GRANEF: Utilization of a Graph Database for Network Forensics

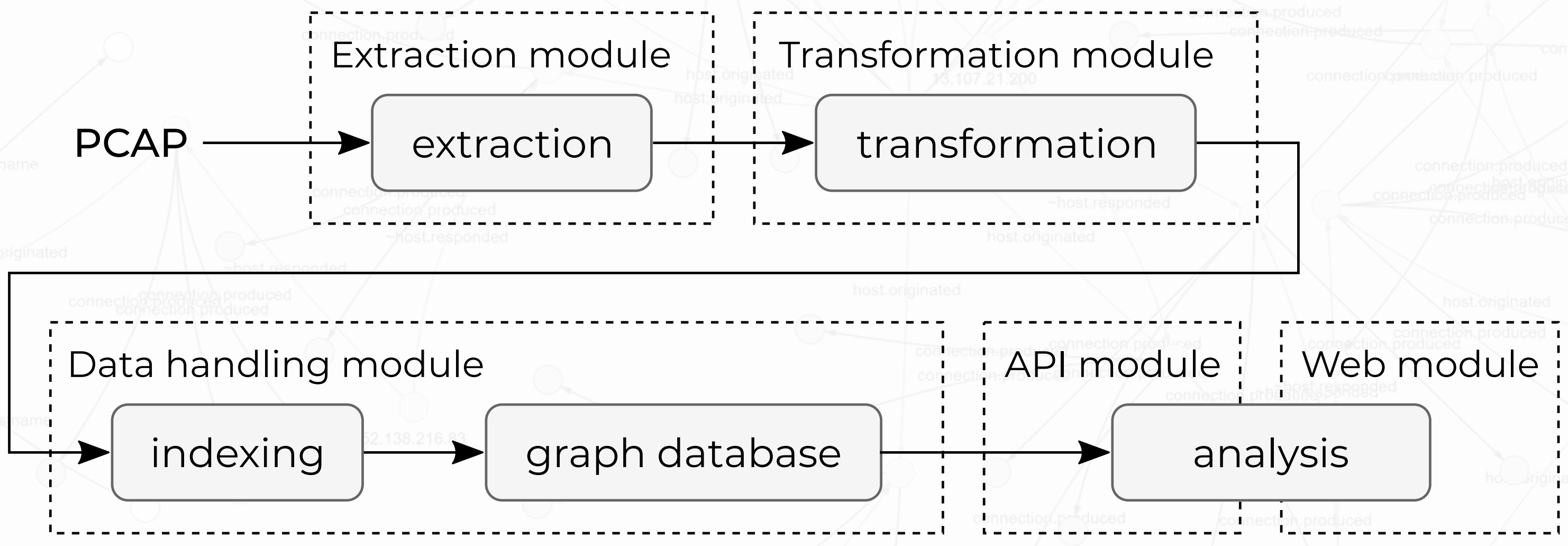
Milan Cermak and Denisa Sramkova

Institute of Computer Science, Masaryk University, Brno, Czech Republic
cermak@ics.muni.cz, denisa.sramkova@mail.muni.cz

Abstract

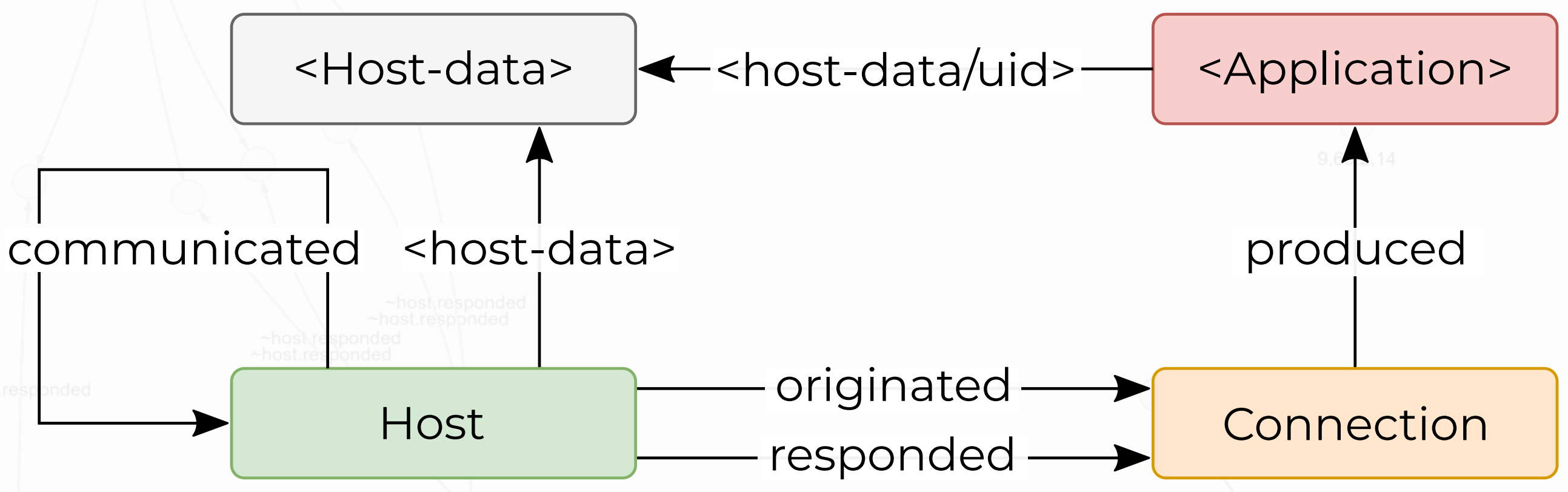
Understanding the information in captured network traffic, extracting the necessary data, and performing incident investigations are principal tasks of network forensics. The analysis of such data is typically performed by tools allowing manual browsing, filtering, and aggregation or tools based on statistical analyses and visualizations facilitating data comprehension. However, the human brain is used to perceive the characteristics of the data in associations, which these tools can provide only in a limited form. To overcome this issue, we introduce a GRANEF toolkit that demonstrates a new approach to exploratory network data analysis based on associations stored in a graph database.

Toolkit Design



- The toolkit's core is a graph database Dgraph (https://dgraph.io/) that stores transformed information from network traffic captures extracted by Zeek (https://zeek.org/) network security monitor.
- Modules are implemented as Docker containers.
- Web interface visualizes data as an interactive relationship diagram.

Database Schema



- Host – a device with IP address observed in the network traffic capture.
- Host-data – data related to the host extracted from network traffic.
- Connection – information about individual network connections.
- Application – application data extracted from the connection.

- The schema follows the format of Zeek logs and eases their extension.
- All edges are directional but allow reverse processing.

Analysis Query

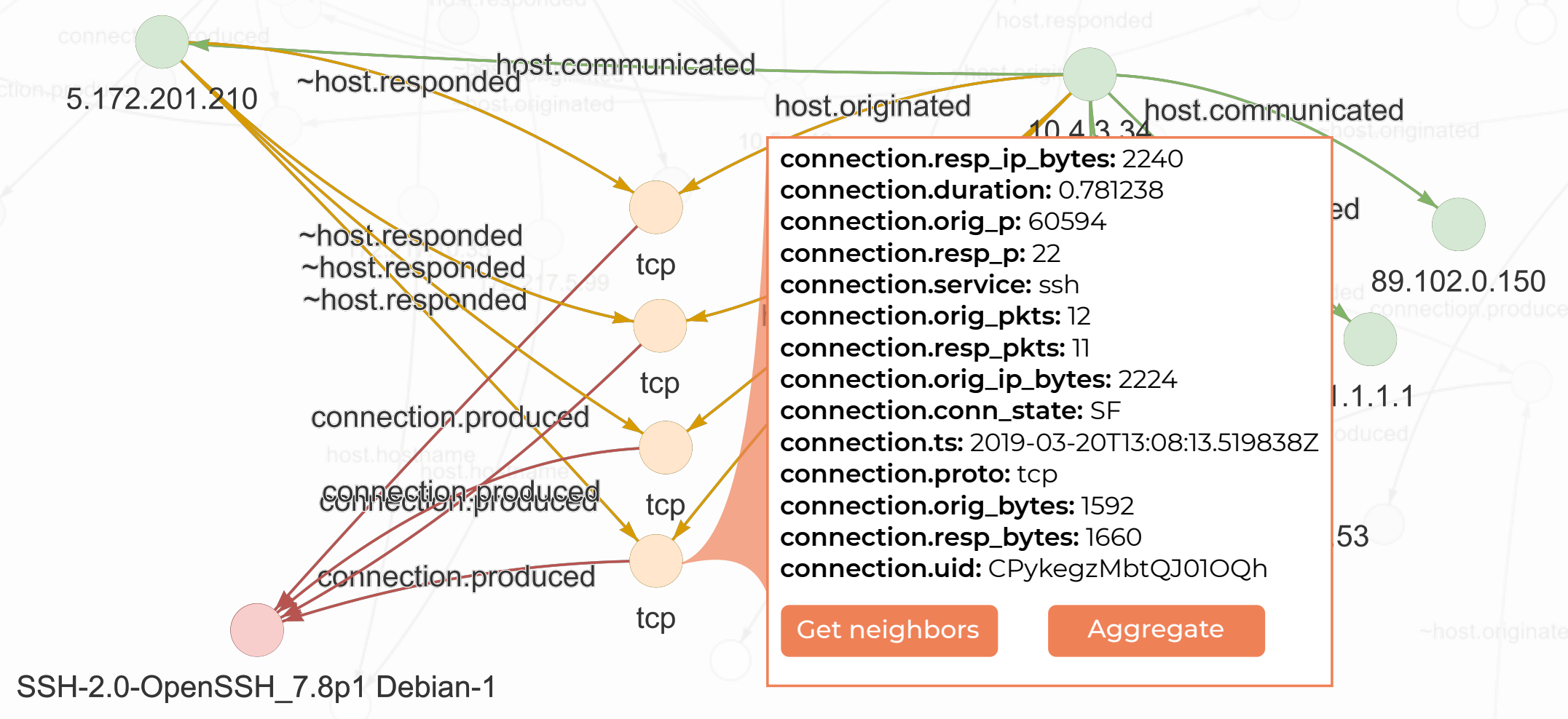
- DQL query with a selection of TCP connections with a file transfer from a local network:

```
{getConn(func: allof(host.ip, cidr, "192.168.0.0/16")) {
  name : host.ip
  host.originated @filter(eq(connection.proto, "tcp")) {
    expand(Connection)
    connection.produced {
      expand(_all_)
      files.fuid { expand(File) }
    }
    ~host.responded { responded_ip : host.ip }
  }
}}
```

- The toolkit contains an abstract layer API with common analysis functions to ease investigation.
- Results are provided as JSON or visualized in an interactive relationship diagram.

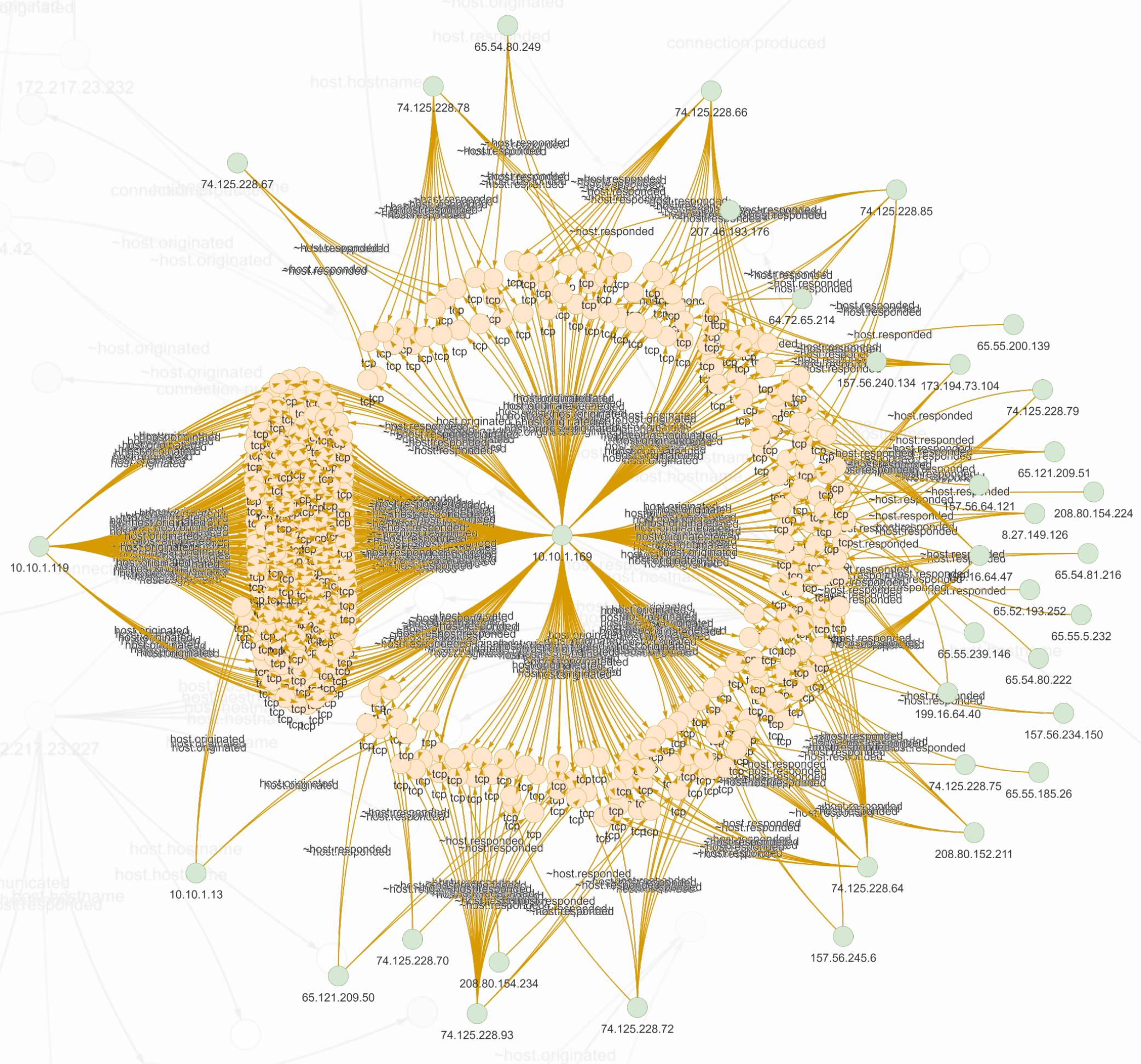
Network Forensics

- The interactive relationship visualization allows the analyst to get details about any selected node, go into the graph’s depth, and gain new observations.
- The API includes additional functionality to get initial insights about network connections and perform anomaly detections.
- Network traffic data can be easily enriched with additional information from external sources linked to existing nodes (e.g., asset management, OSINT, device logs, notes).



connection.resp_ip_bytes: 2240
connection.duration: 0.781238
connection.orig_p: 60594
connection.resp_p: 22
connection.service: ssh
connection.orig_pkts: 12
connection.resp_pkts: 11
connection.orig_ip_bytes: 2224
connection.conn_state: SF
connection.ts: 2019-03-20T13:08:13.519838Z
connection.proto: tcp
connection.orig_bytes: 1592
connection.resp_bytes: 1660
connection.uid: CPykegzMbtQJ01OQh

- The visualization allows the analyst to distinguish regular network traffic from suspicious one just at first glance based solely on the resulting pattern.
- The approach is not only the new method of data storage and querying, but it is a shift of mindset that enables the analyst to perceive network data in a new way.



MUNI ICS

SAPPAN Sharing and Automation for Privacy Preserving Attack Neutralization

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 833418.