               351
                                                                                         "__do_global_dtors_aux",
                         "frame_dummy",
                                                                                         "frame_dummy",
288
                                                               352
                                                               353
                     ] + [function_name]
                                                                                     ] + [function_name]
289
290
                                                               354
                     if function_name in primary_bad_fun
                                                               355
                                                                                     if function_name in primary_bad_fun
291
    ctions:
                                                                    ctions:
292
                         ignored functions += (
                                                               356
                                                                                         ignored functions += (
                                                                                             primary_good_functions + se
                             primary_good_functions + se
293
                                                               357
    condary_good_functions
                                                                    condary_good_functions
294
                                                               358
                                                                                         )
                     else:
                                                               359
                                                                                    else:
295
                         if not (
296
                                                               360
                                                                                         if not (
                             function name in primary go
                                                                                             function_name in primary_go
    od functions
                                                                    od functions
                             or function name in seconda
                                                               362
                                                                                             or function name in seconda
    ry_good_functions
                                                                    ry_good_functions
299
                         ):
                                                               363
                                                                                         ):
300
                              logger.warning(
                                                               364
                                                                                             logger.warning(
301
                                  f"Function {function_na
                                                               365
                                                                                                 f"Function {function_na
    me} not found in primary or secondary good function
                                                                    me} not found in primary or secondary good function
    s or bad functions for testcase {str(testcase)}!"
                                                                    s or bad functions for testcase {str(testcase)}!"
                                                               366
303
                             logger.warning(
                                                               367
                                                                                             logger.warning(
304
                                  f"Candidate functions:
                                                               368
                                                                                                 f"Candidate functions:
      {primary_good_functions}+{secondary_good_function
                                                                     \{primary\_good\_functions\} + \{secondary\_good\_function\}
                                                                    s}+{primary_bad_functions}"
    s}+{primary_bad_functions}"
305
                             )
                                                               369
                                                                                             )
306
                         ignored_functions += primary_ba
                                                               370
                                                                                         ignored_functions += primary_ba
    d functions
                                                                    d functions
307
                                                               371
308
                     # Build list of candidate functions
                                                               372
                                                                                     # Build list of candidate functions
                     candidate_functions = {function_nam
                                                               373
                                                                                     candidate_functions = {function_nam
309
    e?
                                                                    e }
310
                     while True:
                                                               374
                                                                                     while True:
                         new_candidate_functions = set(c
                                                               375
                                                                                         new_candidate_functions = set(c
311
    andidate functions)
                                                                    andidate functions)
312
                         for candidate function in candi
                                                               376
                                                                                         for candidate function in candi
    date_functions:
                                                                    date_functions:
313
                             for dependency in text_sect
                                                               377
                                                                                             for dependency in text_sect
    ion_functions[candidate_function][
                                                                    ion_functions[candidate_function][
314
                                  "Dependencies"
                                                               378
                                                                                                 "Dependencies"
315
                                                               379
                                                                                             1:
316
                                  if dependency in text s
                                                               380
                                                                                                 if dependency in text s
    ection_functions.keys():
                                                                    ection_functions.keys():
                                      new_candidate_funct
                                                               381
                                                                                                     new_candidate_funct
317
    ions.add(dependency)
                                                                    ions.add(dependency)
318
                         if len(new_candidate_functions)
                                                               382
                                                                                         if len(new_candidate_functions)
    == len(candidate_functions):
                                                                    == len(candidate_functions):
319
                                                               383
                         else:
320
                                                               384
                                                                                         else:
                             candidate functions = new c
                                                               385
                                                                                             candidate functions = new c
321
    andidate_functions
                                                                    andidate_functions
                                                               386
322
323
                     context_functions = [
                                                               387
                                                                                     context_functions = [
                         fnn for fnn in candidate functi
                                                               388
                                                                                         fnn for fnn in candidate functi
324
    ons if fnn not in ignored_functions
                                                                    ons if fnn not in ignored_functions
                                                               389
                                                                                     ]
                     ]
```

```
326
                    context = "\n".join(
                                                             390
                                                                                  context = "\n".join(
327
                                                             391
328
                             text_section_functions[fnn]
                                                             392
                                                                                          text_section_functions[fnn]
    ["TextRepresentation"]
                                                                  ["TextRepresentation"]
329
                             for fnn in context_function
                                                             393
                                                                                          for fnn in context_function
330
                                                             394
                        ]
                                                                                      ]
331
                    )
                                                             395
                                                                                  )
332
                                                             396
                    example["Example"] += "\n" + contex
                                                                                  example["Example"] += "\n" + contex
333
                                                             397
334
                elif context_granularity is None:
                                                             398
                                                                              elif context_granularity is None:
                                                             399
335
                    pass
                                                                                  pass
336
                else:
                                                             400
                                                                              else:
                    raise ValueError("Granularity not s
                                                                                  raise ValueError("Granularity not s
337
    upported for context")
                                                                  upported for context")
338
                                                             402
339
                examples += [example]
                                                             403
                                                                              examples += [example]
340
                                                             404
                                                             405
341
        else:
                                                                      else:
            raise ValueError("Granularity not supporte
                                                                          raise ValueError("Granularity not supporte
342
                                                             406
    d")
                                                                  d")
343
                                                             407
                                                             408
344
        return examples
                                                                      return examples
345
                                                             409
                                                             410
347 def label_example(example, label_strategy: LabelStr
                                                             411 def label_example(example, label_strategy: LabelStr
        """Compute a label for a given example dependin
                                                                     """Compute a label for a given example dependin
348
                                                             412
    g on the label strategy."""
                                                                  g on the label strategy."""
       if label_strategy == LabelStrategy.BINARYCLASSI
                                                                    if label_strategy == LabelStrategy.BINARYCLASSI
349
                                                             413
    FICATION:
                                                                  FICATION:
350
            return example["GoodOrBad"]
                                                             414
                                                                          return example["GoodOrBad"]
        elif label_strategy == LabelStrategy.MULTICLASS
                                                                      elif label_strategy == LabelStrategy.MULTICLASS
    CLASSIFICATION:
                                                                  CLASSIFICATION:
            if example["GoodOrBad"] == "Good":
                                                                          if example["GoodOrBad"] == "Good":
352
                                                             416
353
                return "NoWeakness"
                                                             417
                                                                              return "NoWeakness"
354
                                                             418
                return f"CWE{example['Testcase']['Weakn
                                                                              return f"CWE{example['Testcase']['Weakn
355
                                                             419
    ess']['WeaknessID']}"
                                                                  ess']['WeaknessID']}"
                                                             420
356
357
                                                             421
                                                                      else:
            raise ValueError("Label strategy not suppor
                                                                         raise ValueError("Label strategy not suppor
358
                                                             422
    ted")
                                                                  ted")
                                                             423
359
360
                                                             424
      _all__ = ["extract_labeled_examples", "validate_te
                                                                   _all__ = ["extract_labeled_examples", "validate_te
                                                             425
    stcase_files", "label_example"]
                                                                  stcase_files", "label_example"]
                                                             426
362
```