Evaluation

We evaluated our approach by using two incident scenarios that are publicly available[1] for research and training purposes. The data-sets associated with the scenarios are big enough to be realistic, but small enough to allow us to examine the evidence on a single computer. For each incident we modelled the environment and the speculative hypotheses, and generated the corresponding forensic-readiness specification. For each incident we measure the time necessary to generate the specification and the amount of events that our approach prescribes to preserve, and check whether these events can explain how the incident occurred. The rest of the document provides details about how the 2 scenarios were modeled and how we used these scenarios to evaluate our approach.

University Harassment Scenario

The university harassment scenario is inspired by the Nitroba University Harassment Scenario[2], which describes the incident of an academic (Lily Tuckrige) who received harassment emails from one of her students attending the chemistry course.

We considered our scenario to be set in a university, where students and academic staff can send emails by using the university and students' residence internal network. We assumed the routers as available evidence sources. An incident of concern is related to the receipt of harassment emails from academics.

In the model of the environment, we represented a possible to set of complex events. These include: (i) email sending; (ii) cookie setting from an external address to an IP; (iii) sending of messages from an IP address and a browser agent; (iv) sending of anonymous email from an IP address and a browser agent.

The following speculative hypotheses were constructed:

- *h1*: an email is sent to an academic by someone using an external address;
- *h2*: an anonymous email is sent by an individual who can be identified;
- *h3*: an anonymous email is sent by an individual who cannot be identified.

An individual sending an anonymous email can be identified if she has previously exchanged a message by using the same browser agent and IP address from which the anonymous email was sent. Furthermore, the IP address should also be the recipient of a set-cookie message referring to the email address of the individual.

An individual cannot be identified if the browser agent adopted to send the anonymous email was associated with more than one IP address, meaning that more than one individual used the same browser. An individual might also not be identified if the IP address from which the anonymous email was sent cannot be associated with a cookie referring to the email address of the individual.

A representation of the environment and the hypotheses of the example has been made available at https://github.com/lpasquale/minorityReport/tree/master/HarassmentScenario. The folder also has a README file providing explanations about how to run the example scenario. Examples of negative refuting histories used for hypothesis h2 can be found in file *neg-h2.txt*. The specification generated for each hypothesis is also provided in folder *spec*. For example, *spec-h1.txt* provides the specification generated for hypothesis h1. The same applies to the other hypotheses.

The specification generated for h1 is the following (*spec-h1.txt*):

[1] http://digitalcorpora.org/

[2] http://digitalcorpora.org/corpora/scenarios/nitroba-university-harassment-scenario

```
rtrig(preserve(send_fake_email(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens(receive(send_fake_email(V1,V2,V3,V4,V5),V6),V6,V7),email(V1),
    faddr(V2),addr(V3),ip(V4),rout(V5),clock(V6),trace(V7).
```

Event *send_fake_email* is preserved every time the FR Controller receives it.

The specification generated for h2 is the following (*spec-h2.txt*):

```
rtrig(preserve(set_cookie(V1,V2,V3,V4),V5),V5,V6):-
    happens(receive(set_cookie(V1,V2,V3,V4),V5),V5,V6),addr(V1),
    sendIp(V2),recIp(V3),rout(V4),clock(V5),trace(V6).
```

Event *set_cookie* is preserved every time the FR Controller receives it.

```
rtrig(preserve(send_anon_email(V1,V2,V3,V4),V5),V5,V6):-
    happens_prev(preserve(exchange_msg(V1,V2,V3,V7),V5),V5,V6),
    happens(receive(send_anon_email(V1,V2,V3,V4),V5),V5,V6),browser(V1),
    sendIp(V2),recIp(V3),rout(V7),clock(V5),trace(V6),rout(V4).
```

Event *send_anon_email* is preserved if the FR Controller receives it and event *exchange_msg* from the same IP address (`V2`) from which the anonymous email was sent and using the same browser agent (`V1`) has been previously preserved.

```
rtrig(preserve(exchange_msg(V1,V2,V3,V4),V5),V5,V6):-
    happens_prev(preserve(set_cookie(V7,V3,V2,V4),V5),V5,V6),
    happens(receive(exchange_msg(V1,V2,V3,V4),V5),V5,V6),addr(V7),
    recIp(V3),sendIp(V2),rout(V4),clock(V5),trace(V6),browser(V1).
```

Event *exchange_msg* is preserved if the FR Controller receives it and event *set_cookie* referring to an email address (`V7`) and having as a recipient address the same sender address (`V2`) as the *exchange_msg* event has been previously preserved.

The specification generated for h3 is the following (*spec-h3.txt*):

```
rtrig(preserve(send_anon_email(V1,V2,V3,V4),V5),V5,V6):-
    happens(receive(send_anon_email(V1,V2,V3,V4),V5),V5,V6),browser(V1),
    sendIp(V2),recIp(V3),clock(V5),trace(V6),rout(V4).
```

Event *send_anon_email* is preserved if the FR Controller receives it.

Table I shows the total time necessary to synthesise a specification for each hypothesis, as well as the time required by each phase of the approach: hypotheses generation (*HG*), specification verification (SV) and specification synthesis (*SS*). For each hypothesis, we show the number of supporting histories (out of the total number generated) and negative histories necessary to compute a specification, including the maximum length of the histories. The maximum time was taken for the most complex hypothesis (*h2*), which also required the provision of 4 negative histories.

Table I. University Harassment Scenario Performance

	Instances			Execution time (s)			
	#Pos	**#Neg**	**Length**	**HG**	**SV**	**SS**	**Total**
h1	1/3	0	1	~0	0.01	0.23	0.24
h2	1/32	4	3	0.08	0.19	39.913	40.183
h3	1/8	0	1	0.01	0.03	0.301	0.341

To investigate how many event entries our approach would avoid the need to examine for a harassment case, we assume that full data-set would be available to an investigator. This data-set includes the TCP packets captured from the routers located inside the students' residence — *577,760* data streams (application level messages) exchanged in *15,508* communications between different IP addresses. It can be downloaded from the University Harassment Scenario webpage (http://digitalcorpora.org/corp/nps/packets/2008-nitroba/nitroba.pcap).
We implemented the specification of a forensic-readiness controller able to extract data from the TCP traffic dataset.

The implementation we built preserves 4 types of events corresponding to the primitive events we modeled in the environment description.

- **SUE***:* SMTP traffic related to emails sent from external addresses to an academic. This corresponds to primitive event *send_fake_email* defined in the environment description.
- **SC:** incoming HTTP traffic adopted to set-up a cookie. This corresponds to primitive event *set_cookie* defined in the environment description.
- **EM:** outgoing HTTP traffic referring to the browser agent adopted. This corresponds to primitive event *exchange_msg* in the environment description
- **SAE:** outgoing HTTP traffic to send anonymous emails. This corresponds to primitive event *send_anon_email* in the environment description.

The specification was implemented as two bash scripts (*spec-implementation-h1.sh* and *spec-implementation-h2-h3.sh*). The former extracts event SUE necessary to support h1, while the latter preserves events SC, EM and SAE, necessary to support h2 and h3.
The specification implementation extracts events from the network traffic included in the file *nitroba.pcap;* to run the bash scripts it is necessary to have *tshark* installed.

The implemented specification preserves each event type in a different file. In particular, 3 files were created: *set_cookie*, *exchange_msg* and *send_anon_email* to preserve occurrences of the homonymous primitive events. No occurrence of *send_fake_email* was preserved since the data-set does not contain SMTP traffic. The bash scripts and their outputs are at https://github.com/lpasquale/minorityReport/tree/master/HarassmentScenario/spec_implementation

The number and type of event entries preserved by the controller for each hypothesis is shown in Table II; the total number of event entries is 0.54% of the data streams that an investigator would examine from the data-set.

Table II. Number of events preserved in forensic-ready case.

		SUE	SC	EM	SAE
# Events	h1	0	---	---	---
	h2	---	2	3830	300
	h3	---	---	---	300
		Total: 3130 events			

The application shows that h2 might be satisfied since an incoming set-cookie message associated with *jcoach@gmail.com* and received by IP *192.168.015.004* was preserved. Outgoing HTTP messages from the same IP address are associated with a Mozilla browser have also been recorded; the same browser appears to have been used to send the anonymous email. This means that our approach would preserve necessary evidence if such an incident were to occur. This would support investigators in prioritising their efforts, while ensuring that other evidence related to alternative scenarios would be preserved if such scenarios occurred.

Corporate Exfiltration Scenario

The exfiltration scenario is inspired by the M57-Jean scenario[3], which describes the case of document exfiltration, in which a spreadsheet containing confidential information was posted as an attachment in the technical support forum of a competitor's website. The spreadsheet was stored in the CFO's (Jean's) computer.

We considered our scenario to be set in a corporate organisation characterised by its employees, their email addresses, their access rights to computers, and any operation they can perform on them (e.g., email sending/receiving, mounting of storage devices, programs installation). We assume computers (Windows) hard drives as available evidence sources. An incident of concern is related to the exfiltration of the confidential spreadsheet (*m57plan*).

In the model of the environment, we represented a possible set of complex events. These include: (i) email sending which have attachments; (ii) email receipt; (iii) emails exchanges (an email received from an address is subsequently answered with an attachment); (iv) login/logout of an employee to/from a computer; (v) mount/unmount of an external storage device to/from a computer; (vi) copy of a file by an employee on a computer; (vii) install/uninstall of an application on/from a computer; (viii) HTTP connection to a URL from a browser.

The following speculative hypotheses were constructed:

- *h1*: an email with file *m57plan* as attachment is sent by an employee (P1) to another one (P2), without P2 having previously sent emails to P1 (for example, to request the document).
- *h2*: an email with file *m57plan* as attachment is sent by an employee (P1) to an external address, without P1 having previously received emails from this external address (for example, to request the document).
- *h3*: an email previously received by an employee from another employee is subsequently answered by also including file *m57plan* as attachment.
- *h4*: an email previously received by an employee from an external address is subsequently answered by also including file *m57plan* as attachment.
- *h5*: file *m57plan* is copied by an employee while an external storage device is mounted.

[3] http://digitalcorpora.org/corpora/scenarios/m57-jean

- *h6*: a malware that might have silently copied file *m57plan* is installed on a computer in which the document is stored, after an employee accessed to a blacklisted URL.

A representation of the environment and the hypotheses of the example is available at https://github.com/lpasquale/minorityReport/tree/master/ExfiltrationScenario. The folder also has a README file providing explanations about how to run the example scenario. Examples of positive and negative refuting histories used for each hypothesis are also provided. For example, for hypothesis h1, positive histories are in file *pos-h1.txt* and negative histories are in *neg-h1.txt*. The same applies to the other hypotheses. The specification generated for each hypothesis is also provided in folder *spec*. For example, *spec-h1.txt* provides the specification generated for hypothesis h1. The same applies to the other hypotheses.

The specification generated for h1 is the following (*spec-h1.txt*):

```
rtrig(preserve(sys_login(V1,V2,V3),V4),V4,V5):-
    happens(receive(sys_login(V1,V2,V3),V4),V4,V5),emp(V1),pwd(V2),
    comp(V3),clock(V4),trace(V5).
```

Event *sys_login* is preserved every time the FR Controller receives it.

```
rtrig(preserve(send_email_e2e(V1,V2,V3,V3,V4),V5),V5,V6):-
    happens(receive(send_email_e2e(V1,V2,V3,V3,V4),V5),V5,V6),
    happens_prev(preserve(sys_login(V7,V8,V4),V5),V5,V6),email(V1),fi(V2),
    sendAddr(V3),comp(V4),clock(V5),trace(V6),emp(V7),pwd(V8).
```

Event *send_email_e2e n* is preserved every time the FR Controller receives it and event *sys_login* was preserved previously.

```
rtrig(preserve(send_email_ext2e(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens_prev(preserve(sys_login(V8,V9,V5),V6),V6,V7),
    happens(receive(send_email_ext2e(V1,V2,V3,V4,V5),V6),V6,V7),
    emp(V8),pwd(V9),comp(V5),clock(V6),trace(V7),email(V1),fi(V2),
    extAddr(V3),addr(V4).
```

Event *send_email_e2e n* is preserved every time the FR Controller receives it and event *sys_login* was preserved previously.

The specification generated for h2 is the following (*spec-h2.txt*):

```
rtrig(preserve(sys_login(V1,V2,V3),V4),V4,V5):-
    happens(receive(sys_login(V1,V2,V3),V4),V4,V5),emp(V1),pwd(V2),
    comp(V3),clock(V4),trace(V5).
```

Event *sys_login* is preserved every time the FR Controller receives it.

```
rtrig(preserve(send_email_e2ext(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens_prev(preserve(sys_login(V8,V9,V5),V6),V6,V7),
    happens(receive(send_email_e2ext(V1,V2,V3,V4,V5),V6),V6,V7),
    emp(V8),pwd(V9),comp(V5),clock(V6),trace(V7),email(V1),
    fi(V2),addr(V3),extAddr(V4).
```

Event *send_email_e2ext* is preserved every time the FR Controller receives it and event *sys_login* was preserved previously.

```
rtrig(preserve(send_email_ext2ext(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens_prev(preserve(sys_login(V8,V9,V5),V6),V6,V7),
    happens(receive(send_email_ext2ext(V1,V2,V3,V4,V5),V6),V6,V7),
    emp(V8),pwd(V9),comp(V5),clock(V6),trace(V7),email(V1),fi(V2),
    extAddr(V3), extAddr(V4).
```

Event *send_email_ext2ext* is preserved every time the FR Controller receives it and event *sys_login* was preserved previously.

The specification generated for h3 is the following (*spec-h3.txt*):

```
rtrig(preserve(sys_login(V1,V2,V3),V4),V4,V5):-
    happens(receive(sys_login(V1,V2,V3),V4),V4,V5),emp(V1),
    pwd(V2),comp(V3),clock(V4),trace(V5).
```

Event *sys_login* is preserved every time the FR Controller receives it.

```
rtrig(preserve(rec_email_req_e2e(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens_pred(preserve(sys_login(V8,V9,V5),V6),V6,V7),
    happens(receive(rec_email_req_e2e(V1,V2,V3,V4,V5),V6),V6,V7),emp(V8),
    pwd(V9),comp(V5),clock(V6),trace(V7),email(V1),fi(V2),
    sendAddr(V3),recAddr(V4).
```

Event *rec_email_req_e2e* is preserved every time the FR Controller receives it and event *sys_login* was preserved previously.

```
rtrig(preserve(send_email_e2e(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens_prev(preserve(rec_email_req_e2ext(V1,V2,V4,V8,V5),V6),V6,V7),
    happens(receive(send_email_e2e(V1,V2,V3,V4,V5),V6),V6,V7),
    email(V1),fi(V2),recAddr(V4),extAddr(V8),comp(V5),clock(V6),
    trace(V7),sendAddr(V3).
```

Event *send_email_e2e* is preserved every time the FR Controller receives it and event *rec_email_req_e2ext* is preserved previously.

```
rtrig(preserve(send_email_e2e(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens_prev(preserve(rec_email_req_e2e(V1,V2,V4,V3,V5),V6),V6,V7),
    happens(receive(send_email_e2e(V1,V2,V3,V4,V5),V6),V6,V7),email(V1),
    fi(V2),recAddr(V4),sendAddr(V3),comp(V5),clock(V6),trace(V7).
```

Event *send_email_e2e* is preserved every time the FR Controller receives it and event *rec_email_req_e2e* is preserved previously.

```
rtrig(preserve(rec_email_req_e2ext(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens_pred(preserve(sys_login(V8,V9,V5),V6),V6,V7),
    happens(receive(rec_email_req_e2ext(V1,V2,V3,V4,V5),V6),V6,V7),
    emp(V8),pwd(V9),comp(V5),clock(V6),trace(V7),email(V1),fi(V2),
    addr(V3),extAddr(V4).
```

Event *rec_email_req_e2ext* is preserved every time the FR Controller receives it and event *sys_login* is preserved previously.

```
rtrig(preserve(send_email_ext2e(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens_prev(preserve(rec_email_req_e2ext(V1,V2,V4,V3,V5),V6),V6,V7),
    happens(receive(send_email_ext2e(V1,V2,V3,V4,V5),V6),V6,V7),
    email(V1),fi(V2),addr(V4),extAddr(V3),comp(V5),clock(V6),trace(V7).
```

Event *send_email_ext2e* is preserved every time the FR Controller receives it and event *rec_email_req_e2ext* is preserved previously.

```
rtrig(preserve(send_email_ext2e(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens_prev(preserve(rec_email_req_e2e(V1,V2,V4,V8,V5),V6),V6,V7),
    happens(receive(send_email_ext2e(V1,V2,V3,V4,V5),V6),V6,V7),email(V1),
    fi(V2),addr(V4),recAddr(V8),comp(V5),clock(V6),trace(V7),extAddr(V3).
```

Event *send_email_ext2e* is preserved every time the FR Controller receives it and event *rec_email_req_e2e* is preserved previously.

The specification generated for h4 is the following (*spec-h4.txt*):

```
rtrig(preserve(sys_login(V1,V2,V3),V4),V4,V5):-
    happens(receive(sys_login(V1,V2,V3),V4),V4,V5),emp(V1),
    pwd(V2),comp(V3),clock(V4),trace(V5).
```

Event *sys_login* is preserved every time the FR Controller receives it.

```
rtrig(preserve(rec_email_req_ext2e(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens_pred(preserve(sys_login(V8,V9,V5),V6),V6,V7),
    happens(receive(rec_email_req_ext2e(V1,V2,V3,V4,V5),V6),V6,V7),
    emp(V8),pwd(V9),comp(V5),clock(V6),trace(V7),email(V1),fi(V2),
    extAddr(V3),addr(V4).
```

Event *rec_email_req_e2ext* is preserved every time the FR Controller receives it and event *sys_login* is preserved previously.

```
rtrig(preserve(rec_email_req_ext2ext(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens(receive(rec_email_req_ext2ext(V1,V2,V3,V4,V5),V6),V6,V7),
    happens_pred(preserve(sys_login(V8,V9,V5),V6),V6,V7),email(V1),
    fi(V2),sendExtAddr(V3),recExtAddr(V4),comp(V5),clock(V6),trace(V7),
    emp(V8),pwd(V9).
```

Event *rec_email_req_ext2ext* is preserved every time the FR Controller receives it and event *sys_login* is preserved previously.

```
rtrig(preserve(send_email_e2ext(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens_prev(preserve(rec_email_req_ext2ext(V1,V2,V4,V8,V5),V6),
    V6,V7),
    happens(receive(send_email_e2ext(V1,V2,V3,V4,V5),V6),V6,V7),
    email(V1),fi(V2),extAddr(V4),recExtAddr(V8),comp(V5),clock(V6),
    trace(V7),addr(V3).
```

Event *send_email_e2e* is preserved every time the FR Controller receives it and event *rec_email_req_ext2ext* is preserved previously.

```
rtrig(preserve(send_email_ext2ext(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens_prev(preserve(rec_email_req_ext2e(V1,V2,V4,V8,V5),V6),V6,V7),
    happens(receive(send_email_ext2ext(V1,V2,V3,V4,V5),V6),V6,V7),
    email(V1),fi(V2),recExtAddr(V4),addr(V8),comp(V5),clock(V6),
    trace(V7),sendExtAddr(V3).
```

Event *send_email_ext2ext* is preserved every time the FR Controller receives it and event

rec_email_req_ext2e is preserved previously.

```
rtrig(preserve(send_email_ext2ext(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens_pred(preserve(rec_email_req_ext2ext(V1,V2,V4,V3,V5),V6),
    V6,V7),
    happens(receive(send_email_ext2ext(V1,V2,V3,V4,V5),V6),V6,V7),
    email(V1),fi(V2),recExtAddr(V4),sendExtAddr(V3),comp(V5),clock(V6),
    trace(V7).
```

Event *send_email_ext2ext* is preserved every time the FR Controller receives it and event *rec_email_req_ext2ext* is preserved previously.

```
rtrig(preserve(send_email_e2ext(V1,V2,V3,V4,V5),V6),V6,V7):-
    happens_prev(preserve(rec_email_req_ext2e(V1,V2,V4,V3,V5),V6),V6,V7),
    happens(receive(send_email_e2ext(V1,V2,V3,V4,V5),V6),V6,V7),email(V1),
    fi(V2),extAddr(V4),addr(V3),comp(V5),clock(V6),trace(V7).
```

Event *send_email_e2ext* is preserved every time the FR Controller receives it and event *rec_email_req_ext2e* is preserved previously.

The specification generated for h5 is the following (*spec-h5.txt*):

```
rtrig(preserve(sys_login(V1,V2,V3),V4),V4,V5):-
    happens(receive(sys_login(V1,V2,V3),V4),V4,V5),emp(V1),
    pwd(V2),comp(V3),clock(V4),trace(V5).
```

Event *sys_login* is preserved every time the FR Controller receives it.

```
rtrig(preserve(sys_mount(V1,V2),V3),V3,V4):-
    happens_prev(preserve(sys_login(V5,V6,V2),V3),V3,V4),
    happens(receive(sys_mount(V1,V2),V3),V3,V4),emp(V5),pwd(V6),
    comp(V2),clock(V3),trace(V4),st(V1).
```

Event *sys_mount* is preserved every time the FR Controller receives it and event *sys_login* is preserved previously.

```
rtrig(preserve(sys_copy(V1,V2,V3),V4),V4,V5):-
    happens_prev(preserve(sys_mount(V6,V3),V4),V4,V5),
    happens_prev(preserve(sys_login(V1,V7,V3),V4),V4,V5),st(V6),
    comp(V3),clock(V4),trace(V5),emp(V1),pwd(V7).
```

Event *sys_copy* is preserved every time the FR Controller receives it and event *sys_mount* and *sys_login* are preserved previously.

The specification generated for h6 is the following (*spec-h6.txt*):

```
rtrig(preserve(sys_login(V1,V2,V3),V4),V4,V5):-
    happens(receive(sys_login(V1,V2,V3),V4),V4,V5),emp(V1),
    pwd(V2),comp(V3),clock(V4),trace(V5).
```

Event *sys_login* is preserved every time the FR Controller receives it.

```
rtrig(preserve(web_connection(V1,V2,V3),V4),V4,V5):-
    happens_prev(preserve(sys_login(V6,V7,V3),V4),V4,V5),
    happens(receive(web_connection(V1,V2,V3),V4),V4,V5),emp(V6),
    pwd(V7),comp(V3),clock(V4),trace(V5),browser(V1),url(V2).
```

Event *web_connection* is preserved every time the FR Controller receives it and event *sys_login* is preserved previously.

```
rtrig(preserve(sys_install(V1,V2),V3),V3,V4):-
    happens(receive(sys_install(V1,V2),V3),V3,V4),
    happens_pred(preserve(web_connection(V5,V6,V2),V3),V3,V4),
    app(V1),comp(V2),clock(V3),trace(V4),browser(V5),url(V6).
```

Event *sys_install* is preserved every time the FR Controller receives it and event *web_connection* is preserved previously.

Compared to the previous scenario, the corporate exfiltration scenario required us to model more complex hypotheses for which a higher number of supportable and refutable histories existed. This increased the time the approach took to learn a specification.

Table III. Performance for the Corporate Exfiltration Scenario

	Instances			Execution time (s)			
	#Pos	**#Neg**	**Length**	**HI**	**SV**	**SG**	**Total**
h1	2 / 12	2	2	0.01	0.05	7.756	7.816
h2	1 / 4	1	2	0.01	0.05	1.852	1.912
h3	4 / 18	14	3	0.1	0.28	1733.82	1734.2
h4	4 / 16	9	3	0.5	0.18	894.197	894.877
h5	1	4	3	0.05	0.2	43.851	44.101
h6	1	4	3	0.5	0.21	170.356	171.066

Table III shows the total time necessary to synthesise a specification for each hypothesis, as well as the time required by each phase of the approach: hypotheses generation (*HG*), specification verification (SV) and specification synthesis (*SS*). For each hypothesis, we show the number of supporting histories (out of the total number generated) and negative histories necessary to compute a specification, including the maximum length of the histories. The maximum time was taken for the most complex hypothesis (*h3* and *h4*), which required the provision of 4 positive histories and 18 and 16 negative histories respectively.

To investigate how many event entries our approach would avoid the need to examine for the corporate exfiltration scenario case, we assume that the full data-set would be available to an investigator. The dataset provides a copy of Jean's Windows computer hard drive (~1.46GB). The dataset can be downloaded from the M27-Jean Scenario webpage at http://digitalcorpora.org/corpora/drives/nps-2008-m57-jean/nps-2008-jean.E01 and http://digitalcorpora.org/corpora/drives/nps-2008-m57-jean/nps-2008-jean.E02.

To examine the available hard drive we acquired the image through Autopsy[4] and counted the number of primitive events that an investigator would have to examine to demonstrate or refute the hypotheses described above if the system is not forensic ready. Due to the nature of the available data-set, the events that are identified through Autopsy are only those that can be identified as modifications of the file system. This is because the data-set available has been acquired after the incident took place and data have not been preserved proactively. For this reason, not all primitive events modeled in the environment representation can be extracted from the data-set. The events extracted from the data-set are the following:

- **Last User Login** that corresponds to primitive event *sys_login* defined in the environment description.
- **Received emails** that might correspond to primitive events:
  - *rec_email_req_e2e*, if the email is received on an internal corporate account from an internal corporate address;
  - *rec_email_req_e2ext*, if the email is received on an internal corporate account from an external address;
  - *rec_email_req_ext2e*, if the email is received on external account from an internal corporate address;
  - *rec_email_req_ext2ext*, if the email is received on an external account from an external address.
- **Sent emails with an attachment** that might correspond to primitive events:
  - send_email_e2e, if the email is sent from an internal corporate address to an internal corporate address;
  - send_email_e2ext, if the email sent from an internal corporate address to an external address;
  - send_email_ext2e, if the email is sent from an external address to an internal corporate address;
  - send_email_ext2ext, if the email is sent from an external address to an external address.
- **Last Time a Device was mounted** that corresponds to primitive event *sys_mount* defined in the environment description.
- **Installation of an Application** that corresponds to primitive event *sys_install* defined in the environment description.

For each hypothesis, Figure 4 compares the amount of events to be examined to explain the incident if the system is not forensic-ready (No-FR) with the number of events the generated specification requires to preserve. We grouped together the hypotheses requiring the same types of primitive events to be demonstrated or refuted.

To support *h1* the specification requires preserving user logins and the outgoing emails to other employees with the confidential document attached. Due to the nature of the data-set, it was only possible to preserve information about the last logins of user accounts (3 user accounts). No email from the CFO's outbox to another employee was preserved.

The same events were preserved to support *h2*, with the only difference that an email from the CFO's outbox to an external address was preserved as it has the confidential document attached.

If the system is not forensic-ready, it would be necessary to identify the mail clients among the installed applications (133), inspect all the outboxes associated with the accounts registered with the mail clients (2 emails for the CFOs outbox and no emails for the Administrator outbox) and identify the users' last login (3).

To support *h3* and *h4* the generated specification prescribes preserving the same events as for h1

[4] http://www.sleuthkit.org/autopsy/

and h2, respectively, also including all incoming emails received from other employees (31) for h3 and from external email addresses (188) for h4.
If the system is not forensic ready it would be necessary to inspect the same evidence as for h1-h2 also including the CFO's and the Administrator inbox (222 and 1 emails, respectively).

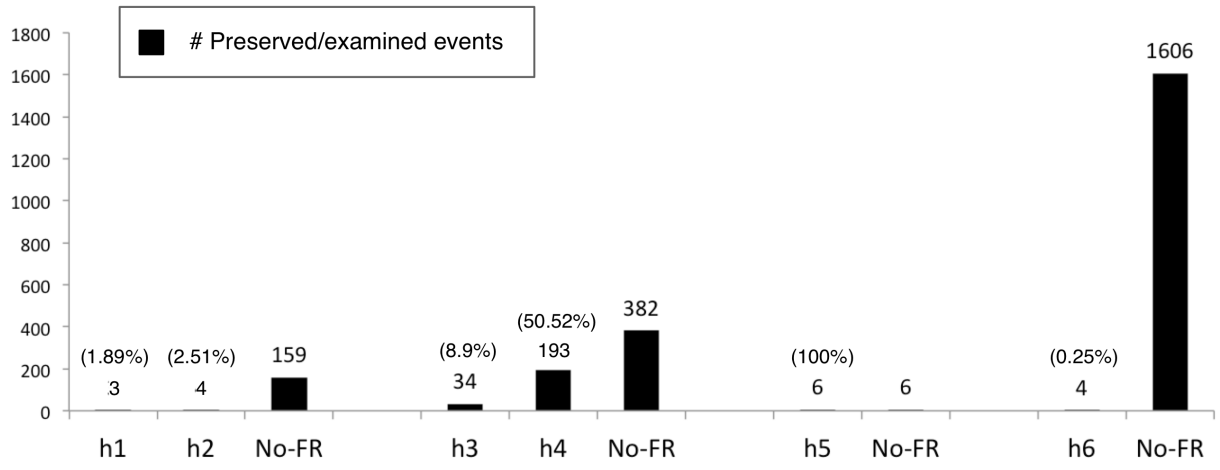

Figure 1. Comparison of number of events to be examined.

To support *h5* the generated specification requires preserving users' last logins, mounted devices and file access operations. From the system registry hive it was possible to preserve events indicating the mounting of 2 USB storage devices and from the file system the time the document was last accessed.
The same amount of data would be examined if the system is not forensic-ready.
To support hypothesis *h6* the generated specification prescribes preserving events indicating installation of malware (i.e. applications not belonging to the NIST National Software Reference Library) and access to untrusted web pages. We preserved events indicating installation of a suspicious application (`QQBubbleArena`) and no visit to untrusted URLs was preserved. If the system is not forensic ready it would be necessary to inspect the complete list of installed application (133) and the web history (1470 entries).

From the data preserved we noticed that h4 might be satisfied since the confidential document was sent in attachment by the CFO in response to another email that she received from an external address (`tuckgorge@gmail.com`). However, we cannot conclude which user was logged on the machine since the data-set only provides information about the last user login.