HORIZON 2020
Information and Communication Technologies
Integrating experiments and facilities in FIRE+

Deliverable D3.6
Privacy Tools Final Iteration

Grant Agreement number:	**687884**
Project acronym:	**F-Interop**
Project title:	**FIRE+ online interoperability and performance test tools to support emerging technologies from research to standardization and market launch** **The standards and innovations accelerating tool**
Type of action:	**Research and Innovation Action (RIA)**
Project website address:	**www.finterop.eu**
Due date of deliverable:	**31/08/2018**
Dissemination level:	**PU/CO**

This deliverable has been written in the context of the Horizon 2020 European research project F-Interop, which is supported by the European Commission and the Swiss State Secretariat for Education, Research and Innovation. The opinions expressed and arguments employed do not engage the supporting parties.


Co-funded by the European Union


Co-funded by the Swiss Confederation

Document properties

Responsible partner	University of Luxembourg
Author(s)/editor(s)	Ion Turcanu (UL), Nizar Msadek (UL), Luca Lamorte (UL), Eunah Kim (DG)
Version	1.0
Keywords	Test tool, Privacy

Abstract

The deliverable D3.6 aims to describe the final release of the Privacy Test Tool, one of the testing tools available in F-Interop Platform, which has been developed within WP3, and specifically as an output of the Task 3.2. In particular, deliverable D3.6 focuses on the new features and achievements integrated in the Privacy Test Tool since the 1st iteration and composed of: general achievements, new features integrated into the Non-encrypted Traffic Analysis (NTA) module, and the description of a new module, namely the Encrypted Traffic Analysis (ETA).

Starting from the general definition of a Test Session, defined in Deliverable D1.1, and adopted as a lifecycle guideline for any F-Interop testing tool, this deliverable also extends the lifecycle guidelines of the Privacy Test Tool provided in deliverable D3.3 by providing a step by step execution guide of a Privacy Test Tool session.

Table of Contents

List of Figures

List of Tables

List of Acronyms

AB	AdaBoost
API	Application Program Interface
CoAP	Constrained Application Protocol
DG	Device Gateway
DHCP	Dynamic Host Configuration Protocol
DM	Data Matcher
DNS	Domain Name System
EU	European Union
ET	Extra-Trees
ETA	Encrypted Traffic Analysis
GUI	Graphical User Interface
HTTPS	Hypertext Transfer Protocol Secure
ICT	Information and Communication Technologies
ID	Identifier
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISO	International Standards Organization
IUT	Implementation Under Test
KNN	K-Nearest Neighbors
LAN	Local Area Network
MAC	Media Access Control
MI	Mandat International
MQTT	Message Queuing Telemetry Transport
NFV	Network Function Virtualization
NTA	Non-encrypted Traffic Analysis
OSI	Open Systems Interconnection
QoS	Quality of Service
RF	Random Forest
RS	Result Store
RSS	Result Store Service
SDN	Software Defined Networking
SME	Small Medium Enterprise
SVM	Support Vector Machine
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UL	University of Luxembourg
URL	Uniform Resource Locator
WAN	Wide Area Network
WLAN	Wireless LAN

1 Introduction

1.1 About F-Interop

F-Interop is a Horizon 2020 European Research project, which proposes to extend the European research infrastructure (FIRE+) with online and remote interoperability and performance test tools supporting emerging technologies from research to standardization and to market launch. The outcome will be a set of tools enabling:

- Standardization communities to save time and resources, to be more inclusive with partners who cannot afford travelling, and to accelerate standardization processes;
- SMEs and companies to develop standards-based interoperable products with a shorter time-to-market and significantly lowered engineering and financial overhead.

F-Interop intends to position FIRE+ as an accelerator for new standards and innovations.

1.2 Deliverable Objectives

1.2.1 Work package Objectives

- Research and develop performance test tools.
- Research and develop tools for privacy risk assessment
- Research and develop spatial representing tools to support experimenters.
- Research and integrate testing tools with network virtualization technologies such as OpenFlow / OpenDayLight based SDN / NFV environments.

1.2.2 Task Objectives

Work: Task 3.2 is to design methods for privacy analysis of the data exchanged while running different kind of tests on the F-Interop platform. First, the State-of-Art of traffic analysis on encrypted data flows is studied, in order to check potential suitable solutions for F-Interop. Then, tools for checking how privacy issues may raise due to information leakage are to be developed. In detail, it investigates how an adversary may get “sensitive” information (e.g., results of a test running on the shared platform) by passively observing patterns of the IoT communication.

Outcome: A tool for assessing the privacy leakage from IoT communication.

1.2.3 Deliverable Objectives and Methodology

The deliverable *D3.6 – Privacy Tools Final Iteration* describes the final iteration of the Privacy Test Tool as output of the Task 3.2. This document complements the information presented in deliverable *D3.3 – Privacy Tools 1st Iteration*, where the design principles of the Privacy Test Tool and a detailed description of each component of the tool have been presented. In particular, deliverable D3.6 focuses on the new features and achievements integrated in the Privacy Test Tool since the 1st iteration.

The Privacy Test Tool follows the generic F-Interop Test Tool Design, a common infrastructure used to encapsulate the tool business logic within the F-Interop platform. Such design is based on the concept of “F-Interop session”. The set of actions that a user (called the F-Interop-User) has to perform in order to execute Remote or Online tests in the F-Interop-Platform has been summarized in Table 1 of deliverable D3.3. To complement that information, this document provides also a video tutorial and a step-by-step guide to accompany the F-Interop-User through the setup and different execution steps of a Privacy Test Tool session (see Section 3.1).

2 Privacy Test Tool Overview

The primary goal of the privacy test tool is to define automatic methods to detect privacy and confidential data leaks while different kind of tests are executed on the F-Interop platform. Differently from other testing tools which are devoted to test Performance, Interoperability and Compatibility of some IoT protocols, this tool wants to identify the compliance with the current European Regulation in terms of data management, increasing the trustiness of the platform while communicating with the public internet.

For this reason, a general framework has been designed to detect privacy issues by analysing both *encrypted* and *non-encrypted* data traffic of IoT protocols. The framework is composed of two main modules: the **Encrypted Traffic Analysis (ETA)** module and the **Non-encrypted Traffic Analysis (NTA)** module. ETA is able to investigate how an adversary can get sensitive information related to IoT device activities by passively observing patterns of encrypted communication. NTA follows a *pattern matching* approach in the data payload of IoT protocols in order to detect what is considered *personal* and/or *private*.

The framework adopts an incremental approach by using plugins. These are components in charge of defining the privacy "patterns" and the best strategy to detect them. Following this approach, the Privacy Test Tool can easily integrate new application protocols, and, simultaneously, tune or add other patterns over the time. Such flexibility gives to the Privacy Test Tool the possibility to fit any needs for an F-Interop User, considering the specific requirements of the test he/she wants to perform and adapting the detection of what would be for him/her the meaning of "not to be disclosure" data.

The general design, modules functionality, as well as a first description of NTA and its corresponding data matchers (DMs) have been presented in detail in D3.3. The following sections describe the achievements and new features that have been added to the Privacy Test Tool since the 1st iteration presented in D3.3, which are divided in three main categories:

1. General achievements – which affect the entire Privacy Test Tool:
   - Test Execution Tutorial
   - F-Interop new APIs integration
   - Result Store Service (RSS) integration
2. New features added to the NTA module:
   - MQTT protocol support
   - Custom Privacy Policy definition
3. ETA module – general description and implementation.

3 General Achievements

This section describes the general achievements with respect to the 1st iteration of the Privacy Test Tool. Technical achievements affect mostly the way the tool interacts with other core components of the F-Interop platform and consists in adaptation to the new F-Interop APIs and integration of the Result Store Service (RSS). Non-technical achievements consist in a video tutorial and a step by step user guide describing a test execution using the Privacy Test Tool.

3.1 Test Execution Tutorial

The video tutorial demonstrating a Privacy Test Tool session can be found at the following link (see Figure 1): https://www.youtube.com/watch?v=YrcyikUFcHM&t=1s.

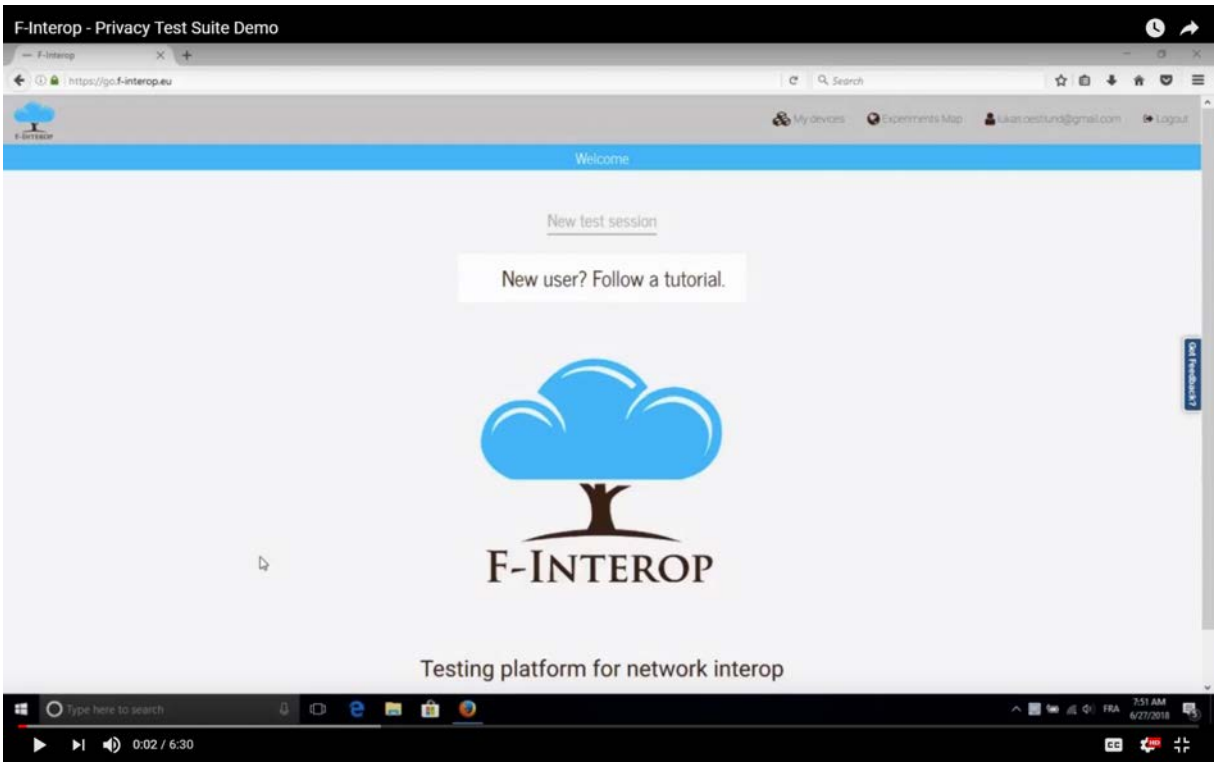
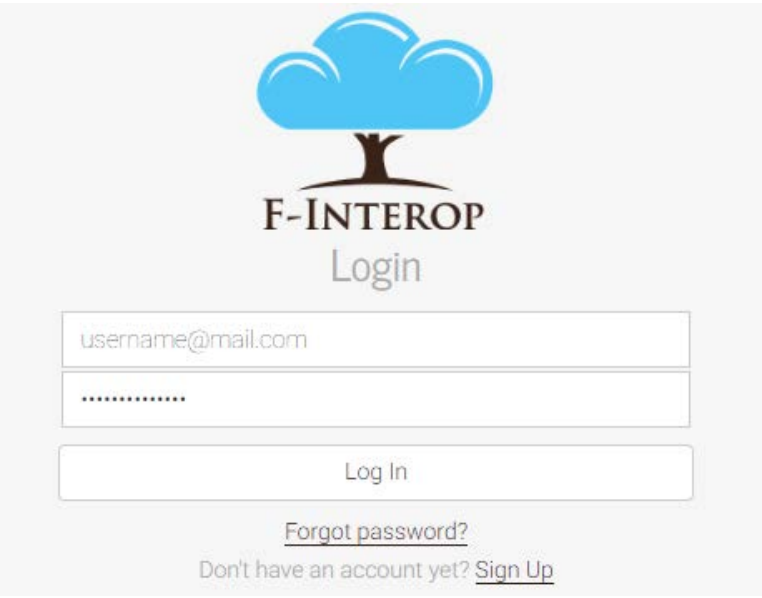
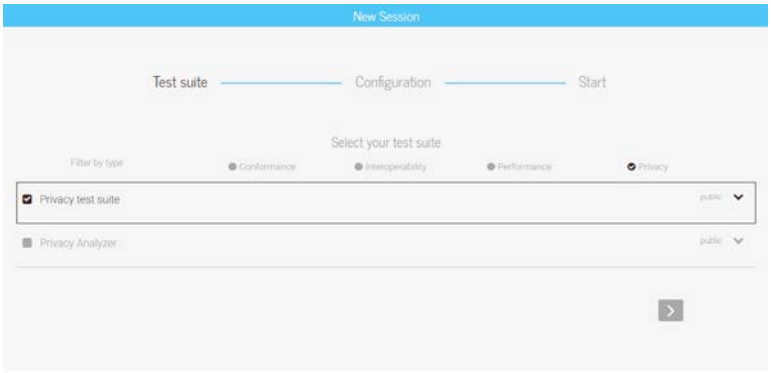
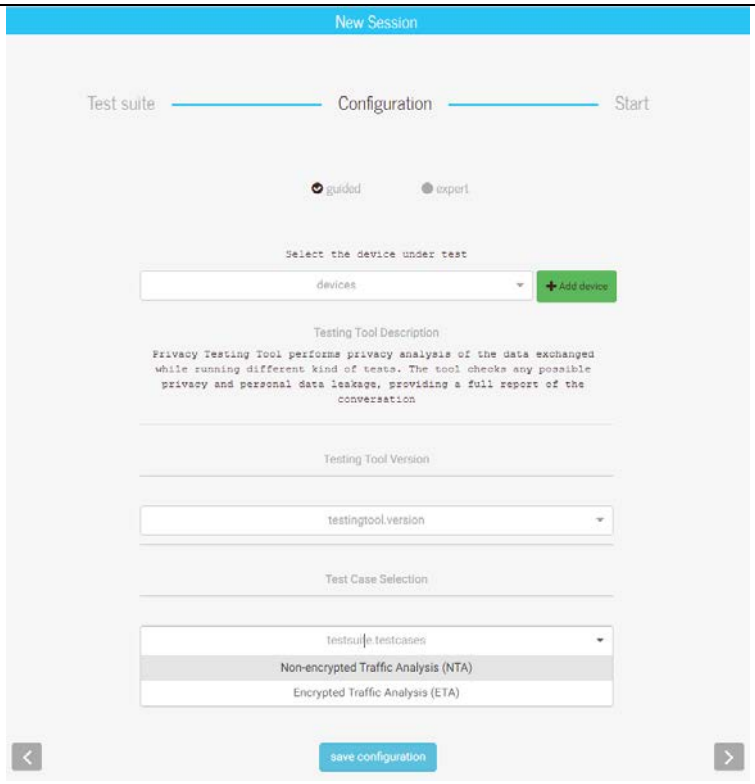
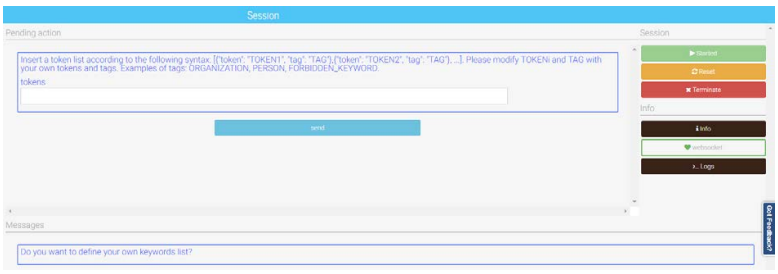


Figure 1: Capture of the video tutorial demonstrating a Privacy Test Tool session

The set of actions for executing the Privacy Test Tool in the F-Interop platform are summarized in the following table:

Step	Action	Description
0	FI-User authentication and authorization. IUT registration / identification	
1	Test suites discovery and selection	
2	Resource description	(testing tool configuration)

		
3	Resource reservation	At this stage there is no need to pre-reserve resources.
4	Resource provisioning, configuration and session setup	(instantiation of the F-Interop session) 
5	Test execution	(Custom Privacy Policy definition, NTA module)  (PCAP file selection)

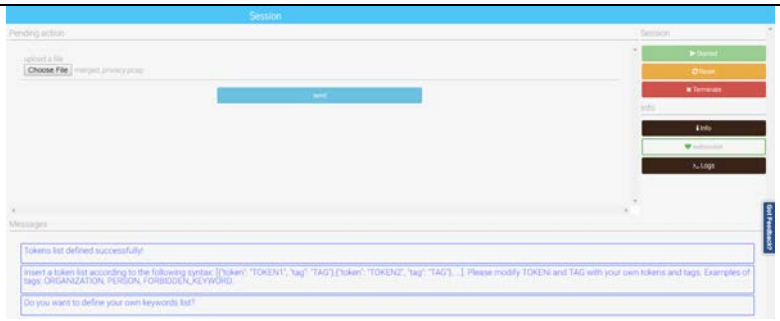
		
6	Results analysis and report	(privacy report containing identified issues, NTA module) 
7	Session storage	Performed by the RSS module, not visualized here.

Table 1: Privacy Test Tool Session

3.2 F-Interop New APIs Integration

3.2.1 Event Bus Overview

F-Interop platform consists of a number of core components and a number of testing tools, which have to interact among each other. The modularity of the F-Interop architecture requires a flexible approach regarding the data exchange among different components. When designing the communication bus, three main requirements have been considered:

- **Reliability** – the communication bus has to guarantee reliable message delivery/exchange.
- **Scalability** – the communication bus must support a scalable number of concurrent test session and messages.
- **Extendibility** – adding new message definitions, tools, and core components must be a relatively easy task.

F-Interop implementation of the communication bus is based on an **Event Bus**, a mechanism that provides a secure channel where all the communication is done, including control messages, raw data packets and logs. As a technological solution, F-Interop has adopted RabbitMQ, which provides all capabilities required by the platform. In the following, we summarize the most important features of the event bus:

- Every message in the event bus contains a **routing key** (called also topic) which categorizes the message. These are standardized by the F-Interop platform.
- Every message in the event bus contains a **payload**.
- Payloads of messages which are part of the **core APIs** are standardized.
- Payloads of some components of the architecture which are not part of the core APIs are not standardized. The testing tools developers are free to define their own messages.
- **JSON** is the data format used for the application messages exchanged in the bus.
- **Isolation** of the test session is performed using virtual hosts (vhosts).
- Each session has an interface for monitoring the load and enable easy debugging.

A more detailed description of the initial implementation of the event bus is provided in Section 2.1.3 of D3.3. An important aspect that is missing in D3.3 and that has been later introduced by the F-Interop

developing team is the core APIs standardization, highlighted in Figure 2. Section 3.2.2 describes the integration of the Privacy Test Tool with these standardized core APIs.

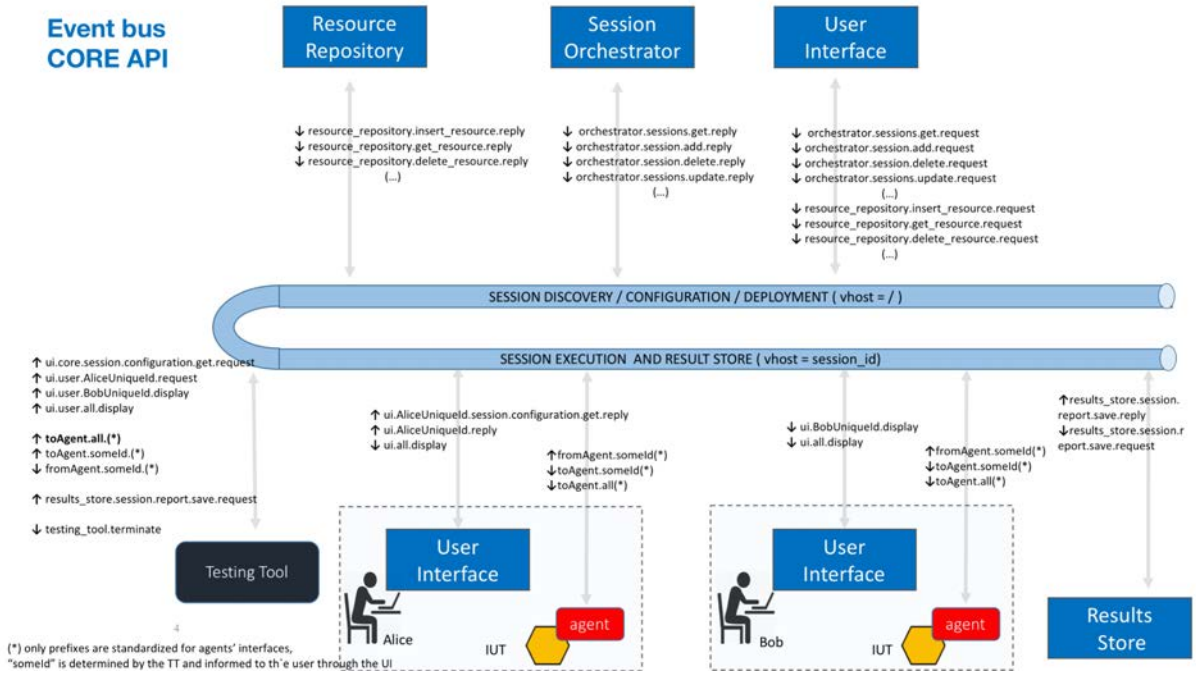


Figure 2: Event Bus core APIs

3.2.2 Integration of F-Interop core APIs

Privacy Test Tool adapted to these new changes of the F-Interop core APIs, which consist mostly in the interaction between the test tool itself and the F-Interop Graphical User Interface (GUI). To this end, in order to display generic messages on the GUI, Privacy Test Tool is using the **MsgUiDisplayMarkdownText** message definition:

```
-  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
Message routing key: ui.user.all.display
-  -  -
Message properties: {
    "timestamp": 1527154018,
    "message_id": "8580421b-f568-464e-a01c-989dd25834c6",
    "content_type": "application/json"
}
-  -  -
Message body: {
    "_api_version": "1.1.0",
    "fields": [
    {
        "type": "p",
        "value": "Hello World!"
    }
    ],
    "level": null,
    "tags": {}
}
-  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
```

Privacy Test Tool allows a user to decide whether he/she wants to define a Custom Privacy Policy as soon as the session has been initialized by the Session Orchestrator, by publishing a YES/NO button using a **MsgUiRequestConfirmationButton**:

```
  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
Message routing key: ui.user.all.request
 -  -  -
Message properties: {
    "reply_to": "ui.user.all.reply",
    "timestamp": 1527154018,
    "correlation_id": "26c87027-6a69-4fa2-8b12-d4bce9f9fddb",
    "message_id": "26c87027-6a69-4fa2-8b12-d4bce9f9fddb",
    "content_type": "application/json"
}
 -  -  -
Message body: {
    "_api_version": "1.1.0",
    "fields": [
        {
            "type": "button",
            "name": "test_button",
            "value": true
        }
    ],
    "tags": {}
}
 -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
```

If the user chooses to define a Custom Privacy Policy, then the Privacy Test Tool provides a means to insert this policy by publishing a **MsgUiRequestTextInput** message:

```
  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
Message routing key: ui.user.all.request
 -  -  -
Message properties: {
    "reply_to": "ui.user.all.reply",
    "timestamp": 1527154018,
    "correlation_id": "039ea17d-3b98-4e41-9539-ebc6ccd424c5",
    "message_id": "039ea17d-3b98-4e41-9539-ebc6ccd424c5",
    "content_type": "application/json"
}
 -  -  -
Message body: {
    "_api_version": "1.1.0",
    "fields": [
        {
            "type": "text",
            "name": "input_name"
```

```
        }
    ],
    "tags": {}
}
 -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
```

In order to allow a user to upload a PCAP file, the Privacy Test Tool sends a **MsgUiRequestUploadFile** message to the GUI:

```
  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
Message routing key: ui.user.all.request
 -  -  -
Message properties: {
    "reply_to": "ui.user.all.reply",
    "timestamp": 1527154018,
    "correlation_id": "b7da591f-b651-404c-99c6-2d587d558e03",
    "message_id": "b7da591f-b651-404c-99c6-2d587d558e03",
    "content_type": "application/json"
}
 -  -  -
Message body: {
    "_api_version": "1.1.0",
    "fields": [
        {
            "type": "file",
            "name": "upload a file"
        }
    ],
    "tags": {}
}
 -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
```

3.3 Result Store Service Integration

Results Store (RS) is a core F-Interop service that allows to store and retrieve results or intermediate results generated from Testing Tools (TT). It uses MongoDB as database, while the results are stored as Binary JSON (BSON).

In order to access the RS, Privacy Test Tool uses the Result Store Service (RSS), an F-Interop service that acts as a bridge between a session virtual host and the RS. To this end, the Supervisor configuration template of the Privacy Test Tool has been modified as follows:

```
; A user can write comments here to document how the processes are launched
and interact with
; each others.
; amqp_url pattern : amqp://user:password@host:port/virtual_host
; {{ my_variable|default('my_variable is not defined') }}


; DOCKER PARAMS:
```

```
; --rm : Automatically remove the container when it exits
; --name : Assign a name to the container
; --privileged=true : Allows processes to create tun and modify network params
; --sysctl net.ipv6.conf.all.disable_ipv6=0 : ipv6 must be enabled testing tool's agent

; IMPORTANT
; put explicit command at the end (e.g. supervisord --nodaemon --configuration supervisor.conf)

[program:{{ session }}|testing_tool]
stopsignal=TERM
killasgroup=true
autostart=false
stdout_logfile = %(here)s/logs/{{ session }}-testing_tool-stdout.log
stderr_logfile = %(here)s/logs/{{ session }}-testing_tool-stderr.log
command = docker run
    --env AMQP_URL={{ amqp_url }}
    --env AMQP_EXCHANGE={{ amqp_exchange }}
    --rm
    --privileged=true
    --sysctl net.ipv6.conf.all.disable_ipv6=0
    --name="session_{{ session }}-testing_tool-privacy"
    testing_tool-privacy
    supervisord --nodaemon --configuration supervisor.conf


[program:{{ session }}|service_results_store]
stopsignal=TERM
killasgroup=true
autostart=false
stdout_logfile = %(here)s/logs/{{ session }}-service-results-store-stdout.log
stderr_logfile = %(here)s/logs/{{ session }}-service-results-store-stderr.log
command = docker run
    --env AMQP_URL={{ amqp_url }}
    --env AMQP_EXCHANGE={{ amqp_exchange }}
    --env RS_AMQP_URL={{ rs_amqp_url }}
    --env RS_AMQP_EXCHANGE={{ rs_amqp_exchange }}
    --rm
    --name="session_{{ session }}-service-results-store"
    service-results-store
```

In order to save the results of the privacy analysis, Privacy Test Tool uses the **MsgReportSaveRequest** message definition, provided by the RSS:

```
- - -
ROUTING_KEY: results_store.session.report.save.request
 - - -
```

```
HEADERS: None
 - - -
PROPS: {
    "reply_to": "results_store.session.report.save.reply",
    "correlation_id": "1a58de57-176e-41c4-bca3-e81b968142c9",
    "content_type": "application/json"
}
 - - -
BODY: {
    "type": "final",
    "data": {
        "some_key": {
            "another_key": "some_value"
        },
        "yet_another_key": 42
    }
}
```

If the results were correctly stored, the RS sends a **MsgReportSaveReply:**

```
- - -
ROUTING_KEY: results_store.session.report.save.reply
 - - -
HEADERS: None
 - - -
PROPS: {
    "reply_to": null,
    "correlation_id": "1a58de57-176e-41c4-bca3-e81b968142c9",
    "content_type": "application/json"
}
 - - -
BODY: {
    "ok": true
}
```

4 Achievements in the NTA Module

This section describes the new features that have been added to the Non-encrypted Traffic Analysis (NTA) module of the Privacy Test Tool. In particular, we focus on the newly added support for the MQTT IoT communication protocol, and the user-defined Custom Privacy Policy capability.

4.1 MQTT Protocol Support

The 1st iteration of the Privacy Test Tool described in deliverable D3.3 used to support only CoAP protocol, which is considered one of the most popular emerging IoT protocols. In the final iteration, the Privacy Test Tool has been extended with the capability to support privacy issues detection from MQTT data traffic, which is also one of the popularly used protocol in IoT domain.

Message Queuing Telemetry Transport (MQTT) is a publish-subscribe-based IoT communication protocol [1]. It is normally used in the network environment of limited network bandwidth. In order to process MQTT and CoAP messages, Privacy Test Tool uses TSHARK dissectors to translate PCAP files into JSON files. In the following, we present an example of an MQTT frame structure exported to JSON format:

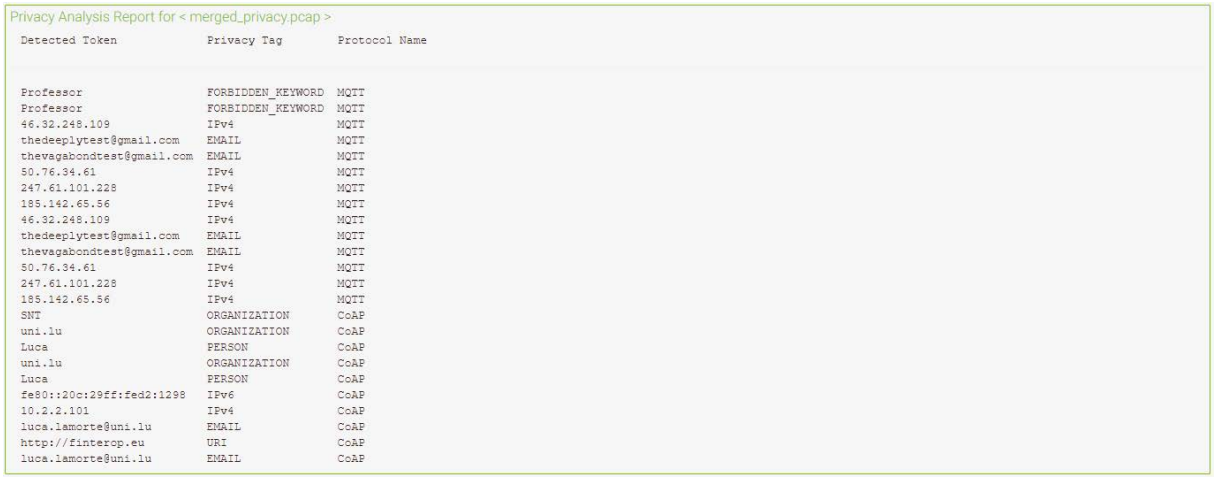

```
{
  "_index": "packets-2018-07-19",
  "_type": "pcap_file",
  "_score": null,
  "_source": {
    "layers": {
      "frame": {
        "frame.encap_type": "1",
        "frame.time": "May 11, 2018 18:11:16.203765000 CEST",
        "frame.offset_shift": "0.000000000",
        "frame.time_epoch": "1526055076.203765000",
        "frame.time_delta": "0.000039000",
        "frame.time_delta_displayed": "0.000039000",
        "frame.time_relative": "39.237071000",
        "frame.number": "5",
        "frame.len": "313",
        "frame.cap_len": "313",
        "frame.marked": "0",
        "frame.ignored": "0",
        "frame.protocols": "eth:ethertype:ipv6:tcp:mqtt",
        "frame.coloring_rule.name": "TCP",
        "frame.coloring_rule.string": "tcp"
      },
      "eth": {
        "eth.dst": "00:00:00:00:00:00",
        "eth.dst_tree": {
          "eth.dst_resolved": "00:00:00_00:00:00",
          "eth.addr": "00:00:00:00:00:00",
          "eth.addr_resolved": "00:00:00_00:00:00",
          "eth.lg": "0",
          "eth.ig": "0"
        },
        "eth.src": "00:00:00:00:00:00",
        "eth.src_tree": {
          "eth.src_resolved": "00:00:00_00:00:00",
          "eth.addr": "00:00:00:00:00:00",
          "eth.addr_resolved": "00:00:00_00:00:00",
          "eth.lg": "0",
          "eth.ig": "0"
```

```
        },
        "eth.type": "0x000086dd"
      },
      "ipv6": {
        "ipv6.version": "6",
        "ip.version": "6",
        "ipv6.tclass": "0x00000000",
        "ipv6.tclass_tree": {
          "ipv6.tclass.dscp": "0",
          "ipv6.tclass.ecn": "0"
        },
        "ipv6.flow": "0x000e6b3b",
        "ipv6.plen": "259",
        "ipv6.nxt": "6",
        "ipv6.hlim": "64",
        "ipv6.src": "::1",
        "ipv6.addr": "::1",
        "ipv6.src_host": "::1",
        "ipv6.host": "::1",
        "ipv6.dst": "::1",
        "ipv6.addr": "::1",
        "ipv6.dst_host": "::1",
        "ipv6.host": "::1"
      },
      "tcp": {
        "tcp.srcport": "48718",
        "tcp.dstport": "1883",
        "tcp.port": "48718",
        "tcp.port": "1883",
        "tcp.stream": "1",
        "tcp.len": "227",
        "tcp.seq": "39",
        "tcp.nxtseq": "266",
        "tcp.ack": "5",
        "tcp.hdr_len": "32",
        "tcp.flags": "0x00000018",
        "tcp.flags_tree": {
          "tcp.flags.res": "0",
          "tcp.flags.ns": "0",
          "tcp.flags.cwr": "0",
          "tcp.flags.ecn": "0",
          "tcp.flags.urg": "0",
          "tcp.flags.ack": "1",
          "tcp.flags.push": "1",
          "tcp.flags.reset": "0",
          "tcp.flags.syn": "0",
          "tcp.flags.fin": "0",
          "tcp.flags.str":
"\u00c2\u00b7\u00c2\u00b7\u00c2\u00b7\u00c2\u00b7\u00c2\u00b7\u00c2\u00b7\u00c2\u00b7\u
00c2\u00b7AP\u00c2\u00b7\u00c2\u00b7\u00c2\u00b7\u00c2\u00b7"
        },
        "tcp.window_size_value": "342",
        "tcp.window_size": "342",
        "tcp.window_size_scalefactor": "-1",
        "tcp.checksum": "0x0000010b",
        "tcp.checksum.status": "2",
        "tcp.urgent_pointer": "0",
        "tcp.options": "01:01:08:0a:57:cd:62:2d:57:cd:62:2d",
        "tcp.options_tree": {
          "tcp.options.nop": "01",
          "tcp.options.nop_tree": {
```

```
            "tcp.option_kind": "1"
          },
          "tcp.options.nop": "01",
          "tcp.options.nop_tree": {
            "tcp.option_kind": "1"
          },
          "tcp.options.timestamp": "08:0a:57:cd:62:2d:57:cd:62:2d",
          "tcp.options.timestamp_tree": {
            "tcp.option_kind": "8",
            "tcp.option_len": "10",
            "tcp.options.timestamp.tsval": "1473077805",
            "tcp.options.timestamp.tsecr": "1473077805"
          }
        },
        "tcp.analysis": {
          "tcp.analysis.acks_frame": "4",
          "tcp.analysis.ack_rtt": "0.000039000",
          "tcp.analysis.bytes_in_flight": "227",
          "tcp.analysis.push_bytes_sent": "227"
        },
        "Timestamps": {
          "tcp.time_relative": "0.000092000",
          "tcp.time_delta": "0.000039000"
        },
        "tcp.payload":
"30:e0:01:00:04:74:65:73:74:54:68:65:20:49:6e:74:65:72:64:69:73:63:69:70:6c
:69:6e:61:72:79:20:43:65:6e:74:72:65:20:66:6f:72:20:53:65:63:75:72:69:74:79
:2c:20:52:65:6c:69:61:62:69:6c:69:74:79:20:61:6e:64:20:54:72:75:73:74:20:28
:53:6e:54:29:20:63:6f:6e:64:75:63:74:73:20:69:6e:74:65:72:6e:61:74:69:6f:6e
:61:6c:6c:79:20:63:6f:6d:70:65:74:69:74:69:76:65:20:72:65:73:65:61:72:63:68
:20:69:6e:20:69:6e:66:6f:72:6d:61:74:69:6f:6e:20:61:6e:64:20:63:6f:6d:6d:75
:6e:69:63:61:74:69:6f:6e:20:74:65:63:68:6e:6f:6c:6f:67:79:20:28:49:43:54:29
:20:77:69:74:68:20:68:69:67:68:20:72:65:6c:65:76:61:6e:63:65:20:63:72:65:61
:74:69:6e:67:20:73:6f:63:69:6f:2d:65:63:6f:6e:6f:6d:69:63:20:69:6d:70:61:63
:74:2e",
        "tcp.pdu.size": "227"
      },
      "mqtt": {
        "mqtt.hdrflags": "0x00000030",
        "mqtt.hdrflags_tree": {
          "mqtt.msgtype": "3",
          "mqtt.dupflag": "0",
          "mqtt.qos": "0",
          "mqtt.retain": "0"
        },
        "mqtt.len": "224",
        "mqtt.topic_len": "4",
        "mqtt.topic": "test",
        "mqtt.msg": "The Interdisciplinary Centre for Security,
Reliability and Trust (SnT) conducts internationally competitive research
in information and communication technology (ICT) with high relevance
creating socio-economic impact."
      }
    }
  }
}
```


The conversion from PCAP to JSON has a twofold beneficial effect: the first one enables an easier and efficient way to access packet fields, by using well-known JSON libraries available in different programming languages; the second one translates each layer of the frame using the CoAP and

MQTT dissectors. These dissectors are plugins of the TSHARK tool, with the scope of converting raw data into specific protocol layer fields, such as headers, flags, payloads, etc. Figure 3 presents an example of a privacy analysis report containing privacy issues detected from both CoAP and MQTT data traffic.



Privacy Analysis Report for < merged_privacy.pcap >

Detected Token	Privacy Tag	Protocol Name
Professor	FORBIDDEN_KEYWORD	MQTT
Professor	FORBIDDEN_KEYWORD	MQTT
46.32.248.109	IPv4	MQTT
thedeeplytest@gmail.com	EMAIL	MQTT
thevagabondtest@gmail.com	EMAIL	MQTT
50.76.34.61	IPv4	MQTT
247.61.101.228	IPv4	MQTT
185.142.65.56	IPv4	MQTT
46.32.248.109	IPv4	MQTT
thedeeplytest@gmail.com	EMAIL	MQTT
thevagabondtest@gmail.com	EMAIL	MQTT
50.76.34.61	IPv4	MQTT
247.61.101.228	IPv4	MQTT
185.142.65.56	IPv4	MQTT
SNT	ORGANIZATION	CoAP
uni.lu	ORGANIZATION	CoAP
Luca	PERSON	CoAP
uni.lu	ORGANIZATION	CoAP
Luca	PERSON	CoAP
fe80::20c:29ff:fed2:1298	IPv6	CoAP
10.2.2.101	IPv4	CoAP
luca.lamorte@uni.lu	EMAIL	CoAP
http://finterop.eu	URI	CoAP
luca.lamorte@uni.lu	EMAIL	CoAP

Figure 3: Report containing privacy issues detected from CoAP and MQTT data traffic

4.2 Custom Privacy Policy Definition

Custom Privacy Policy can be defined as the capability to create specific lists of keywords that an F-Interop user deems to be private. Privacy Test Tool allows users to define Custom Privacy Policies as soon as the session is up and running, as can be seen from Figure 4.

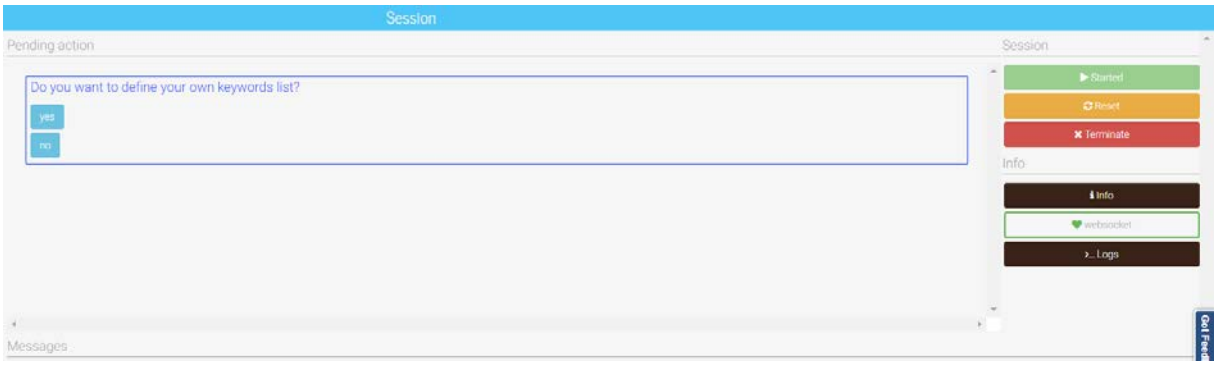


Figure 4: A capture showing the possibility to define a Custom Privacy Policy

If the user decides to define a Custom Privacy Policy, he/she will be provided with the possibility to insert a string, as can be seen in Figure 5.

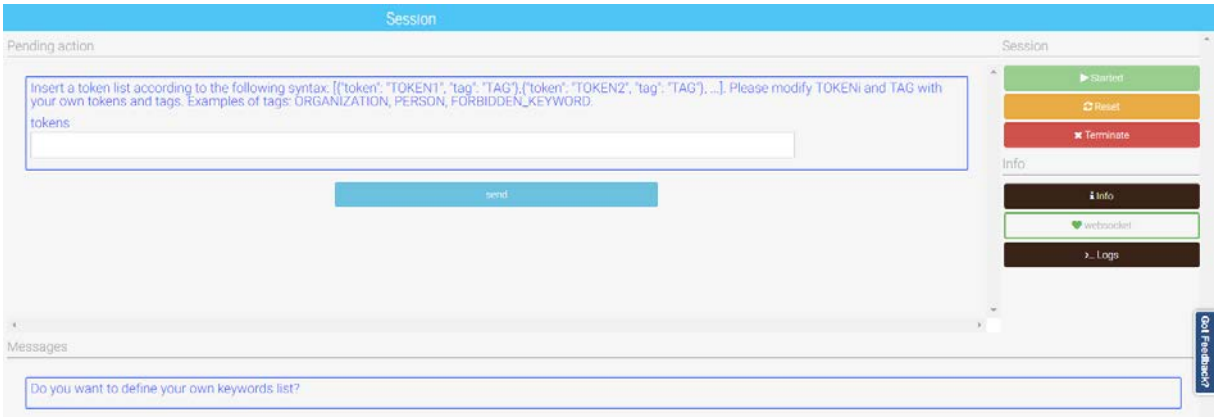


Figure 5: A capture illustrating the possibility to provide input text

The policy must follow a JSON format syntax:

```
[{"token": "TOKEN1", "tag": "TAG"},{"token": "TOKEN2", "tag": "TAG"}, ...]
```

Here `"token"` and `"tag"` are special identifiers used to identify the token and the tag, and they should not be changed by the user. `"TOKENi"` and `"TAG"` have to be defined by the user and represent the actual keywords which the F-Interop user wants to identify and the associated privacy tag to be assigned (e.g., ORGANIZATION, PERSON, FORBIDDEN_KEYWORD, etc). If the above mentioned syntax is not followed by the user (i.e. the string cannot be interpreted by a JSON parser, or `"token"` and `"tag"` have been altered by the user), an error message will be displayed, as can be seen in Figure 6. Notice that there are no restrictions with respect to the `"TOKENi"` and `"TAG"` relationship, meaning that the user can associate any `"TAG"` to any `"TOKENi"`.

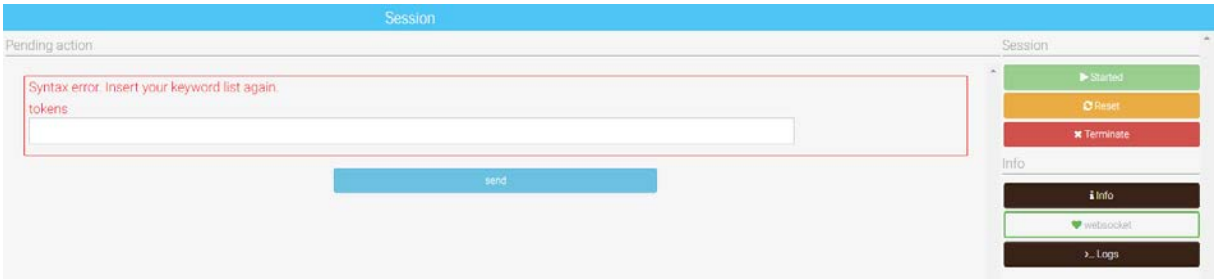


Figure 6: The error message displayed if the policy syntax format is not followed

5 ETA Module: Design and Implementation

Even though the F-Interop platform [2] has been designed to keep all communications secure – using special paradigms and protocols - there are still techniques that allow for inferring sensitive information about device activities. These can be achieved by analyzing the flow of exchanged encrypted communications. The 1st iteration of the Privacy Test Tool does not consider this issue. It was only designed to perform privacy tests based on (plain-text) payload contents. Therefore, we think there is room for improvement with the Encrypted Traffic Analysis (ETA) module proposed in this section. We aim to investigate how an adversary may get sensitive information related to device activities by passively observing patterns of encrypted communication. These patterns are extracted from meta-information such as packet size and timing, without breaking the encryption. We argue that this is important not only because it is one of the major requirements of F-Interop, but also because such test is required by every User that wants to use the platform.

Based on this consideration, the ETA module provides an assessment report on how accurately an adversary can identify sensitive information, such as the type of devices, solely based on the analysis of encrypted traffic. To this end, this section offers as contribution the following aspects:

1. A formal introduction of the problem statement and goals considered in our study in order to indicate the specific purpose of the ETA module (see Section 5.1).
2. An overview of the ETA approach as well as an explanation of its different functional parts including traffic segmentation based on sliding windows, feature extraction by means of network analysis, and a survey of several classification algorithms (see Section 5.2).
3. An implementation of the ETA approach supporting the underlying architecture of F-Interop platform [3].

After the description of the algorithm which is given in the following, Section 5.2 introduces the reference dataset considered in this study. Then, the main steps for generating the output results are explained in Section 5.3.

5.1 ETA Approach

5.1.1 Basic Idea

The encrypted traffic analysis approach is the core algorithm implemented in the ETA module. The basic idea behind it is roughly spoken "*if we can accurately identify sensitive information much like the type of devices solely based on encrypted traffic analysis, the communication of those devices is not secure*". Therefore, as shown in Figure 7, the algorithm takes in input a dump of encrypted traffic, and applies sophisticated learning techniques to it. As a result, the algorithm outputs a privacy report indicating of what accuracy degree one device type can be identified according to a set of performance indicator metrics.

In the following, a precise problem statement for the proposed device identification approach is given. Afterwards, we discuss the main methodology of it.

5.1.2 Problem Statement and Goal

Given a distributed system of several IoT devices D and a set of encrypted traffic P, we treat the task of device identification as a multi-class classification problem. We wish to map each encrypted traffic to the device type that is most likely to have produced it. In this work, we limit our attention to the case that D is finite and we have meta-information on all devices in the system network. Let $P^{d_i} = <.., p_t^i, ..>$ be the traffic activities generated by $d_i \in D$ where p_t^i corresponds to one of its packets, t its arrival time, and $[p_t^i]$ its duration. The main goal is to look for a Machine Learning classification model that once trained - and benchmarked using its best tuning parameters - is able to identify the corresponding device d_i for any unseen encrypted traffic P^{d_i}, and thus for $\forall\, d_i \in D$. So, the first step towards this goal is to enable an extraction of features at runtime. The second step is to build a classifier that takes as input those features and makes classifications in order to infer the type of devices. In the following the different parts of our approach are introduced.

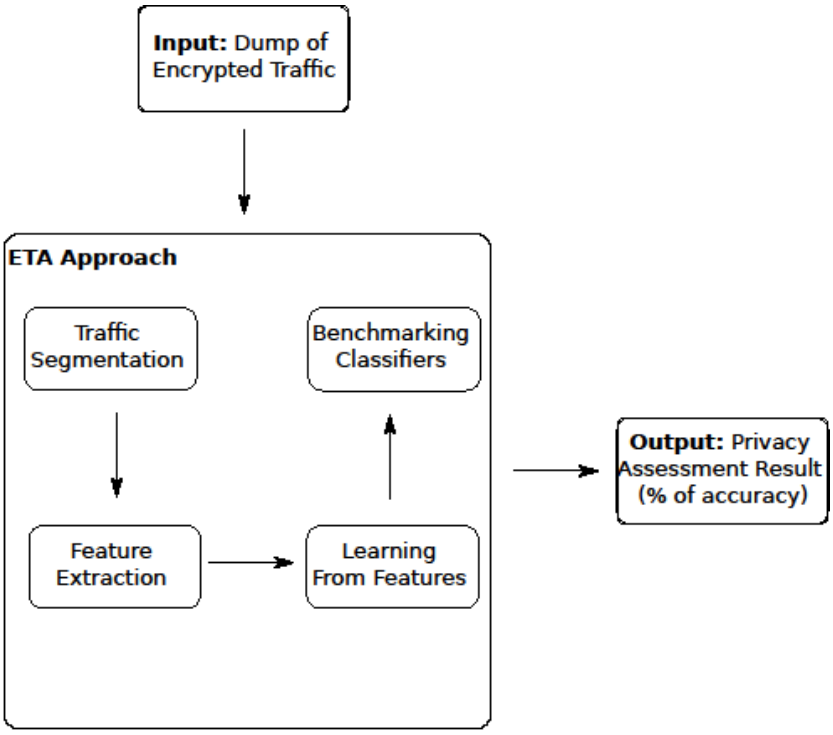


Figure 7: A simplified representation of the ETA module working flow

5.1.3 Methodology

In this subsection, we describe the research paradigms we use for our encrypted traffic analysis approach. This consists of the following aspects: (i) a traffic segmentation based on a sliding window technique; (ii) an analysis of encrypted traffic to extract features, and (iii) a study of various classification algorithms.

5.1.3.1 Traffic Segmentation

For traffic segmentation we adopt the concept of sliding windows. This technique comes along with the main advantage to easily isolate traffic activities processed in the past from recent ones. It makes use of two important parameters: namely length l and shift r. Figure 8 gives an overview of this technique using an example of three devices $d_1, d_3, d_3 \in D$, where $l = r$, j is the time at which the first window begins and $j + 1$ the time at which it terminates and the next window begins. For the first time, the traffic activities are extracted from window w_j. We denote by $P^{d_i}_{[w_j]} = < p^i_j, \dots , p^i_{j+l-1}, >$ the traffic activities of device d_i within window w_j. Then, the next window moves by shifting r from w_j to w_{j+r} and the traffic activities are extracted from $P^{d_i}_{[w_{j+r}]} = < p^i_{j+r}, \dots , p^i_{j+r+l-1}, >$. This process is iteratively executed until all windows are explored.

5.1.3.2 Feature Extraction

We are interested to look for features that are time-invariant, or at least remain the same for a reasonably long time. One simple technique would be to consider the payload content of packets. However, as discussed earlier, with encrypted traffic we cannot rely on payload content. We therefore extract our features only from packet headers, using the following two techniques:

- *Basic Features* based on dominant protocol analysis: we analyse the different OSI layer protocols that the devices use to interact. We try to identify the set of dominant protocols which are important to extract features that can implicitly reveal unique characteristics about devices. We study their impact in terms of predictive performance and tune their selection iteratively. The set of features considered in this study include the type of dominant protocols,

port intervals, packet size, packet amount, availability time, inter-arrival times, cumulative

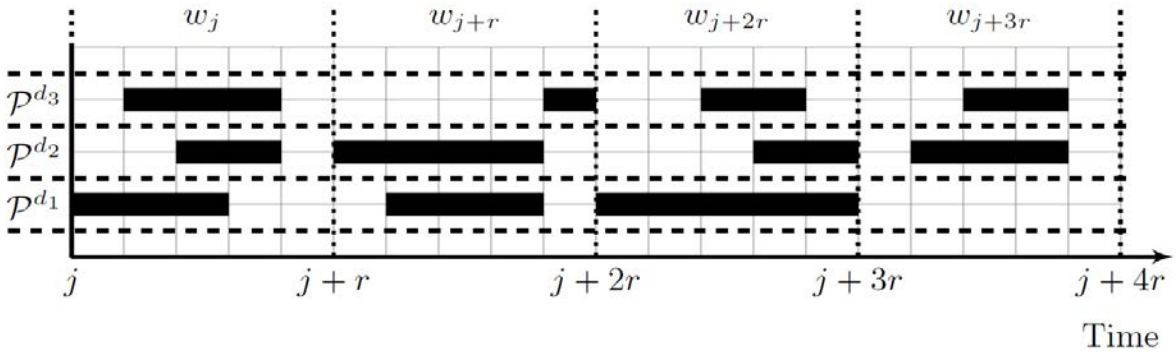


Figure 8: Traffic segmentation using the concept of sliding windows

count, etc.

- *Derived Features* based on statistical traffic rates: which are obtained by converting the flow of basic features into statistical distributions. This set includes minimum, maximum, mean, median, standard deviation, variance, etc.

5.1.3.3 Learning Algorithms

To find a suitable learning model for our problem, we started with a survey [4] [5] [6] and selected different classification algorithms to be benchmarked later in the evaluation. The following introduces the list of algorithms considered in this work along with their corresponding tuning parameters.

- **K-nearest Neighbors (KNN)** [7] are simple non-parametric learning algorithms which do not require a model to be fitted. Given an input sample x, the k training examples closest in distance to x are determined. Sample x is classified using a majority vote among these k neighbors. The parameter k characterizing this amount of neighbors is also known as $n\_neighbors$, when using the terminology of *scikit-learn* [8]. The other parameter used to determine the way of prediction is called *weights*. If it is set to *uniform* then all points in each neighbourhood are weighted equally. However, it is set to *distance* then the weight points will be calculated based on the inverse of their distance, which means that closer neighbours of a query point will have a greater influence than neighbours that are further away.
- **Support Vector Machine (SVM)** [9] is a machine learning method able to find an optimal boundary between two different classes, the so-called *separating hyperplane*. This is done by maximizing the margin amongst patterns belonging to the classes. For this purpose, the algorithm calculates support vectors, which are data points from the training data that lie close to the decision surface. The SVM algorithm does not only perform linear classification, but also non-linear classification using what is called the kernel function, i.e. cases where the decision function is not a linear function of the data. When training SVM, two parameters must be considered: *gamma* and C. The *gamma* parameter defines how much influence a single training example has. The parameter C, however, trades off misclassification of training examples against simplicity of the decision function, i.e., a low C makes the decision function smooth, while a high C aims at classifying all training examples correctly.
- **Random Forest (RF)** [10] is an ensemble learning techniques that combines numerous decision trees at training time using several bootstrapped samples of the training data. The class prediction is the result of combining these individual tree predictions. The algorithm makes use of a randomly chosen subset of features to find the best split for each node and it is robust against overfitting. Two important parameters must be considered: *max_depth* and *max_features*. The first parameter represents the maximum number of trees in the forest, whereas the second stands for the maximum number of considered features for node splitting.
- **AdaBoost (AB)** [11] is a machine learning technique invoking weak learners in a series of iterations. One of the main tasks of the algorithm is to maintain a set of weights over the training set. Initially, all weights are set equally. After each iteration, the weights of incorrectly classified samples are increased so that each learner is forced to focus on the misclassified

instances. Finally, a confidence value is given to each learner based on its predictive performance, and a new pattern is classified through a weighted voting mechanism that merges the prediction of all the weak learners. The most important parameter is *learning_rate*. It is within the interval $[0,1]$ and defines the step size while learning.

- **Extra-Trees (ET)** [12] is an ensemble classification method consisting in several tree classifiers trained independently. It performs like an ordinary RF, but additionally the top-down splitting in the tree learner is done randomly. Instead of determining the best split combination (based on, e.g., *entropy* or the *Gini impurity*) a random value is selected for the split. Similar to RF, the *max_depth* parameter represents the maximum depth of the tree, whereas *max_features* stands for the maximum feature number to be considered when looking for the split.

In our selection, we favoured algorithms with ensemble methods. This is due to the fact that classifiers of this kind can generally achieve better accuracy than other single classifiers [13] [5]. The performance measurements are conducted using the scikit-learn package library [8]. To take the advantage of parallel computations, we set the parameter *n_jobs* to -1, meaning that the computations are run simultaneously on all cores of the machine. All other parameters are set to their default values according to the scikit version 0.19.1.

5.2 Reference Dataset

5.2.1 Experimental Setup

The reference dataset[1] used in this study is obtained from the University of New South Wales (UNSW) [14] [15], and the same dataset was recently used by IBM Research and Cisco Systems in [16].

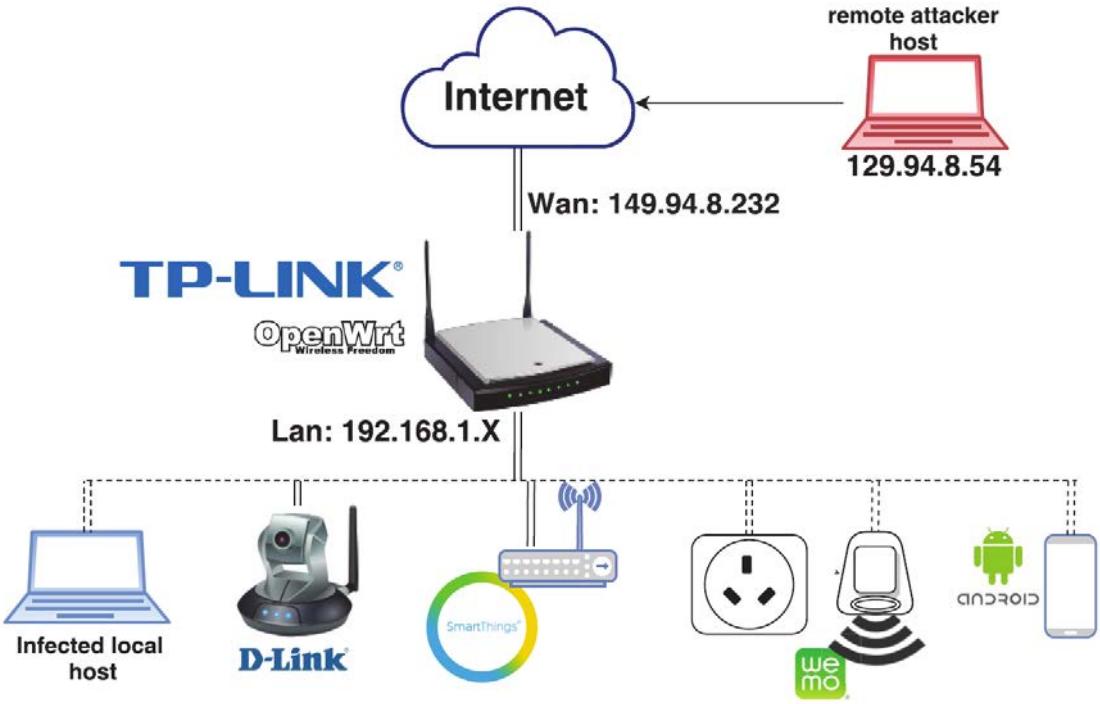


Figure 9: Testbed showing the IoT devices and gateway

Figure 9 [16] illustrates the testbed adopted by the conducted dataset experiment: A campus network was instrumented with a diversity of IoT devices. See Table 2 for more detailed information about the IoT devices. These devices include cameras, lights, activity sensors, health and well-being monitors. The TP Link router shown in Figure 9 serves as a gateway to the public Internet. It was flashed with

[1] http://149.171.189.1/

OpenWrt as well as some other packages to enable traffic capturing, i.e., including for instance the TCPDUMP tool. The WAN interface of the gateway is connected to the global Internet while the IoT devices are attached to the LAN/WLAN interfaces. The aim of this experiment is to passively gather traffic activities about all devices. These activities are collected over a period of three weeks and stored as PCAP files on an external hard drive attached to the gateway. We make use of these PCAP files to analyze relevant activities as well to extract informative features with unique patterns enabling us to accurately distinguish between the devices.

Category	Device	Wireless/Wired
Hubs	Smart Things Amazon Echo	Wired Wireless
Cameras	Netatmo Welcome TP-Link Day Night Cloud camera Samsung SmartCam Dropcam Insteon Camera Withings Smart Baby Monitor	Wireless Wireless Wireless Wireless Wired/Wireless Wired
Switches & Triggers	Belkin Wemo switch TP-Link Smart plug iHome Belkin wemo motion sensor	Wireless Wireless Wireless Wireless
Air quality sensors	NEST Protect smoke alarm Netatmo weather station	Wireless Wireless
Healthcare devices	Withings Smart scale Blipcare Blood Pressure meter Withings Aura smart sleep sensor	Wireless Wireless Wireless
Light Bulbs	Light Bulbs LiFX Smart Bulb	Wireless
Electronics	Triby Speaker PIX-STAR Photo-frame HP Printer	Wireless Wireless Wireless

Table 2: List of IoT devices in the smart campus environment

5.2.2 Feature Analysis and Selection

5.2.2.1 Dominant Protocol Analysis

The focus here lies on the application layer protocol as seen in Figure 10, where we examine the most used destination ports for TCP/UDP packets. We found that the port number distribution is not uniform across all packets. Some devices tend to send their packets more using well-known ports than another's port. We therefore decide to group the ports into ranges of system ports from 0 to 1023, registered ports from 1024 to 49151, and dynamic ports from 49152 to 65535. Based on this arrangement, we define feature $f_{range}(port)$ as a function of port as follows:

$$f_{range}(port) = \begin{cases} 1 & if\ port \in [0,1023] \\ 2 & if\ port \in [1024,49151] \\ 3 & if\ port \in [49152,65535] \\ 0 & otherwise \end{cases}$$

We afterwards concentrate on the range of [0,1023] due to the fact that system ports maintain the official list of well-known protocols. Zooming into this interval let us to find that TCP Port 443 -

indicative of HTTPS - is the most used TCP protocol by all devices. It has a rate of 72% by number of packets, meaning that the majority of packets contents in this range are encrypted and we cannot rely on the idea to infer sensitive data solely from payloads. The second dominant protocol is TCP Port 80 - representing HTTP - which constitutes 27% of TCP traffic. Our analysis shows also that TCP Ports 853/53, 445, 548 and 22 - indicative of DNS, SMB, AFP and SSH respectively - are present among the other used protocols. We see, on the other hand, UDP packets in the range of system ports as well. They are associated essentially to UDP Port 1900, indicative of the SSDP protocol. This protocol is used for advertising the presence of devices in the network as well as for discovering new services. It appears with a rate of 20% by numbers of UDP packets. Lastly, DNS and NTP packets are visible among the UDP packets using ports 53/853 and 123 respectively. The complete list of considered protocols is given in Table 3. This consists of 18 protocols we choose to define features. The output of each feature is boolean and can thus take either the value of 0 or 1. It is set to 1 if the protocol is used, otherwise it has the value of 0.

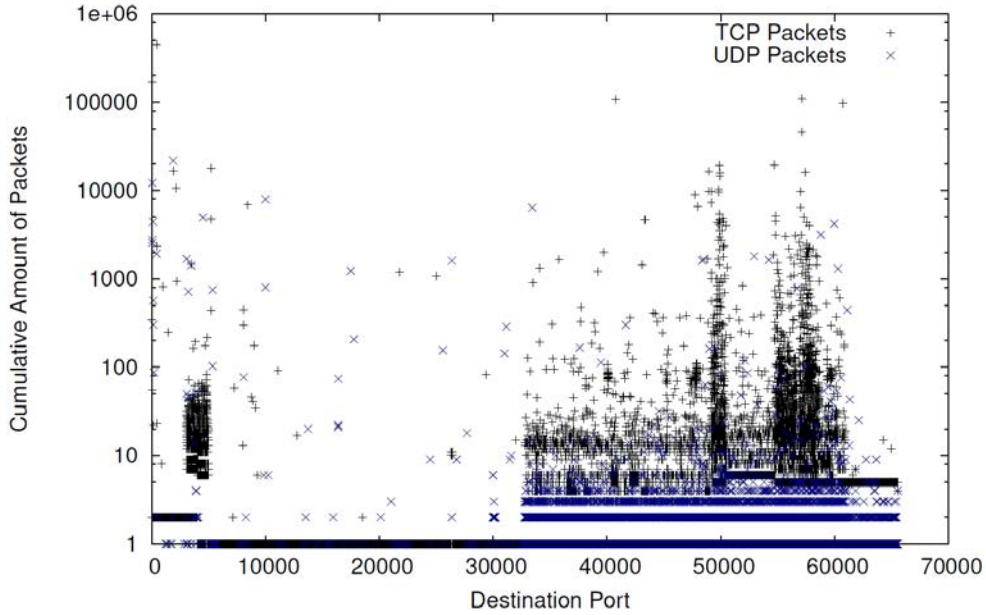


Figure 10: Correlation between used ports and amount of packets

OSI Layer	Protocol	Feature type
Application layer (10)	HTTP / HTTPS / DHCP SSDP / DNS / MDNS NTP / SMB / AFP / SSH	bool. (0/1) bool. (0/1) bool. (0/1)
Transport layer (2)	TCP / UDP	bool. (0/1)
Network layer (4)	IP / ICMP ICMPV6/IGMP	bool. (0/1) bool. (0/1)
Data link layer (2)	ARP / LLC	bool. (0/1)

Table 3: Considered protocols used for defining features

5.2.2.2 Statistical Analysis from Traffic Rates

There are additional features that can be used for identifying the devices. These include - but are not limited to - the mean and standard deviation of traffic volume as well as the inter-arrival time of packets. Regarding the latter attribute, we found that there is a unique pattern for some devices. For instance, inter-arrival times of 90, 60 and 20 seconds occur respectively for the HP Printer, iHome

switch, and the Netatmo Welcome. Note that such occurrence of inter-arrival times appear with a probability more than 70%. The other attribute we consider is the traffic volume.

Figure 11 shows our analysis in this regard, whereas the values on the x-axis represent the mean traffic volume of devices and their standard deviation is depicted on the y-axis. We found that the usage of both features combined achieves a good classification output, and the devices cluster well by their combination. For example, the mean and standard deviation of Belkin Wemo and Smart Things differ by almost an order of magnitude over the dataset period and thus helps obviously to distinguish between the devices.

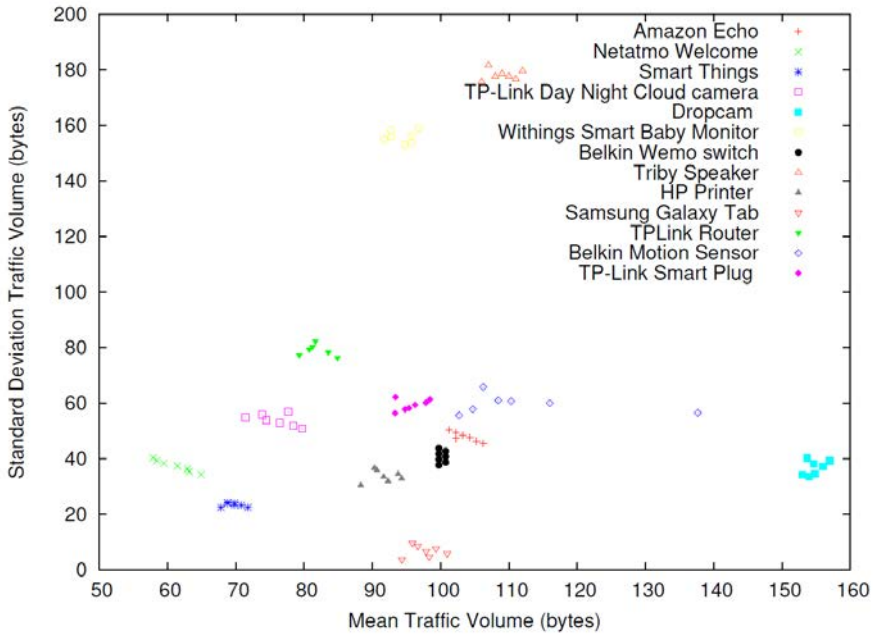


Figure 11: Mean and standard deviation of packet volume belonging to devices that are responsible for dominant protocol traffic

5.3 Evaluation

In the evaluation, a series of measurements have been conducted for the above analysed reference dataset in order to generate the final report. This subsection discusses the methodology used in this regard. It consists of the following two steps:

- **Exploring Evaluation**: In the first step, an exploring investigation process is performed to examine fundamental properties of different classifiers and their sensibility for certain tuning parameters.
- **Comparative Evaluation**: Based on the results of the exploring evaluation, a comparative predictive analysis for the classifiers is investigated according to a set of predictive performance metrics.

5.3.1 Exploring Evaluation

The first barrier to compare the classifiers are obviously their different tuning parameters (also known in the literature as *hyperparameters*). These influence the time used for making prediction as well as the overall accuracy. To be able to benchmark the classifiers fairly, we decide to explore the behavior of each one using different settings. In this study, we train the classifiers and tune parameters via *10-fold cross-validation* with the *RandomizedSearchCV* function from scikit-learn [17]. The *RandomizedSearchCV* is an estimator used for optimizing *hyperparameters* from a set of parameter settings. In contrast to *GridSearchCV*, not all parameter values are explored, but rather a fixed number of parameter settings is sampled from the specified distributions. We set 40% of the dataset as testing set and 60% as training set. The splitting is performed with a fixed random state. This has the benefit

that the experiments are reproducible and independent of any special properties of e.g. interference delay between cores. The parameter *n_iter* is set to 60. We also set *AUC* as *scoring metric* in *RandomizedSearchCV*. In the following, we report the achieved exploration results for each classifier:

- **KNN Best Parameter Results**
  - KNeighborsClassifier (algorithm = 'auto', leaf_size = 30, metric = 'minkowski', metric_params = None, n_jobs = 1, n_neighbors = 4, p = 2, weights = 'distance ')
  - Time taken for finding best estimator: 0.64s
  - Mean cross-validated score: 0.823529411765
- **SVM Best Parameter Results**
  - SVC (C = 5.40948345269, cache_size = 200, class_weight = None, coef0 = 0.0, decision_function_shape = 'ovr', degree = 3, gamma = 0.129798709295, kernel = 'rbf', max_iter = -1, probability = False, random_state = None, shrinking = True, tol = 0.001, verbose = False)
  - Time taken for finding best estimator: 0.93s
  - Mean cross-validated score: 0.529411764706
- **RF Best Parameter Results**
  - RandomForestClassifier (bootstrap = True, class_weight = None, criterion = 'gini', max_depth = 80, max_features = 'auto', max_leaf_nodes = None, min_impurity_decrease = 0.0, min_impurity_split = None, min_samples_leaf = 1, min_samples_split = 2, min_weight_fraction_leaf = 0.0, n_estimators = 400, n_jobs = 1, oob_score = False, random_state = None, verbose = 0, warm_start = False)
  - Time taken for finding best estimator: 87.66s
  - Mean cross-validated score: 0.852941176471
- **AB Best Parameter Results**
  - AdaBoostClassifier (algorithm = 'SAMME .R', base_estimator = None, learning_rate = 0.05, n_estimators = 100, random_state = None)
  - Time taken for finding best estimator: 4.98s
  - Mean cross-validated score: 0.911764705882
- **ET Best Parameter Results**
  - ExtraTreesClassifier (bootstrap = False, class_weight = None, criterion = 'gini', max_depth = 6, max_features = 0.7, max_leaf_nodes = None, min_impurity_decrease = 0.0, min_impurity_split = None, min_samples_leaf = 1, min_samples_split = 2, min_weight_fraction_leaf = 0.0, n_estimators = 50, n_jobs = 1, oob_score = False, random_state = None, verbose = 0, warm_start = False)
  - Time taken for finding best estimator: 3.43s
  - Mean cross-validated score: 0.897058823529

5.3.2 Comparative Evaluation

In the second step, the classifiers are tuned and trained according to best parameters they provided in the exploration phase. We evaluate their predictive performance by making use of five key performance indicators: precision, recall, F1 score, mean accuracy, and misclassification (see equations below).

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + \text{recall}}$$

TP denotes the number of devices whose types are correctly classified as positive (True Positive), FN is the number of devices whose types are incorrectly classified as negative (False Negative), TN

stands for the number of devices whose types are correctly classified as negative (True Negative), and FP denotes the number of devices whose types are incorrectly classified as positive (False Positive). Finally, the mean accuracy $(MAcc)$ is defined as the overall score of the predictions, whereas the Misclassification $(Misc)$ is calculated as the inverse probability of it:

$$MAcc = \frac{(precision + recall)}{2}$$

$$Misc = 1 - MAcc$$

5.3.3 Conclusion of the Results

Table 4 summarizes the main results of our proposed approaches in comparison to the baseline measurement. This baseline measurement is conducted using the *scikit-learn's DummyClassifier*. The metric selected for estimating its prediction is *most_frequent*. This means that the predictions in the baseline approach are always performed by considering the most frequent type of devices. From the results, it can be seen that all classifiers achieve improvements in terms of $(MAcc)$ over the baseline based on the testing set. Indeed, we can say that our extracted and selected features are informative for the device type identification problem, since the result attests a good predictive performance in all kind of classifiers. The strongest performance, 96% $(MAcc)$, is obtained using the learning model AB. This $(MAcc)$ offers a significant relative improvement of 79% compared to the baseline. When comparing however the time taken for optimizing parameters, it is clear that AB is not among the top 3 best classifiers. It needs about 4:98s for finding its best estimator. This demonstrates that the performance of AdaBoost is susceptible to the time needed for optimizing tuning parameters, which is consistent with the results obtained by the previous researchers [18].

		Key Performance Indicators (KPIs) [%]						
		MAcc	Misc	Precision	Recall	F_1-score	Baseline	Improvement
Learning Models	KNN	78%	22%	78%	78%	78%	17%	61%
	SVM	39%	61%	39%	39%	39%	17%	22%
	RF	78%	22%	78%	78%	78%	17%	61%
	AB	96%	4%	96%	96%	96%	17%	79%
	ET	78%	22%	78%	78%	78%	17%	61%

Table 4: Performance comparison of the proposed classification models with the baseline

5.4 Integration

This subsection describes the ETA module integrated in the F-interop platform. It consists of a step by step user guide illustrating the test steps needed to perform the encrypted traffic analysis. In the current implementation of the F-Interop platform, the process of ETA involves, among others, the interaction and cooperation of three actors, namely F-Interop User, GUI, and ETA module (see Figure 12). The basic interaction flow of ETA testing process consists of the following five steps:

1. The F-Interop User initiates the process by specifying the desired test case selection (in our case -- ETA), as described in **Erreur ! Source du renvoi introuvable.**, step 2.
2. The User uploads the PCAP file containing the sniffed encrypted packets belonging to the device under test.
3. The GUI forwards the PCAP file to the ETA module.

4. The ETA module merges the obtained PCAP with the reference dataset in order to generate one single PCAP file on which the analysis is conducted.
5. The ETA testing is performed on the merged dataset and the privacy analysis results are displayed in the GUI.

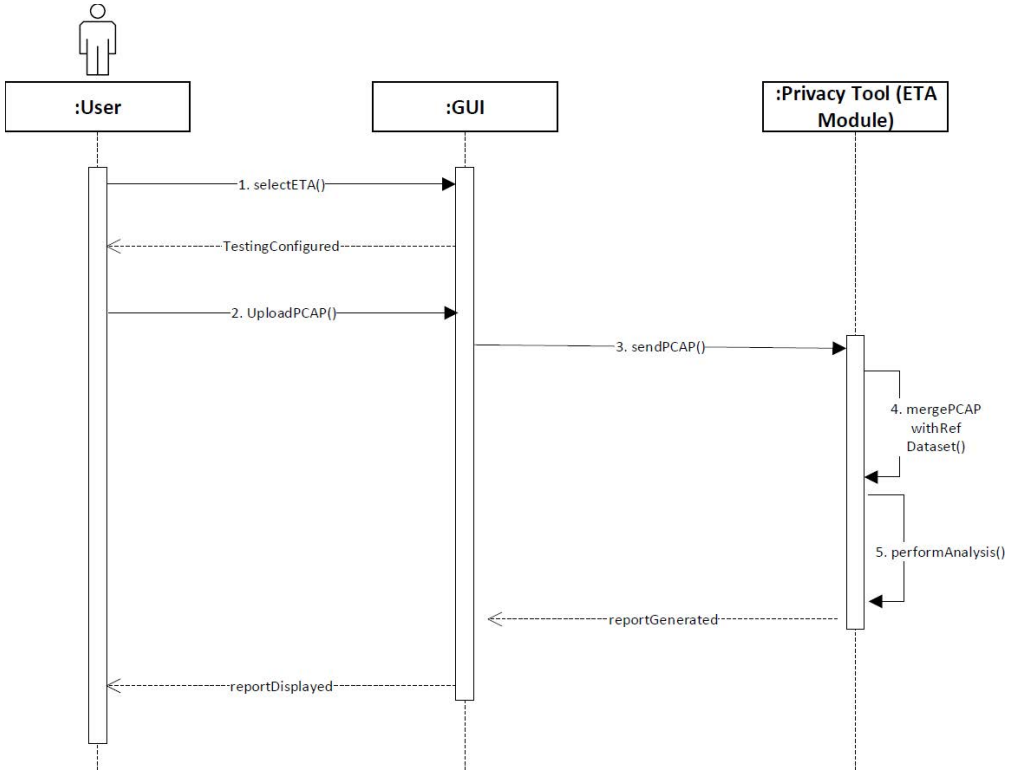


Figure 12 Interaction Flow Diagram of ETA

The added value of the ETA module can be better explained with the following example. Let's assume an F-Interop user has one device that he/she wants to test for privacy issues, called *Device under Test* (or DuT for short). The User starts a new ETA testing session, as shown in **Erreur ! Source du renvoi introuvable.**. Then, the system will change the testing context and will allow the uploading of a PCAP file containing the sniffed encrypted packets belonging to this DuT. The file is sent to the ETA module, which performs the encrypted traffic analysis and generates a report, as shown in **Erreur ! Source du renvoi introuvable.**.

The ETA analysis report contains the privacy assessment result indicating the identification accuracy level obtained with different classifiers. It shows how well an attacker is able to identify the DuT. The first column "Classifier" reveals the type of classifier used for DuT identification. The second column "Identification Accuracy" depicts the level of accuracy found with each classifier. The accuracy values range always within [0,1]. An accuracy value of 0 means that the DuT cannot be identified at all, while a value of 1 means that the DuT can be identified with 100% accuracy. In this particular example (illustrated in **Erreur ! Source du renvoi introuvable.**), we achieved the following classifier results:

- KNN with an identification accuracy of 0.67 (67%)
- SVM with an identification accuracy of 0.33 (33%)
- AB with an identification accuracy of 1.0 (100%)
- ET with an identification accuracy of 1.0 (100%)
- RF with an identification accuracy of 1.0 (100%)

It can be seen that most of classifiers (AB, ET, and RF) can indeed achieve an accuracy of 100% for identifying this particular DuT. This means that its communication is not secure against passive monitoring attacks.

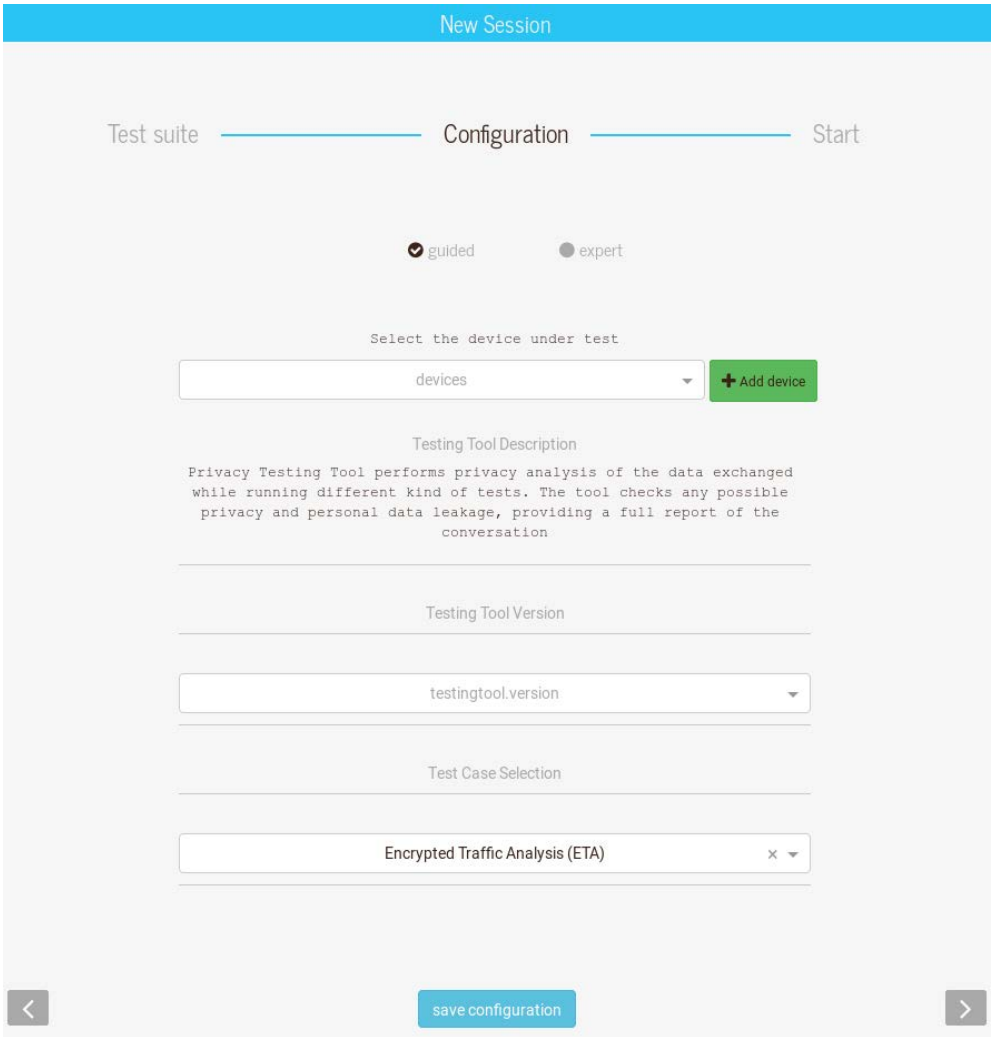


Figure 13 ETA Test Selection

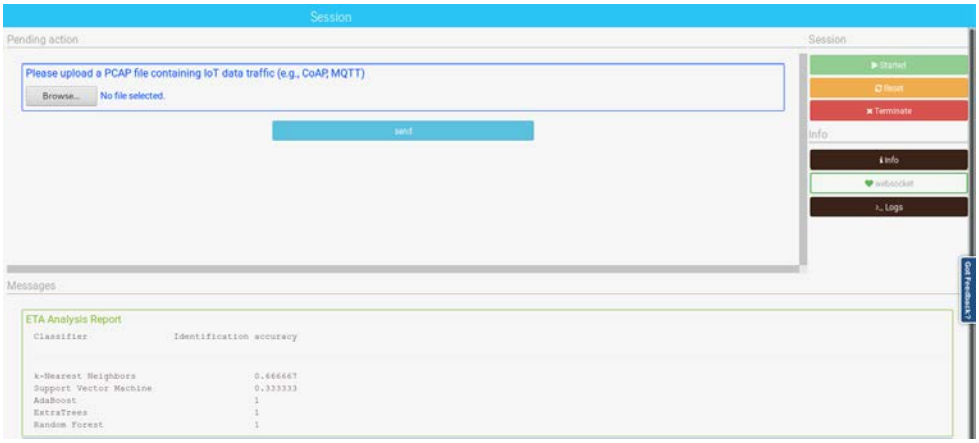


Figure 14 Example of an ETA Analysis Report

6 Conclusion

Task 3.2 is responsible of designing methods for privacy analysis of the data exchanged while running different kind of tests on the F-Interop platform. This document described the final iteration of the Privacy Test Tool, of the testing tools available in the F-Interop platform.

The Privacy Test Tool is able to detect privacy issues by analysing both *encrypted* and *non-encrypted* data traffic of IoT protocols. The tool is composed of two main modules: the **Encrypted Traffic Analysis (ETA)** module, the newly added module in the final iteration and the **Non-encrypted Traffic Analysis (NTA)** module, which is introduced in the previous iteration in D3.3. ETA is able to investigate how an adversary can get sensitive information related to IoT device activities by passively observing patterns of encrypted communication. NTA follows a *pattern matching* approach in the data payload of IoT protocols in order to detect what is considered *personal* and/or *private*.

This deliverable provided a detailed description of the ETA module, which has been recently integrated into the Privacy Test Tool. It also presented some update of the NTA module, as well as the achievements with respect to the general architecture of the tool itself.

7 References

[1] International Organization for Standardization, " ISO/IEC 20922:2016 Information technology -- Message Queuing Telemetry Transport (MQTT) v3.1.1," ISO, 2016.

[2] S. Ziegler, S. Fdida, T. Watteyne and C. Viho, "F-Interop - Online Conformance, Interoperability and Performance Tests for the IoT," in *{Conference on Interoperability in IoT (InterIoT)}*, Paris, 2016.

[3] "Privacy Test Tool: Encrypted Traffic Analysis Module," [Online]. Available: https://gitlab.f-interop.eu/f-interop/privacy_testing_tool.

[4] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," in *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, Amsterdam, The Netherlands, The Netherlands, 2007.

[5] T. G. Dietterich, "Ensemble Methods in Machine Learning," in *Multiple Classifier Systems*, Berlin, 2000.

[6] M. Woniak, M. Gra\ {n}a and E. Corchado, "A Survey of Multiple Classifier Systems As Hybrid Systems," *Inf. Fusion,* vol. 16, pp. 3-17, 3 2014.

[7] M.-L. Zhang and Z.-H. Zhou, "A k-nearest neighbor based algorithm for multi-label classification," in *2005 IEEE International Conference on Granular Computing*, 2005.

[8] A. Gron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 1st ed., O'Reilly Media, Inc., 2017.

[9] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications,* vol. 13, pp. 18-28, 7 1998.

[10] M. S. Alam and S. T. Vuong, "Random Forest Classification for Detecting Android Malware," in *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 2013.

[11] T. Jan, "Ada-Boosted Locally Enhanced Probabilistic Neural Network for IoT Intrusion Detection," in *Complex, Intelligent, and Software Intensive Systems*, Cham, 2019.

[12] P. Geurts, D. Ernst and L. Wehenkel, "Extremely randomized trees," *Machine Learning,* vol. 63, pp. 3-42, 01 4 2006.

[13] R. Maclin and D. W. Opitz, "Popular Ensemble Methods: An Empirical Study," *CoRR,* vol. abs/1106.0257, 2011.

[14] F. Loi, A. Sivanathan, H. H. Gharakheili, A. Radford and V. Sivaraman, "Systematically Evaluating Security and Privacy for Consumer IoT Devices," in *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, New York, NY, USA, 2017.

[15] A. Sivanathan, D. Sherratt, H. H. Gharakheili, V. Sivaraman and A. Vishwanath, "Low-cost flow-based security solutions for smart-home IoT devices," in *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2016.

[16] A. Sivanathan, D. Sherratt, H. H. Gharakheili, A. Radford, C. Wijenayake, A. Vishwanath and V. Sivaraman, "Characterizing and classifying IoT traffic in smart cities and campuses," in *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2017.

[17] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt and G.

Varoquaux, “API design for machine learning software: experiences from the scikit-learn project,” *CoRR,* vol. abs/1309.0238, 2013.

[18] R. Bardenet, M. Brendel, B. K{\'e}gl and M. Sebag, “Collaborative Hyperparameter Tuning,” in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, Atlanta, 2013.