REbus

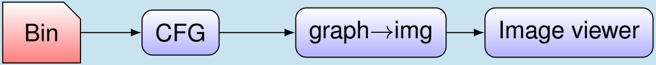
Make your security tools cooperate

Raphaël Rigo with slides by Xavier Mehrenberger

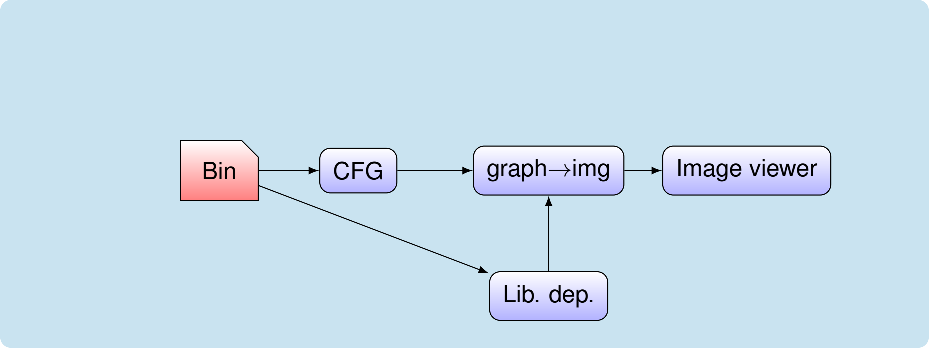
July 5th / RMLL Sec 2016

AIRBUS
GROUP

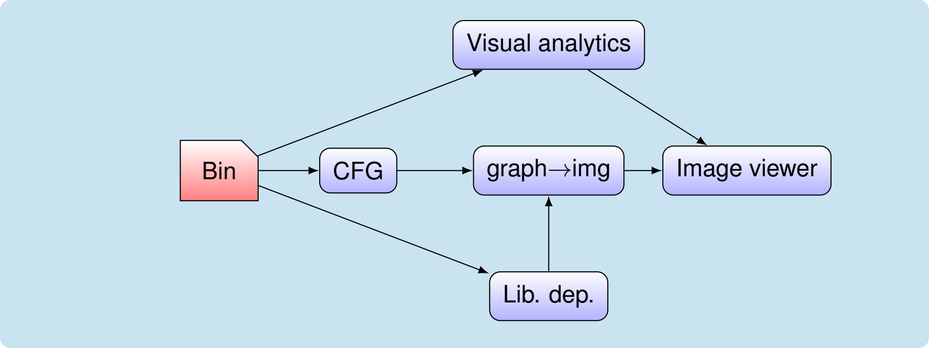
Example malware CFG analysis workflow



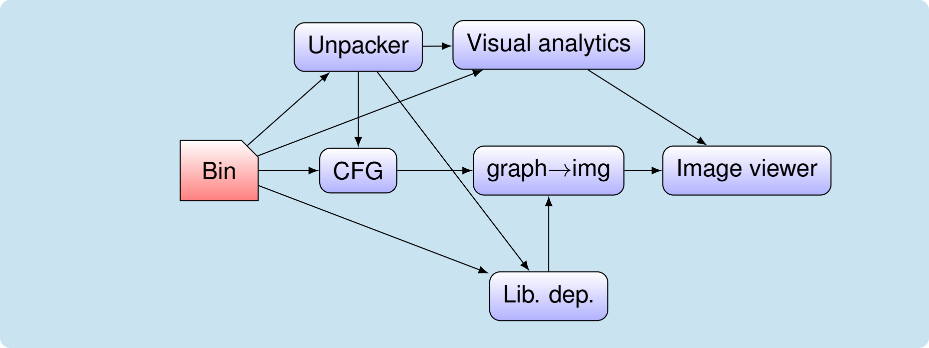
Example malware CFG analysis workflow



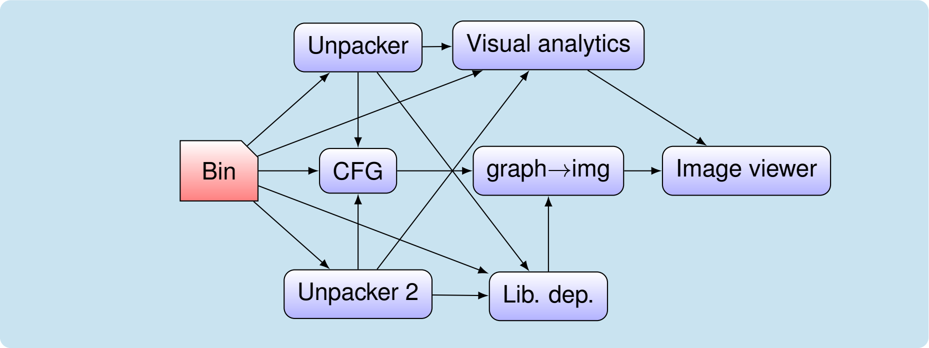
Example malware CFG analysis workflow



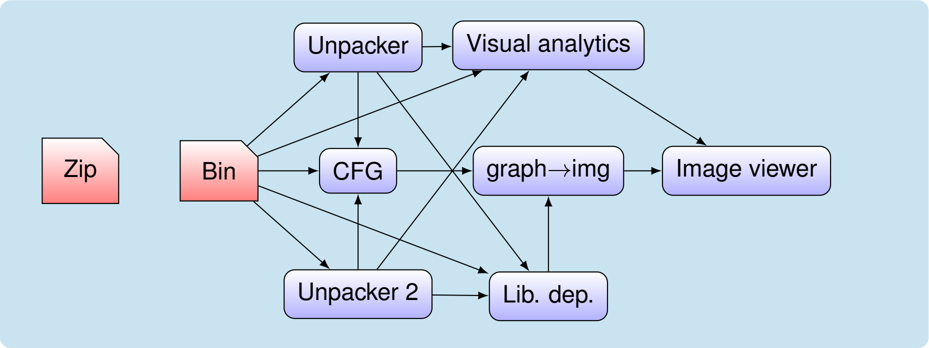
Example malware CFG analysis workflow



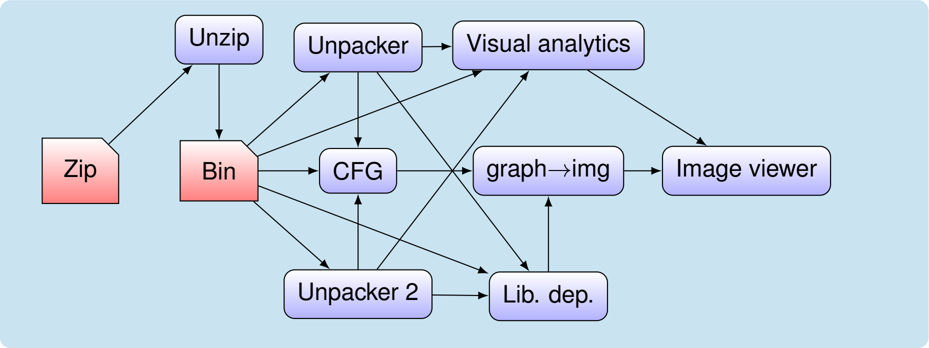
Example malware CFG analysis workflow



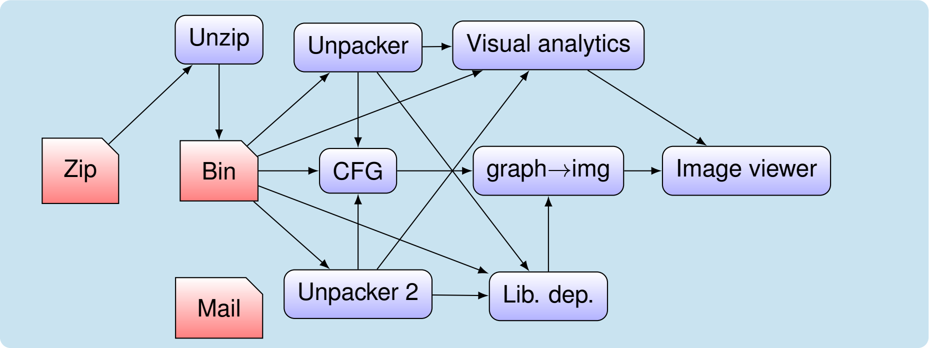
Example malware CFG analysis workflow



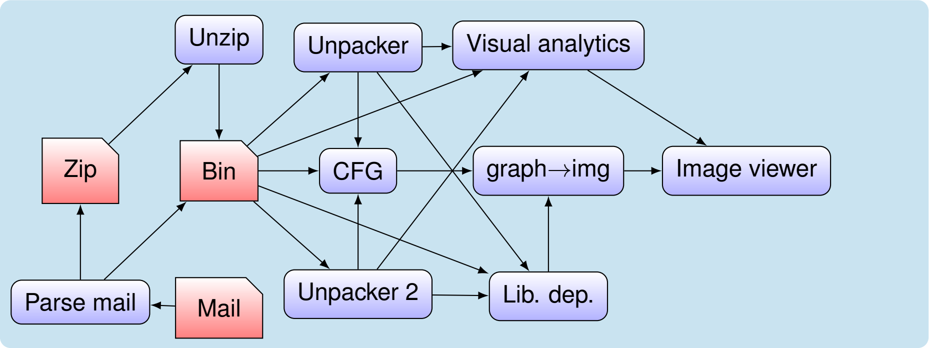
Example malware CFG analysis workflow



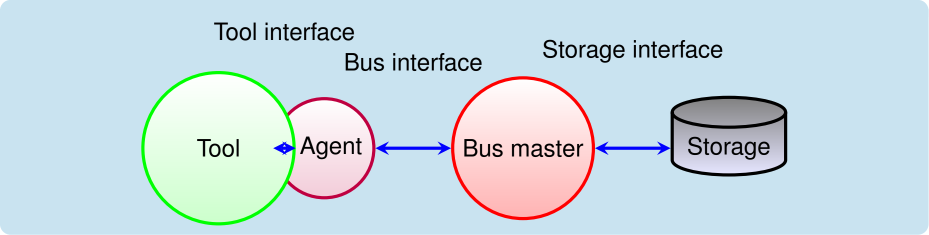
Example malware CFG analysis workflow



Example malware CFG analysis workflow



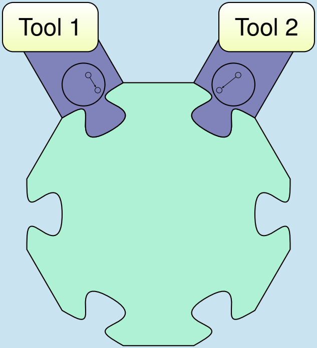
REbus interfaces



REbus architecture

Framework, with a decentralised workflow

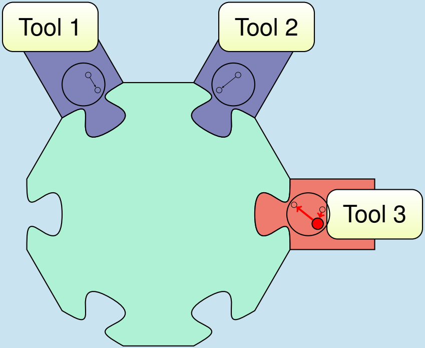
Decentralized workflow



REbus architecture

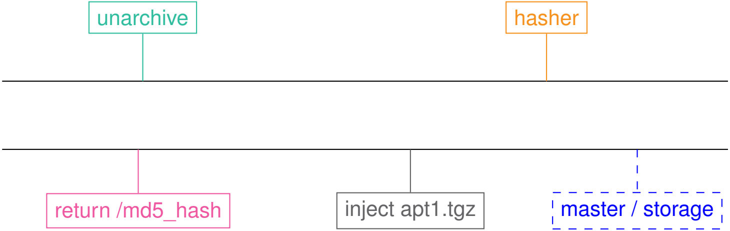
Framework, with a decentralised workflow

Adding a new agent



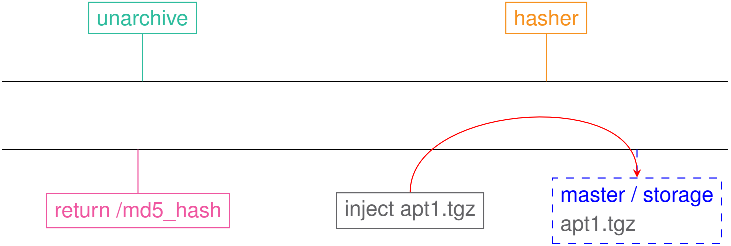
Data exchange across the bus

Goal: compute md5sum of each file contained in provided `.tgz` archive



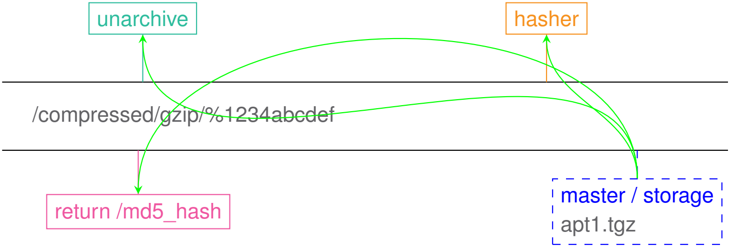
Data exchange across the bus

Goal: compute md5sum of each file contained in provided `.tgz` archive



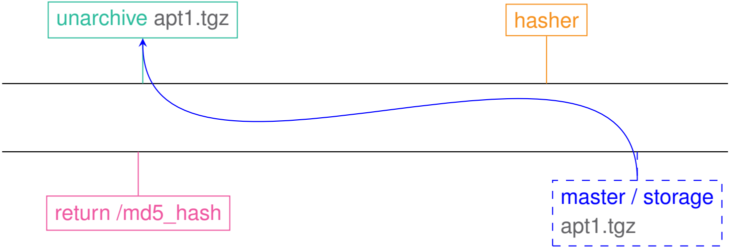
Data exchange across the bus

Goal: compute md5sum of each file contained in provided `.tgz` archive



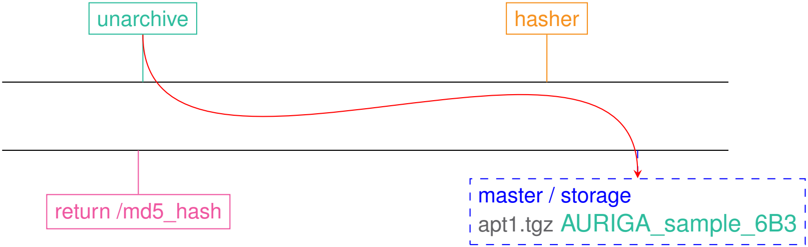
Data exchange across the bus

Goal: compute md5sum of each file contained in provided `.tgz` archive



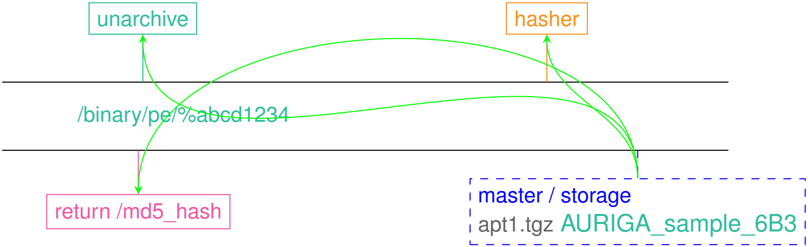
Data exchange across the bus

Goal: compute md5sum of each file contained in provided `.tgz` archive



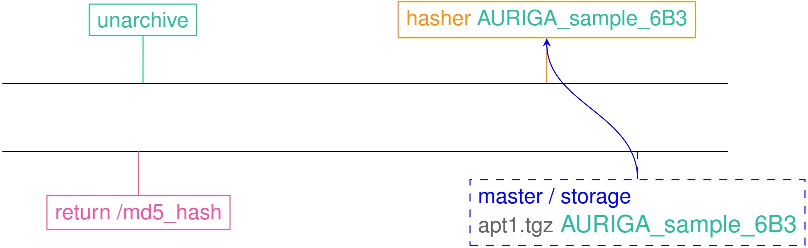
Data exchange across the bus

Goal: compute md5sum of each file contained in provided `.tgz` archive



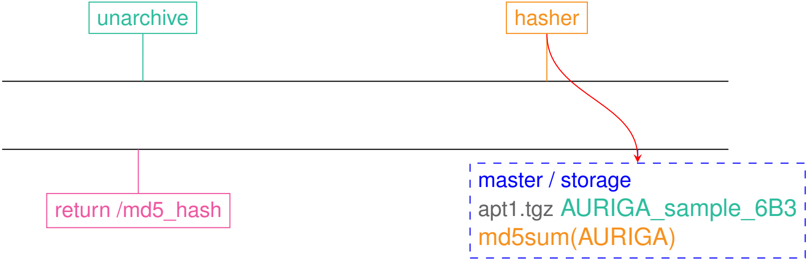
Data exchange across the bus

Goal: compute md5sum of each file contained in provided `.tgz` archive



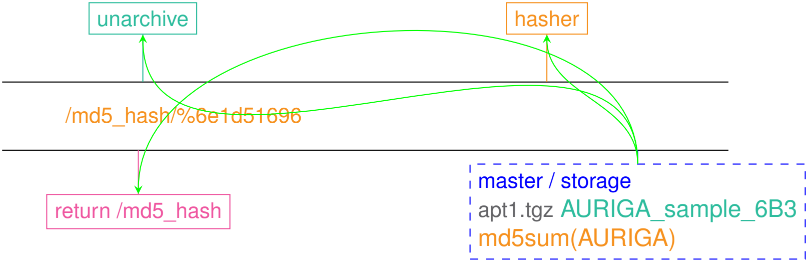
Data exchange across the bus

Goal: compute md5sum of each file contained in provided `.tgz` archive



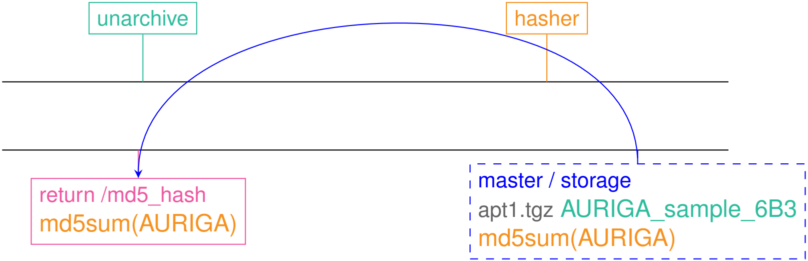
Data exchange across the bus

Goal: compute md5sum of each file contained in provided `.tgz` archive



Data exchange across the bus

Goal: compute md5sum of each file contained in provided `.tgz` archive



Example agent combination

```
$ rebus_agent -m rebus_demo.agents hasher unarchive \
                                   inject ~/apt1.tgz -- \
                                   return --short md5_hash

apt1.tgz:AURIGA_6B31344B40E2AF9C9EE3BA707558C14E =
    6b31344b40e2af9c9ee3ba707558c14e
apt1.tgz:AURIGA_CDCD3A09EE99CFF9A58EFEA5CCBE2BED =
    cdcd3a09ee99cff9a58efea5ccbe2bed
apt1.tgz:BANGAT_468FF2C12CFFC7E5B2FE0EE6BB3B239E =
    468ff2c12cffc7e5b2fe0ee6bb3b239e
[...]
```

```python
from rebus.agent import Agent
from rebus_demo.tools import hash_tools


@Agent.register
class Hasher(Agent):
    _name_ = "hasher"
    _desc_ = "Return md5 of a binary"

    def selector_filter(self, selector):
        # Indicate that this agent is only interested in descriptors whose
        # selector start with "/binary"
        return selector.startswith("/binary/")

    def process(self, desc, sender_id):

        # call the very complex tool on the received value
        md5_hash = hash_tools.md5hasher(desc.value)

        # Create a new child descriptor
        new_desc = desc.spawn_descriptor("/md5_hash", unicode(md5_hash),
                                         self.name)

        # Push the new descriptor to the bus
        self.push(new_desc)
```

Listing 1: Agent REbus to compute md5sum of binary files

Try REbus

- BSD licence
- Download & docs at `https://bitbucket.org/iwseclabs/rebus`
- Demo agents at `https://bitbucket.org/iwseclabs/rebus_demo`