positive hack days12

# Spotting Bugs in PyTorch with Continuous Fuzzing

Theodor Arsenij Larionov-Trichkine
Daniil Kuts

# / whoami

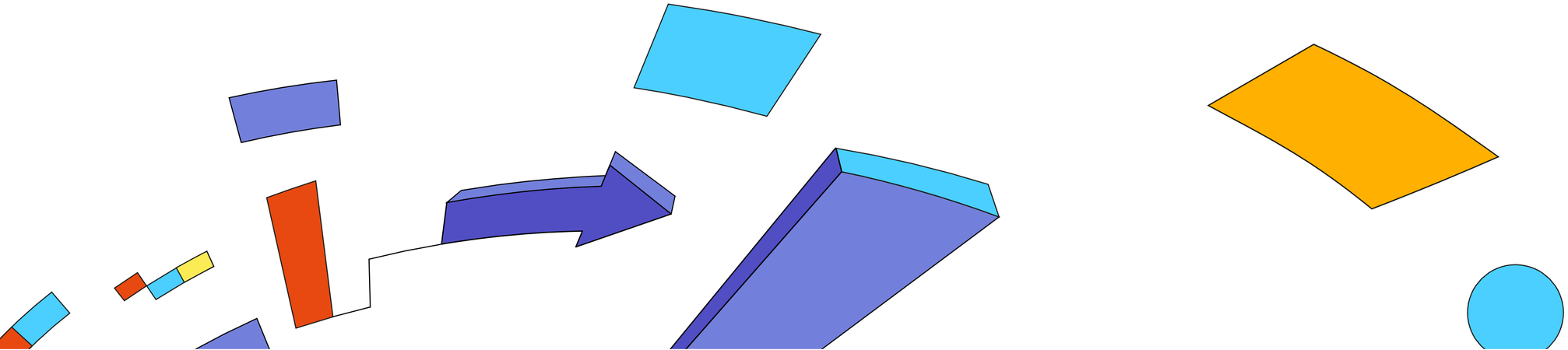
## Theodor Arsenij Larionov-Trichkine

ex. Kaspersky, Postgres Pro, ISP RAS

Reverse Engineer, Fuzzing Enthusiast, Software Engineer, Vulnerability Researcher, and CTF player

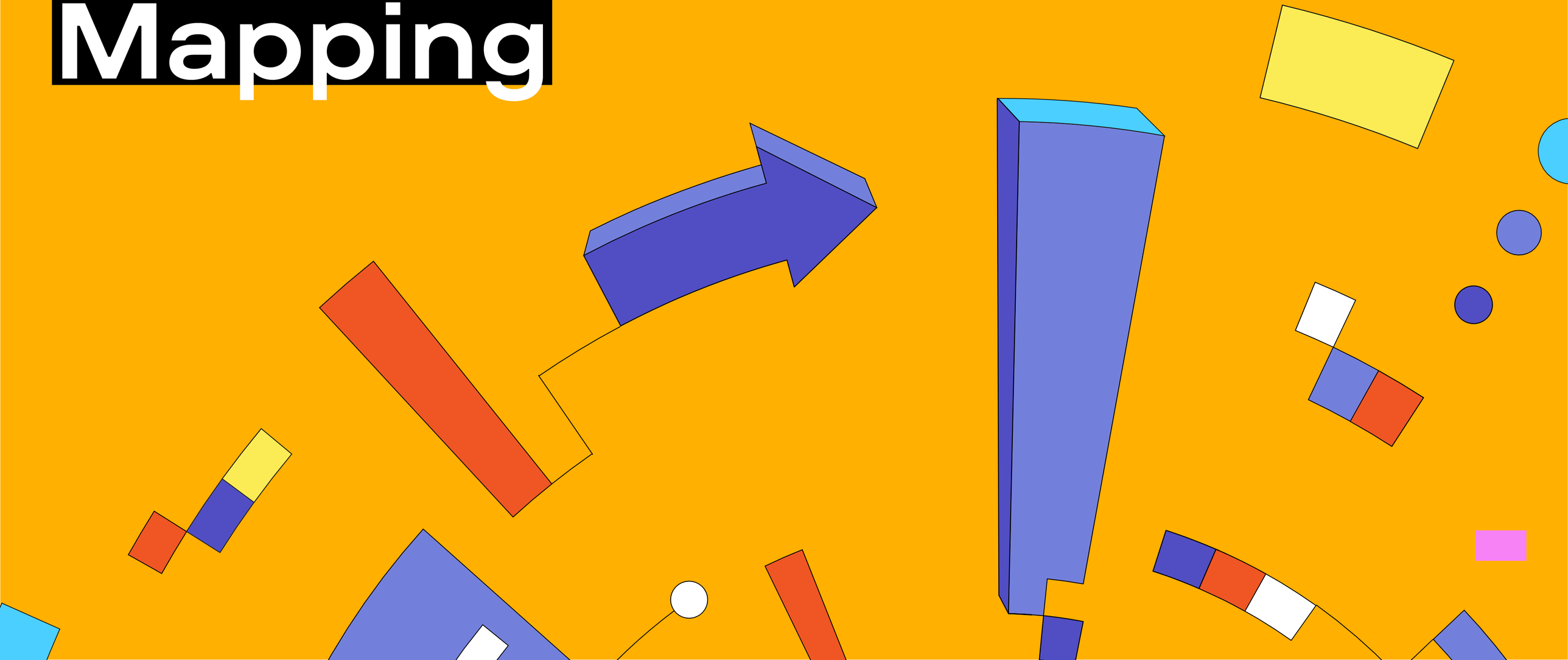
# / Motivation

1. PyTorch serves as the fundamental building block for various ML/DL products.
2. PyTorch is developed at an incredible rate. As a result, multiple bugs are being shipped to customers.
3. PyTorch doesn't have fuzzing support out-of-the-box.

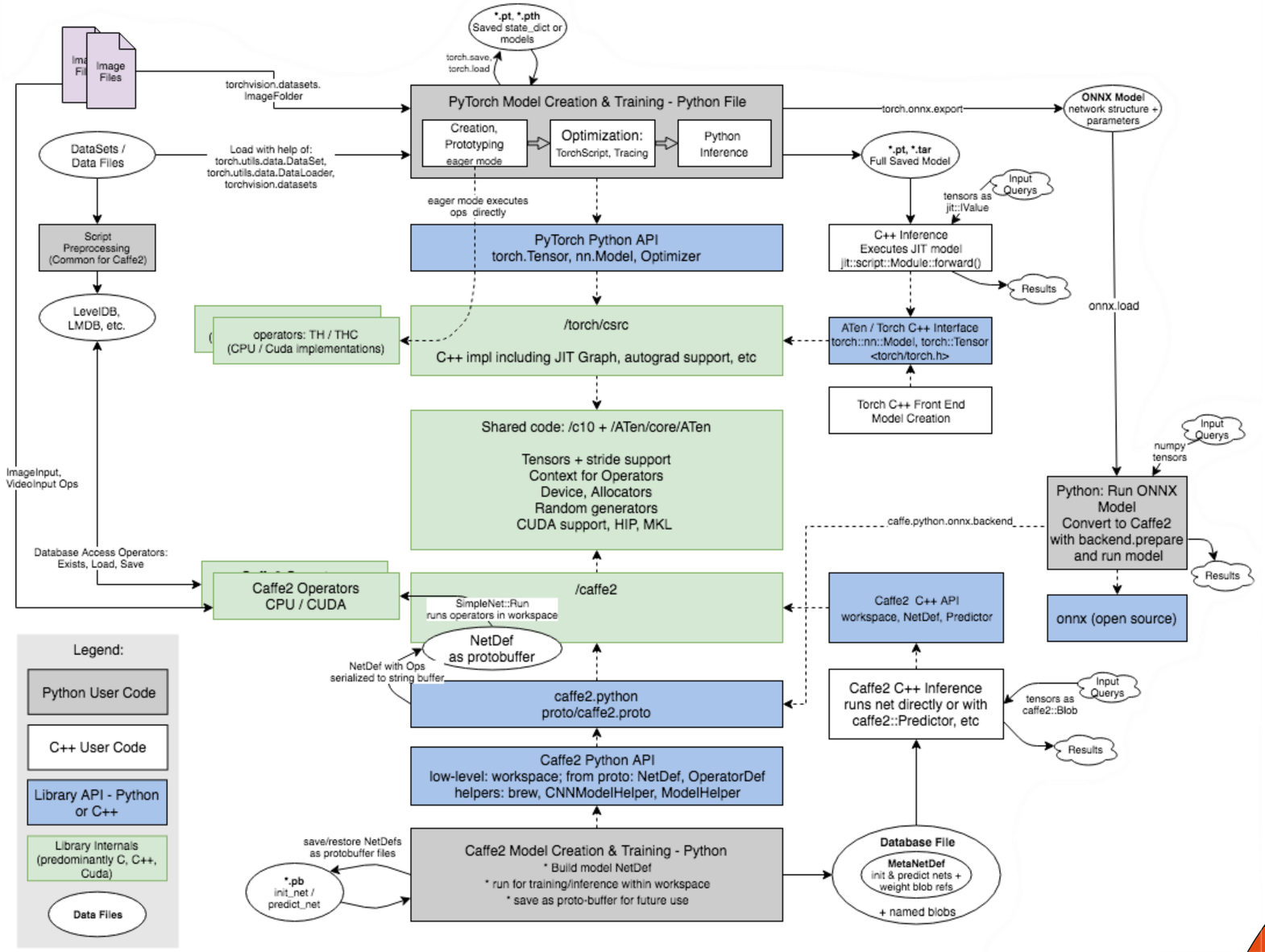


# Attack Surface Mapping

positive hack days12



# / PyTorh's Architecture




[PyTorch Data Flow and Interface Diagram]

# / Finding Interesting Targets

phd 12

## Manual Analysis

+ Allows you to find the most interesting targets

- Time-consuming

# / Finding Interesting Targets

phd 12

## Automatic Analysis

+ Allows you to find good entry points
+ Scalable
- Doesn't "understand" the code


weggli

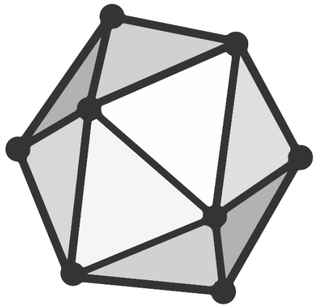
# / CodeQL Query

phd 12

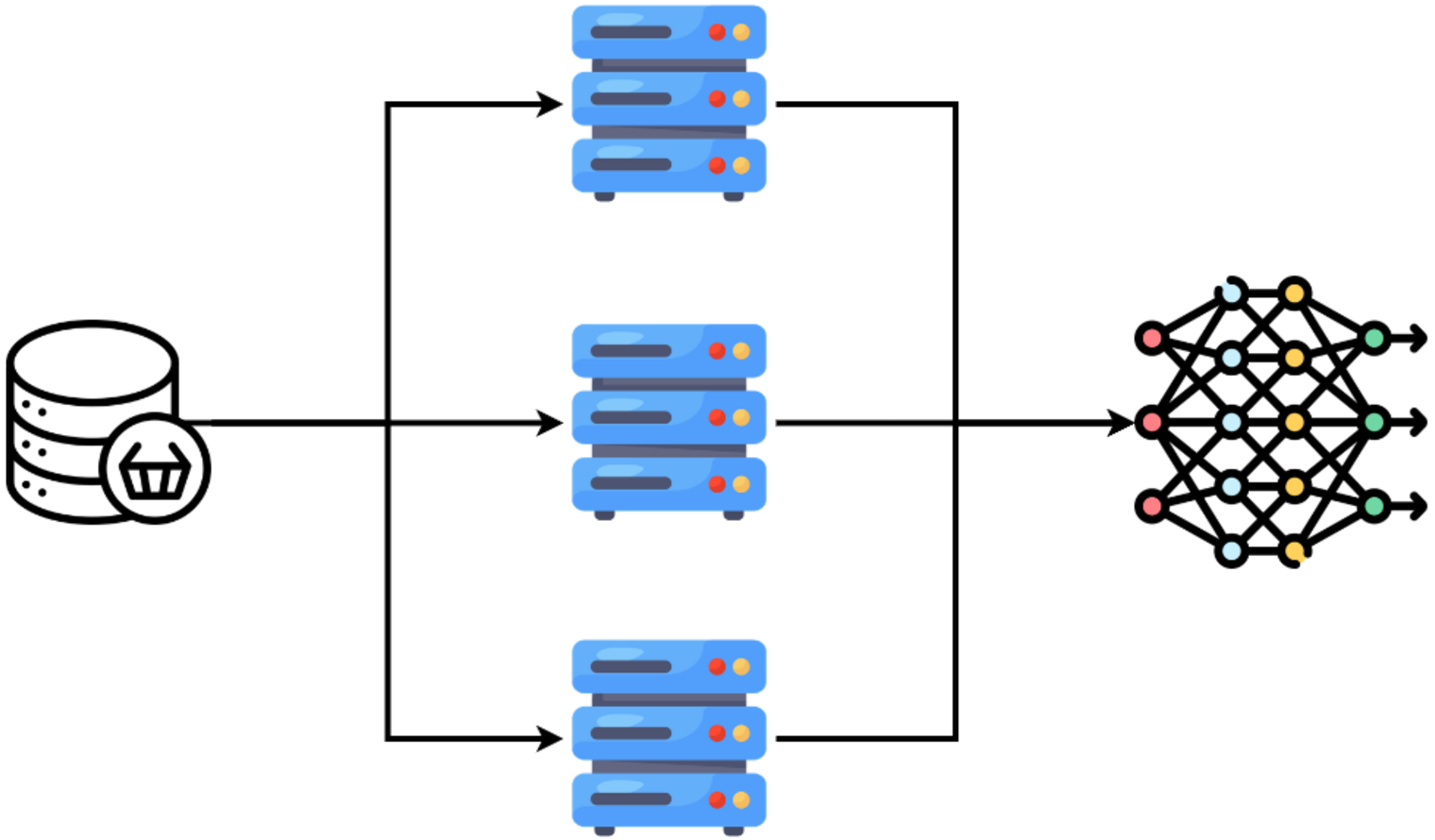
```
predicate libfuzzerFuzzable(MetricFunction mf) {
    mf.getNumberOfParameters() = 2 and
    mf.getAParameter().getUnderlyingType() instanceof IntegralType and
    exists(PointerType ptr |
        ptr = mf.getAParameter().getUnderlyingType() and
        ptr.getUnderlyingType().getName().regexpMatch(".*(char|int|byte|void)+.*")
    )
}

from MetricFunction mf, int cc
where
    cc = getComplexity(mf) and
    goodFile(mf) and
    goodName(mf) and
    libfuzzerFuzzable(mf)
select "Complexity: ", cc,
       "Function: ", mf,
       "Declaration", mf.getFullSignature(),
       "File: ", mf.getFile() order by cc desc
```

# / Model Loading

```
model = TheModelClass()
torch.save(model, 'model.pt')

model = torch.load('model.pt')
model.eval()
```



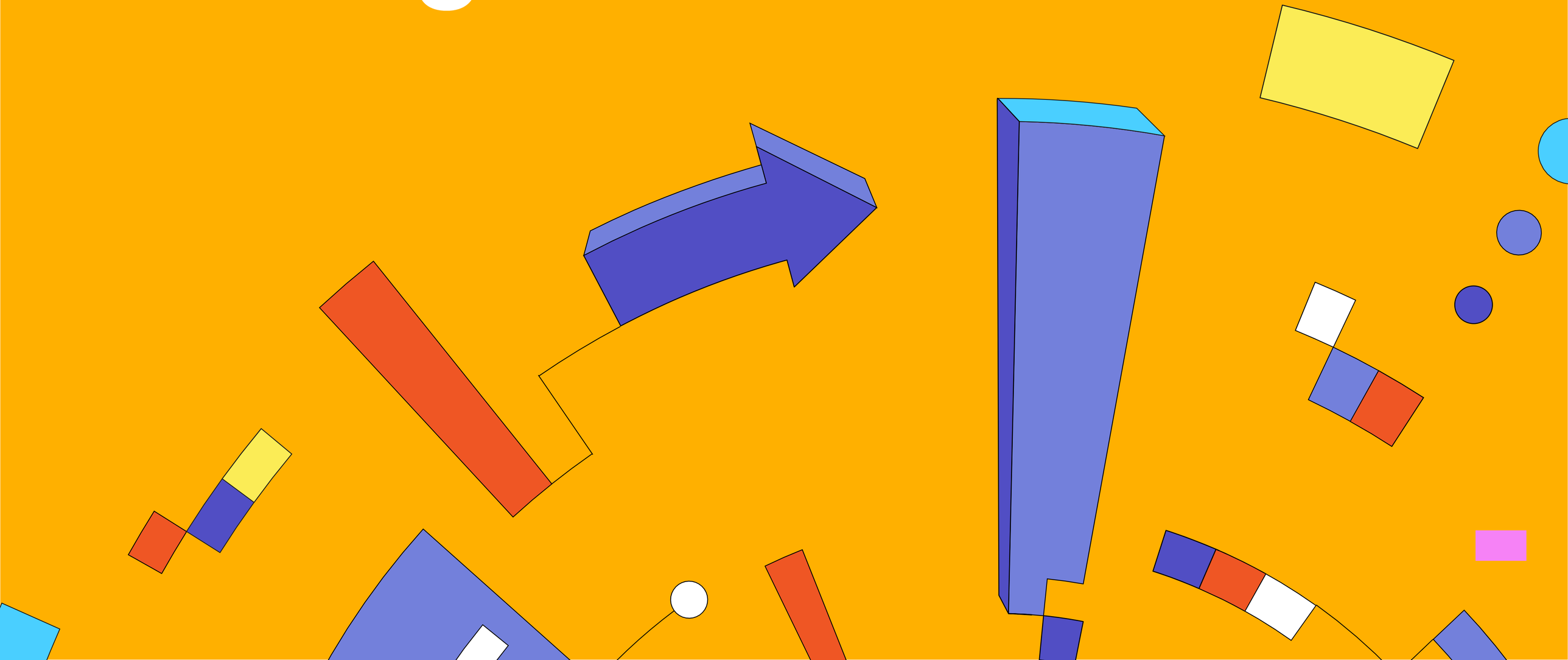
# / Distributed Learning (RPC)



Distribute Data  Compute Gradients  Update Model

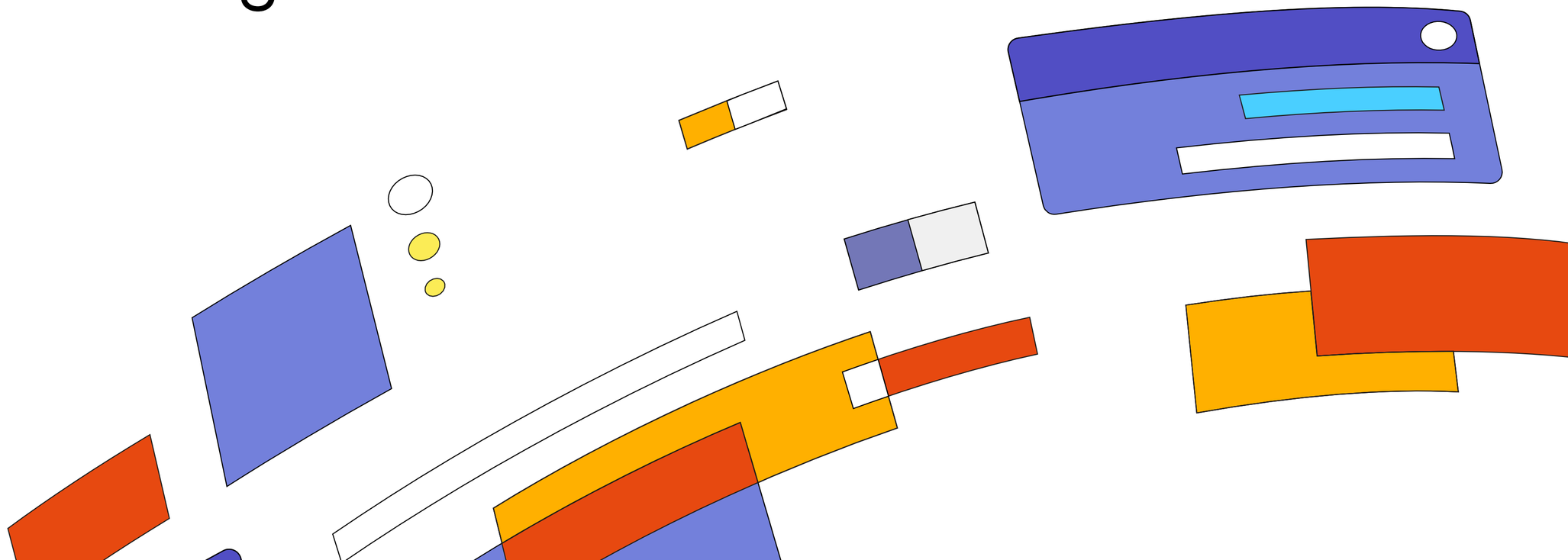
# Preparing for Fuzzing

positive hack days12



# / Preparing for Fuzzing

1. Fuzzing Harness Development
2. Corpus Collection
3. Docker Image Creation



# / Fuzzing Harness Development

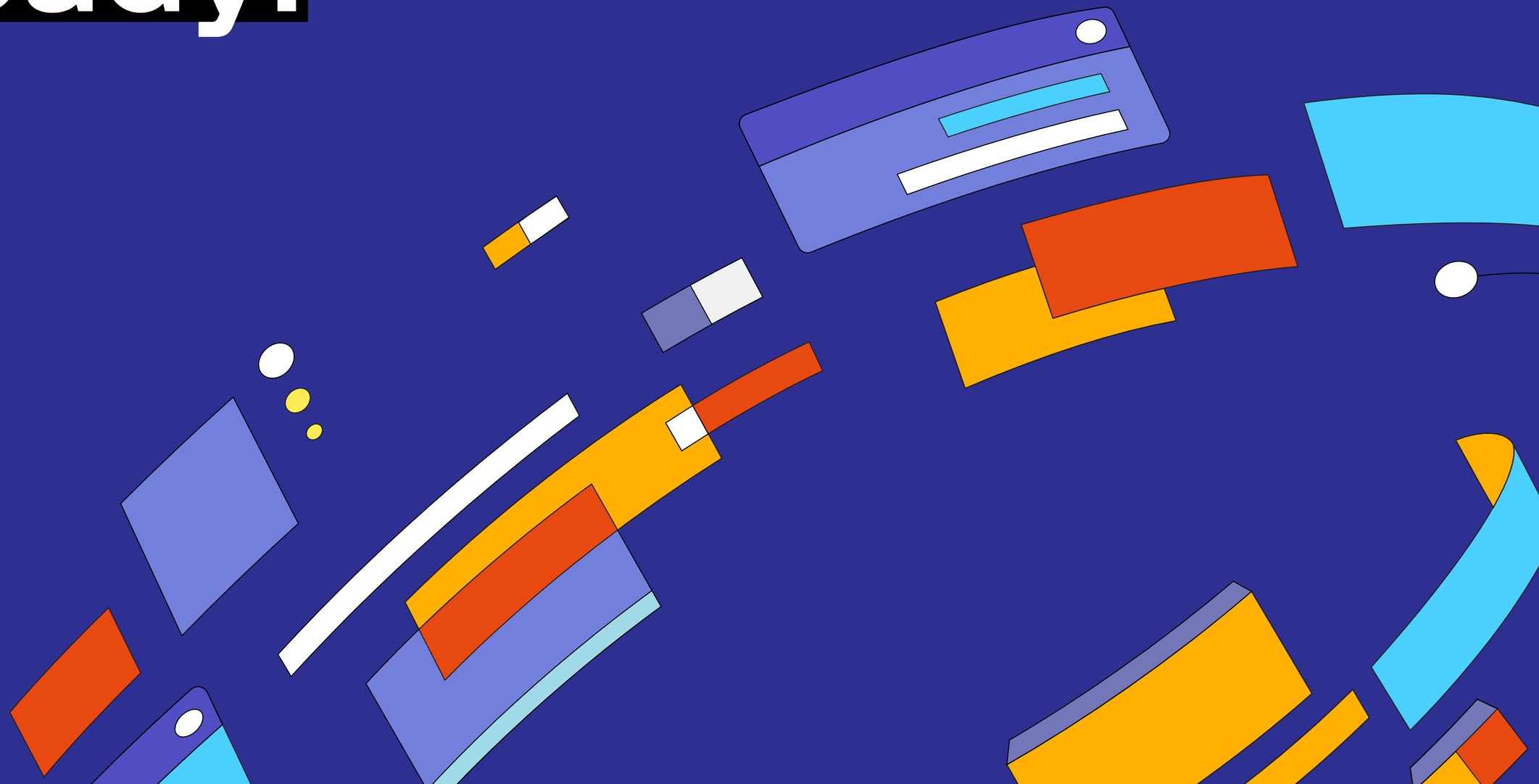
phd 12

```cpp
extern "C" int LLVMFuzzerTestOneInput(const uint8_t *data, size_t size) {
  std::stringstream ss;
  std::copy((char *)data, (char *)data + size,
            std::ostreambuf_iterator<char>(ss));

  try {
    auto m = torch::jit::load(ss);
  } catch (const c10::Error &e) {
    return 0;
  } catch (const torch::jit::ErrorReport &e) {
    return 0;
  } catch (const std::runtime_error &e) {
    return 0;
  }

  return 0;
}
```

load_fuzz.cc: github.com/ispras/oss-sydr-fuzz

# Let's Fuzz it Already!

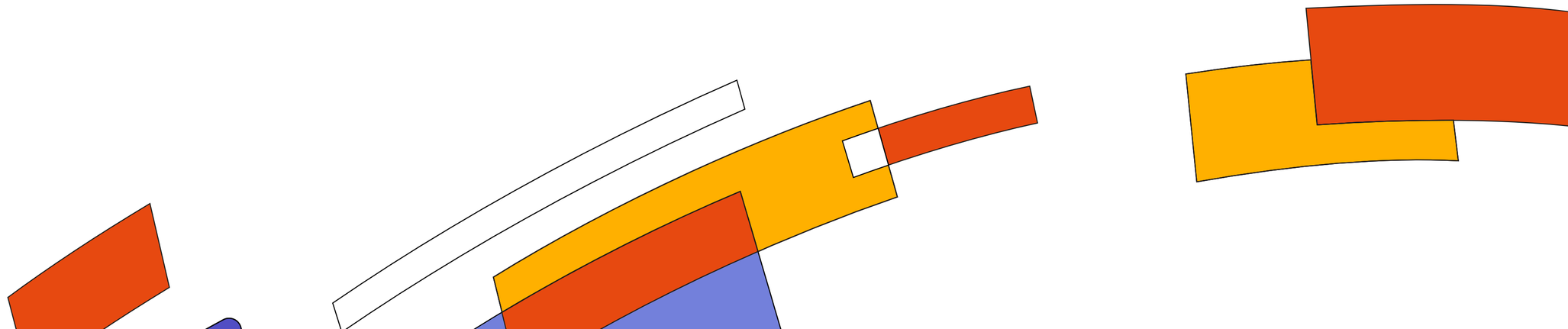
positive hack days 12



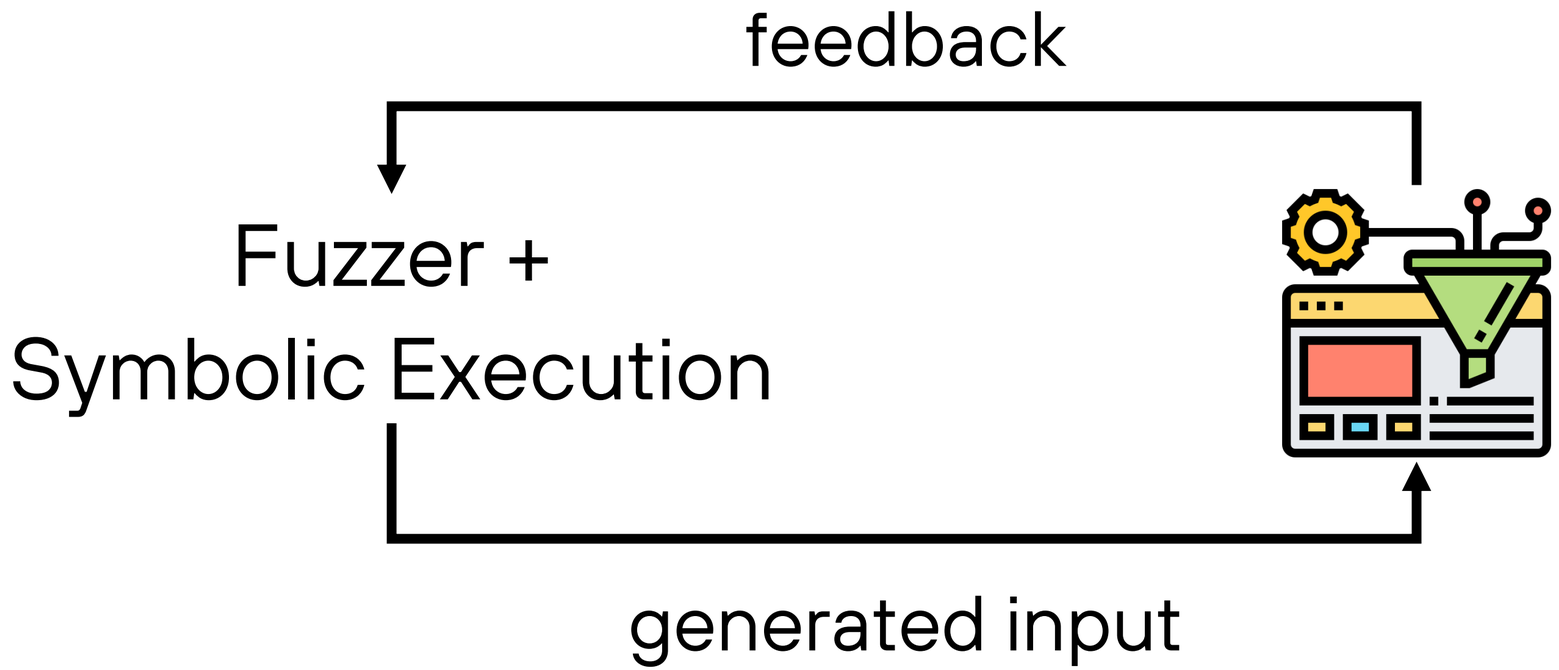
# / Dynamic Analysis Pipeline

phd 12

1. Hybrid Fuzzing
2. Fuzzing Corpus Minimization
3. Targeted Errors Search
4. Crashes Triaging
5. Fuzzing Campaign Assessment



# / Hybrid Fuzzing




afl++/sydr
libfuzzer/sydr
afl++/symqemu
afl++/symcc
afl++/fuzzolic

# / Fuzzing Corpus Minimization

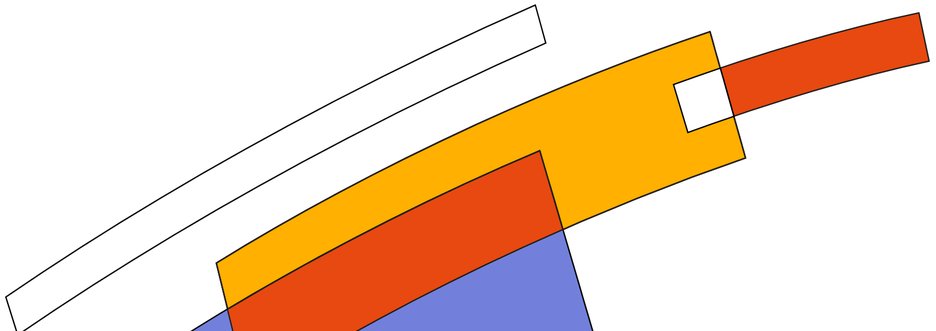
**afl-cmin**

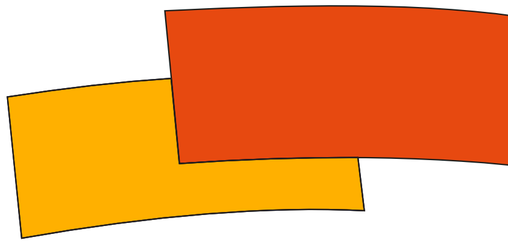
**libfuzzer –merge=1**

# / Targeted Errors Search

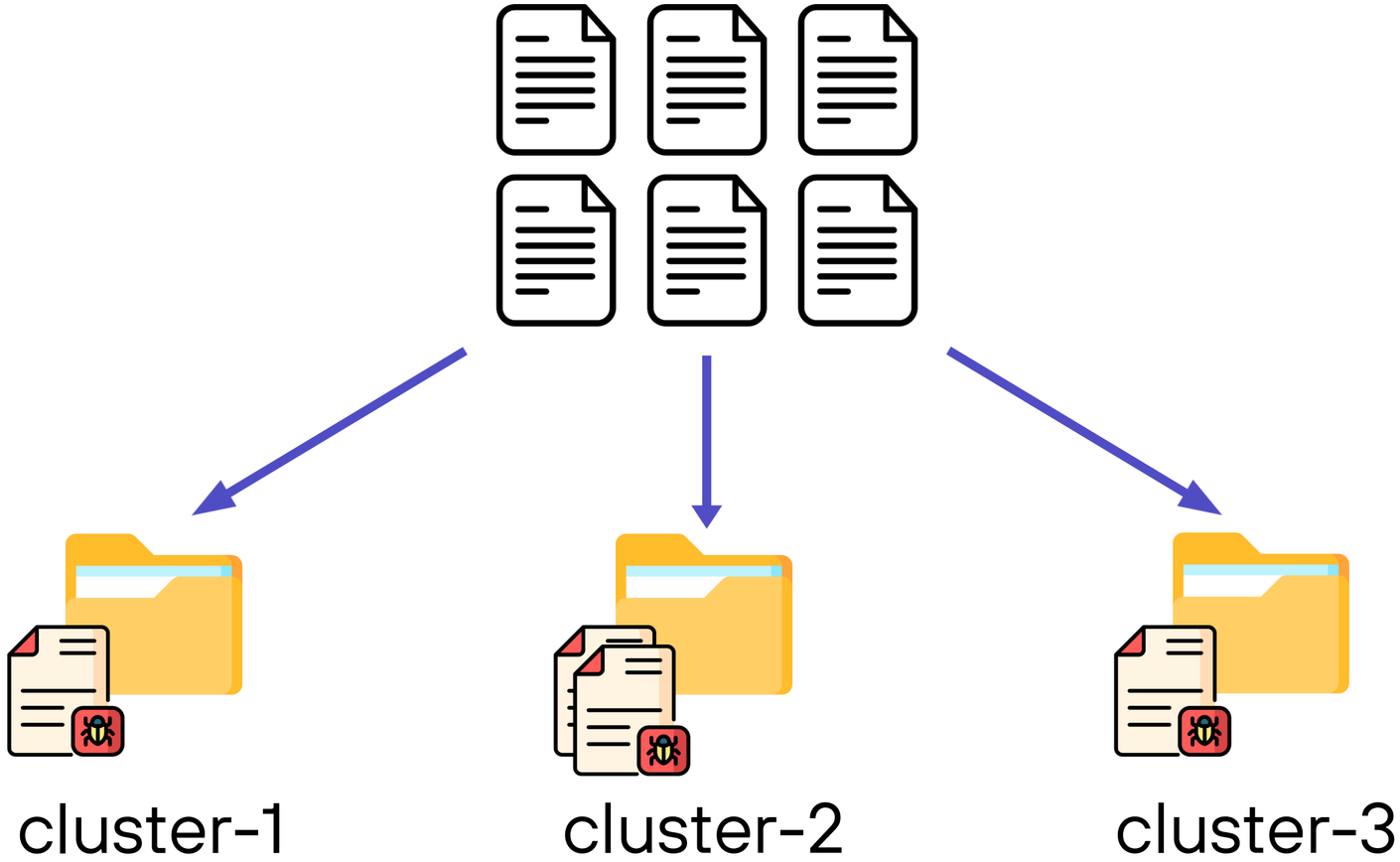
phd 12

```
char data[64];
uint index = read_int();
if (index < 74)
    data[index] = 0x37;
```





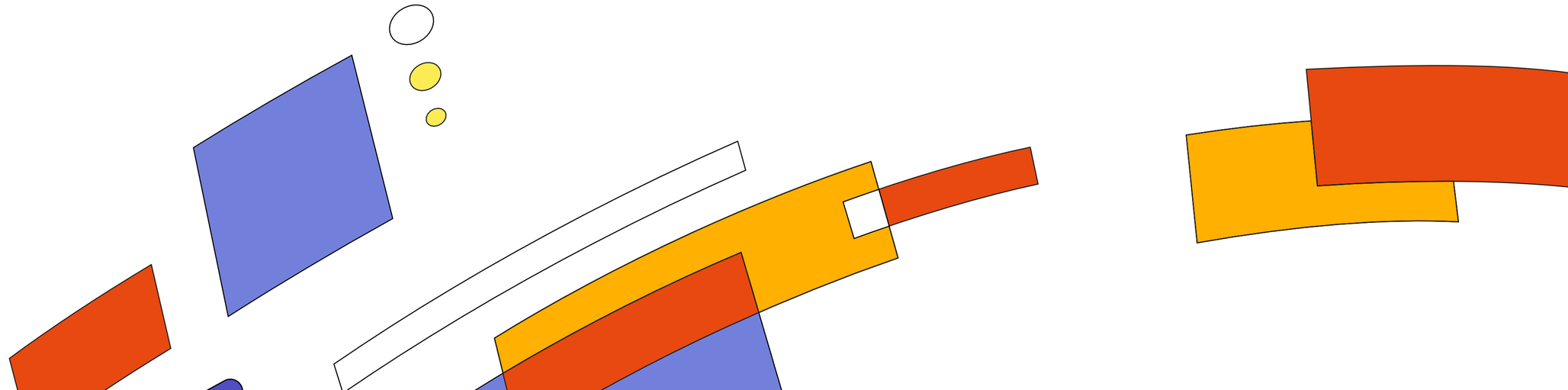
# / Crashes Triaging




https://github.com/ispras/casr

# / Fuzz Campaign Assessment

1. **Code Coverage Assessment**
2. **Bug Reports, Fixes**
3. **Fuzz-loop new iteration?**



positive hack days 12

# Bugs Found Case Studies

# / RPC OOB access

```
 std::unique_ptr<ScriptCall> ScriptCall::fromIValues(
     std::vector<at::IValue>& ivalues) {
+  TORCH_INTERNAL_ASSERT(
+      ivalues.size() > 1,
+      "At least 2 IValues are required to build a ScriptCall.");
+
   const std::string& qualifiedName = ivalues.back().toStringRef();
```

Pull-request #94297

# / Unpickler OOB access

```
      case PickleOpCode::LONG_BINGET: {
-       stack_.push_back(memo_table_.at(read<uint32_t>()));
+       auto pos = read<uint32_t>();
+       TORCH_CHECK(
+           memo_table_.size() > pos,
+           "Parsing error: out of bounds access at ",
+           (size_t)pos,
+           " to memo_table_ which is of size ",
+           memo_table_.size());
+       stack_.push_back(memo_table_.at(pos));
      } break;
      <...SNIP...>
      case PickleOpCode::BINPERSID: {
+       TORCH_CHECK(!stack_.empty(), "Parsing error: stack_ is empty");
        auto tuple = pop(stack_).toTuple();
        const auto& args = tuple->elements();
```

Pull-request #91401

positive hack days12

# Conclusion

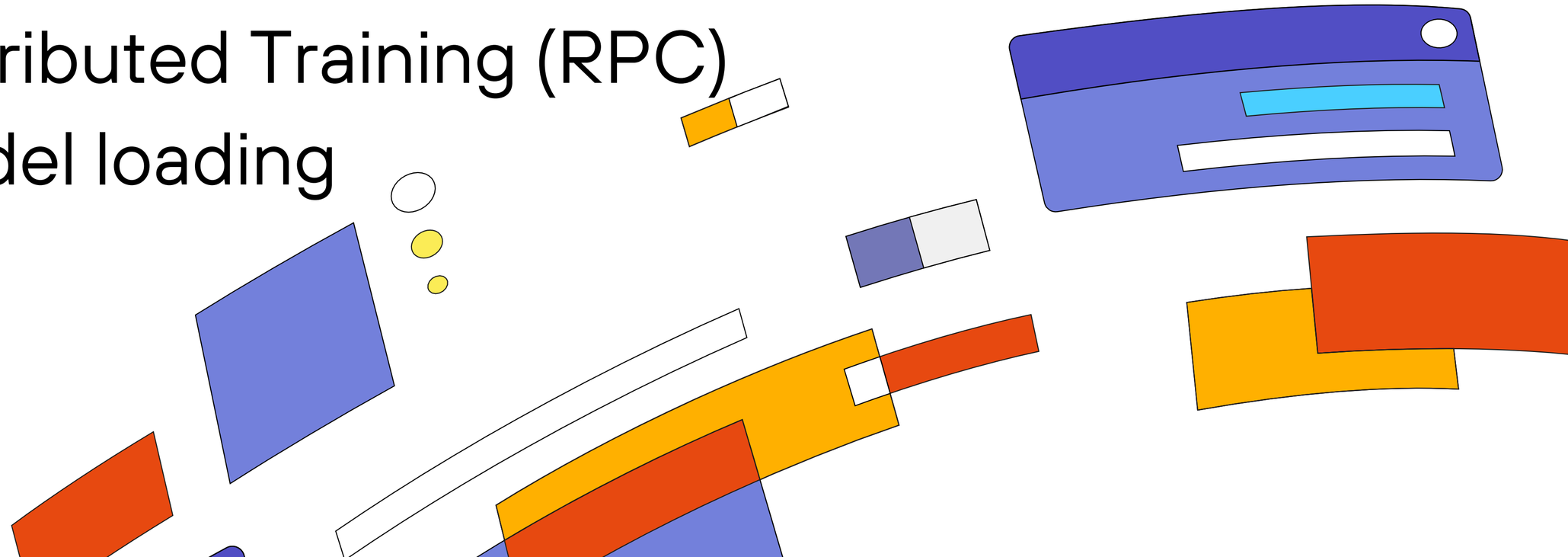
# / Total Bugs Found

phd 12

## 9 Pull Requests >>

Numerous bug-fixes in various modules, including:

1. Distributed Training (RPC)
2. Model loading



positive
hack
days12

# Thanks!

m4drat

RE pwn.report