Protecting Future Personal Computing: Challenging Traditional Network Security Models

Helen Angela Brumfitt
Liverpool John Moores University
School of Computing and Mathematical Sciences
Liverpool, UK
H.A.Brumfitt@2008.ljmu.ac.uk

Dr Robert Askwith
Liverpool John Moores University
Department of Networked Systems and Security
Liverpool, UK
R.J.Askwith@ljmu.ac.uk

Dr Bo Zhou
Liverpool John Moores University
Department of Networked Systems and Security
Liverpool, UK
B.Zhou@ljmu.ac.uk

***Abstract*— The Internet is a notoriously two-way street. If multiple computers can communicate sensitive data across the internet, malicious entities can access the network and collect this data also. The range and number of connected devices is increasing dramatically and with this expansion so is the security risk. Collection of ever rising quantities of data, especially sensitive and personal data, raises many challenges and questions about the suitability of current security. The key problem our research investigates is how we can adapt traditional security models to enhance it both current and future deployment. The work is not aimed to replace existing security although it builds upon it to complement it and enhance existing methods. We utilise the timeliness of the Internet of Things as a focus to develop and experiment with our work. In this paper we present our novel framework and introduce our initial work to prove the concept is feasible. Our initial results are encouraging as to the impact the framework could have on future security.**

Keywords- *Network security; mobile security; smartphone; malware detection; in-network; Collaborative; Internet of Things*

I. Introduction

Security challenges have existed since information technology was first introduced. In 1989 it was recognised that the evolution of computers and networking had advanced too rapidly for security personnel to keep up with it [1]. It was conveyed that the main threats at the time included fire damage, water damage and people (insiders). Slowly the number of threats increased to include terrorism and viruses. In 1997 widespread scanning of vulnerable systems became common, followed by the actual compromise of the vulnerable system, propagating the attack and the coordinated management of attack tools. By 2002 these steps had merged making the process faster as well as attack tools being advanced enough to initiate the attack cycle themselves [2].

The internet was initially implemented to connect people with machines. Users owned a computer that could browse the internet, send emails, create files and complete simple processing. Now in 2015, smart devices have stimulated a new era in terms of technology, creating devices that run our lives. As a result of this, some people can not live without their devices. Smart devices including phones, tablets and others such as watches, fridges and TVs have created the supporting structure for the long awaited Internet of Things (IoT).

Multiple connected things can communicate data about themselves and access data aggregated by other devices. Effectively allowing “anything” to be connected “anywhere” at “anytime”. This allows scenarios such as the alarm clock waking you up early due to too much traffic on your intended route to work, whilst communicating this also to your coffee machine, lights and heating. However, as the internet already has its own security challenges the introduction of millions of connected devices is only adding to these challenges.

Our previous work investigated mobile devices and the impact they have on our personal security as a survey [3]. It investigated threats which are prevalent on them and also the security challenges that they contain. In this paper we build upon this literature review and have used it to challenge current thinking by developing our own novel approach. We present our contribution in the form of a framework consisting of three components. Each component is designed to enhance existing security solutions using both existing techniques and new mechanisms. Overall the framework is designed to be applicable on different networking environments. The rest of the paper is organised as follows. Section II provides an overview of the security challenges both generally and also on up and coming networks such as the Internet of Things (IoT). Section III highlights our contribution in the form of a framework. Section IV discusses the design of the framework. Section V presents some of our experimentation work and results. Our work is finally discussed and future works highlighted in section VI.

II. Security Challenges

Security is enforced to ensure the availability, integrity and confidentiality of the network. Below is a quick overview of network and IoT security challenges [4].

A. Network Security Challenges

Vulnerabilities, threats and attacks are constant security challenges to a network. Firstly, vulnerabilities can include technology weaknesses such as TCP/IP protocol weaknesses, operating system weaknesses or network equipment weaknesses. Configuration weaknesses involve unsecured user accounts, easy passwords, unsecured default settings and misconfigured network equipment or internet services. Security policy weaknesses can be caused by lack of written policies, politics, lack of disaster recovery plans, software and hardware installations not following the enforced policies.

Threats can also be apparent in a number of forms and are structured or unstructured. Structured threats are usually completed by technically competent hackers who are highly motivated and often use sophisticated hacking techniques. Unstructured threats are usually performed by inexperienced people who often use hacking software readily available on the internet. This type of threat can be damaging to a business even without much skill. Many threats are external, originating from outside of the network. Sometimes disgruntled employees can initiate an insider attack taking advantage of the access to the network they already have.

Attacks come in many forms divided into four primary groups. Reconnaissance, access, denial of service or worms, viruses and Trojan horses.

B. *Future network security challenges*

The introduction of environments such as the IoT and mobile networks has influenced new challenges that hinder attempts to secure them. Mobile devices are ubiquitous and heterogeneous in nature, creating complex systems that are difficult to manage safely both in the network and for a users own privacy. The following are some of the factors that contribute to this problem [5].

- Mobile devices have an increased number of vectors in which they can be infected, including by Bluetooth, MMS, HTTP and SMS or generally through attacks in the application layer, communication protocols and operating system.
- Mobile devices are usually always on and with the user.
- Smaller devices means potentially less resources, CPU and memory.
- Numerous sensors that may be present are always on and sensing their environment.
- Constant movement between numerous unknown networks.
- Some users do not seem to know much about these advanced devices and how to use them safely and securely.
- Low physical protection surrounding a mobile device.
- Contradicting goals of current security.
- The Bring Your Own Device (BYOD) policies have a compromise in the way they can't control what a user uses their device for and what they put on their device as well as the security they run on it.

C. Existing Security

As a starting point network models can be either open or closed. In an open model, there isn't usually many security measures such as firewalls, IDS systems or Virtual Private Networks. This keeps the cost and time associated with implementing these low. It allows for full internet and external access, business advantages and is flexible for the users. On the negative side though it would require a very strict security policy to be enforced and a very efficient recovery plan. It would be more difficult to secure and monitor with an increased number of threats.

In the closed model security measures will exist and are configured specifically for the type of network they are implemented on which makes them easier to monitor. This normally works on the assumption that even the users can't be trusted so the threat of an attack is higher therefore restricting access further. This prevents external access from business partners for example and also provides low flexibility.

Security measures which can be enforced include:

- Firewalls
- Intrusion Detection and Prevention Systems
- Virtual Private Networks
- Tunnelling
- Network Access Control
- Security Scanners
- Protocol Analysers
- Authorisation, authentication and accounting

However even with these measures in place we still see many major cyber attacks. Most notably the attack on Xbox and PlayStation services on Christmas Day in 2014 made the news, stopping players from using the consoles for a number of days. Also in the news towards the end of 2014 was the Apple iCloud security flaw which saw intimate images of celebrities stolen from their accounts and leaked on the internet. Again towards the end of 2014 a crushing cyber attack was also launched at Sony Studios. Many of these cyber attacks and data breaches and more are listed in [6]

D. Related Works

Much of the research over general security problems only highlight the number of challenges that exist within the security of IoT. This can be seen in [7] in which the authors recognise a number of challenges. In particular they focus on authentication, as every individual device cannot possibly connect to every other device without security problems. In [8] the authors address mobile security challenges and develop Mobile Guardian, a framework for security policy enforcement on mobile devices. The impact of the IoT in health care is discussed in [9]. The authors propose solutions to alleviate communication problems in the medical profession. These include adding digital certificate management for authentication and also adding ciphers to communications between the peer to peer clients.

Works such as in [10] present new authentication methods for mobile devices. The authors present a four way fusion of user authentication techniques for efficient usable security on mobile devices. This is a good approach as user is not required to remember any alphanumeric password. The location traces, gait pattern, emotion of user and context of an image is used as metric for authentication. Although there are a number of scenarios in which these personal traits of a user may change which needs to be considered within the work.

Current security is not well adapted and needs to be enhanced by new thinking and not just small solutions filling gaps but by a new 'framework' way of approaching it. Security would benefit by being enhanced in a way that will allow the same framework to be applicable in numerous applications in different environments. This is more important now as technology is constantly evolving at an immense rate, we don't know and have no way of knowing what could be released within the next ten years. Keeping up with the fast pace of technology and its prevailing threats is a challenge in itself as is trying to keep ahead of malicious threats, attacks and vulnerabilities. There is outstanding research going into mechanisms to replace sections of existing security or theories that replace it completely. In order to overcome these challenges we have gone for a solution that neither replaces the full security or sections of it. Our framework integrates fully, and the main aim of it is to ensure it detects new threats quickly and efficiently. It must however be applicable in various settings such as the IoT, mobile and cloud computing.

III. The Framework

In order to address the challenges identified above, we have designed a novel framework which is aimed at enhancing general security as well as providing stable building blocks to develop future security. Our framework consists of three components; 1) a Lightweight Forensics Application (LFA) that runs on the device 2) a Central Security Manager (CSM) that runs in the network and 3) a collaborative component that will run between devices in a network. The three components are each designed to complete the own tasks, however the three of them are integrated and work with each other in order to create an effective framework for security. Our framework is novel due to its design to be applicable on any type of network including IoT, cloud, home, enterprise, public or specialised such as medical networks, alongside the components and how they use and integrate with each other. Unlike other works, it is completely non specific, which is beneficial for security as it can be used for a variety of tasks and more importantly work with existing security software which many other works do not. In this work we have used the IoT as a focus and tested mechanisms with smartphones due to their availability. However this does not limit the frameworks use purely to these devices. The work is forensics inspired, meaning the framework will not necessarily know what it is looking for, it doesn't look for specific threats, until it finds something out of place. This has not be used for security so far. The integration of the three components can be seen in figure 1. Both the LFA and Collaborative (Col) components collect data for the CSM. The networks security advisor also has an input into the CSM but it out of the scope of this work.

A. *Lightweight Forensics Application (LFA)*

The LFA is designed to be lightweight to ensure it can be dynamic to adapt to any device it will be integrated on. This includes devices with very low processing power such as small sensors. It has a priority to collect data that the CSM can use in order to enforce security. It does this by adhering to a predefined set of classifications. This data is then sent to the CSM component for further analysis. The LFA doesn't need to know why it is looking for specific data. It has been inspired by forensics as it has a set number of actions to look for but does not know what the end result of these actions could be.

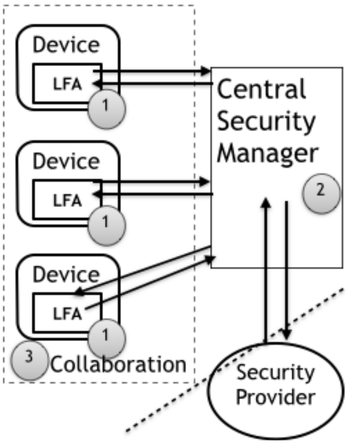


Fig. 1. Framework Component Integration

We are assuming the LFA will run in a protected kernel within the device itself, although saving minimal data and none about the user makes it a low target for malicious users. The LFA may respond to one CSM in work, and another at home, allowing the companies to specify policies and implement them.

B. Central Security Manager (CSM)

The CSM contains all the heavy processing and decision making. The CSM collects data from the LFA and Col components. Resources and processing power will be required in order to utilise data being sent to it from the LFA and Col components. Placing the CSM in the network therefore ensures it has access to what it needs, as well as other assets which will be discussed later in the work. The CSM is designed to be highly automated, making decisions for itself rather than waiting for an admin to assess the problem. Reports will be created by the CSM so security personnel can be kept in the loop of what is happening, and some events may require their attention.

C. Collaborative Component

The Col component collects data in the distributed network which can indicate possible vulnerabilities, threats and attacks on a larger scale. This component also allows for different policies to be applied in different networks. For example a user may have a personal Col network at home with family devices connected to one CSM. As they arrive in work their device automatically switches to the business Col network to ensure policies are followed. This will ensure devices are useable at home as normal although in work restrictions may take hold in order to follow the corporate policies for BYOD. The Col component also allows us to further ensure the LFA remains lightweight by including shared processing and tasks as well as ensuring the LFA on each device is not compromised and secure.

IV. Components Design

We have briefly introduced each of the components and how they integrate with each other. We will now look into their design further to show how they can enhance current security.

A. Lightweight Forensics Application

It could be that in the future that the LFA will facilitate the collection of the data for other purposes such as medical applications, however we are only concerned with the data collection in this work for the CSM. In order to implement this we have decided the LFA has the following main mechanisms:

App Runtime - We are going to assume that the LFA will run in protected kernel space within the mobile device, therefore not accessible via application or operating system vulnerabilities. The LFA quietly monitors actions on the device and will trigger an event when a condition of one or more of the classifications are met. It doesn't store any user data, and will only store triggered events for a short period of time. Purely a data collection component this reduces the risk of it becoming a target for compromise.

Classifications - We will not go through the full design of the classifications in this work. In this section we give a brief overview of what they are designed to do. We don't want the LFA to be searching for everything at any one time as it will use too many resources. Generally an Intrusion Detection System (IDS) uses either anomaly detection or misuse detection to identify malicious actions on a system. In our work, we will be using neither of these as they are too resource intensive for most smaller mobile devices to maintain. We use our own novel technique which will prove lightweight and more efficient to our needs. Our mechanism is formed using classifications, which minimises the list of events the LFA is monitoring for at any one time, although once it detects one event it will then look for a new set relating to that event. This method also ensures low false positive events and aids the LFA in prioritising certain trigger events.

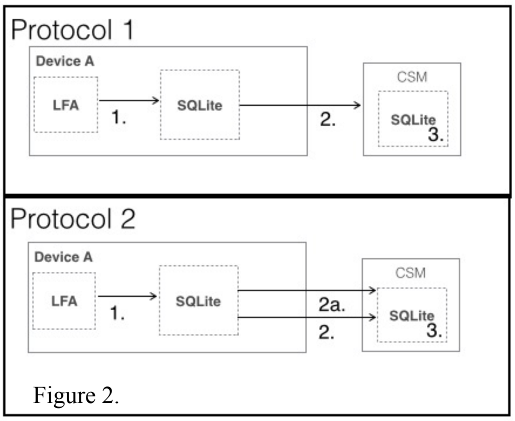


Fig. 2. Trigger events for the classification

The classifications are created and provided by the CSM. Classifications can be used to detect general events or specific threats or events depending on what the CSM or the admin deems necessary. Threats which have not yet been seen before could be recognised as a result of this although the LFA will not recognise it as a new threat. Using this mechanism, we can ensure the LFA remains as lightweight as possible, whilst also not missing suspicious events. This work is not limited to just detecting known problems, even though this system does not know it is detecting anything new.

Trigger Events - When a suspicious event is detected on the device, a report will firstly be sent to the LFAs own event log system shown in figure 2 as point 1. It will then follow one of two protocols. If the device detects a low priority event, then protocol 1 is implemented. At point 2 the event file will be sent to the CSM when requested by the CSM. In point 3 the CSM processes the data. If the device LFA detects a high level event, then protocol 2 will be implemented. This protocol is the same as protocol 1, although when an immediate threat is detected point 2a sends the event to the CSM immediately. In order to reduce the space that the app takes on the device, the event file within the LFA will only store event triggers over a certain time scale and then discard them after the CSM has requested them. By keeping a log of reports sent to the CSM, the CSM will be able to use this to ensure the integrity of the reports that arrive at the CSM.

CSM reaction - When the LFA reports back to the CSM, the CSM may respond with further events to monitor for, for example if the CSM recognises it as a specific attack and wants more information to make a decision. This introduces a dynamic security loop between the LFA and CSM that will use events to direct its next classification rules.

B. Central Security Manager

The LFA and the Collaborative component are the CSMs gateway into the mobile device and the distributed network. Although the CSM itself has a number of mechanisms which help it to enhance current security.

The CSM will receive four types of security reports or data input. The first data type will be from the LFA present on each device. The second type of data will be that from the Collaborative component regarding problems in the network. The third type of data will be form other external sources, such as anti- virus, anti-malware, firewalls and IDS. The fourth and final type of data will be backups of the device if they are required. The CSM will receive these event reports and investigate them in order of priority, requesting more data if required. The first three types of data will be used to ensure the security is running smoothly. The fourth type of data relies on the CSM receiving backups of the devices, which will then be analysed for security and to determine how effective the device is performing.

When the CSM receives data from the four inputs it will analyse it to determine what is happening. It will then use the results to determine what action needs to be taken. The CSM mechanisms are explained below.

- **Broker -** After the initial event analysis the CSM will determine what the most relevant mechanism is to resolve any problem.
- **Connected Devices & Collaborative Networks -** This allows a security manager to monitor all the devices connected to the CSM. Reports from each individual device will also be stored within a record. The collaborative feature allows the admin to see a visual representation of each collaborative network that connected devices form. Reports created by the CSM using the analysis of the collaborative data will also be accessible within here.
- **Security Reports -** Individual reports can be accessed.
- **Assets -** External inputs such as firewall, anti-virus and IDS will be listed here along with their reports and also security policies for the network.
- **Classification Manager -** This mechanism allows the security administrator to review current classifications that are running on a device. They are also given an option to create new classifications based on intel they have acquired and also review the CSMs own automatic classifications based on intel its acquired from the external input assets.
- **Automatic Preventative Security Manager -** Mechanism uses current event analysis to predict possible outcomes and determine if any malicious actions are possible.
- **Data Processing -** Allows the security administrator to add in new analysis techniques to be implemented when a new event record is received by the CSM.
- **Notifications -** This section alerts the security administrator to immediate issues or reports that require their attention. The administrator will also be alerted in other ways such as email and SMS to ensure they don't miss any situations with high priorities.

C. Collaborative Component

Attacks in the distributed network such as a botnet operating by sending texts a few at a time from different devices wouldn't be detected as high priority by the LFA. Multiple devices in the trusted collaborative network however may send low priority event alerts to the CSM, which will then increase the priority as there are multiple devices detecting the same problem. This can be seen in figure 3. The CSM will then instruct the connected devices to warn each other to speed up recovery time. This shows the benefit of the collaborative component in detecting problems and attacks in the distributed environment. This will ultimately enhance security and speed up recovery time.

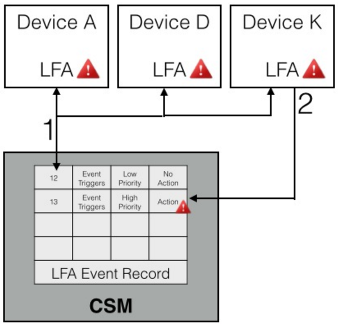


Fig. 3. Collaborative component design

The collaborative component will ensure devices in each collaborative network are trusted. Assuming this the devices will communicate with each other and be able to share tasks if one is struggling. This ensures the LFA's present on the devices remain lightweight and stay effective. This component can also contribute to enhance current security by monitoring devices to ensure they have not been compromised by an external source. Authentication mechanisms will also be relevant here to ensure that devices can only connect to other safe devices with similar security interests.

D. Summary

Overall, our framework enhances the way mobile devices and their security should be developed and deployed using novel mechanisms in each of the components and the approach itself. Our work complements the security implementations in the future, it may not resemble our designs exactly, but will definitely retain the characteristics of our work. We do not focus on any specific attacks in this work, however we work with various forms of attack in order to improve and experiment with our work.

V. Experimentation and Results

In this remainder of this paper we will discuss the experimental implementation we have undertaken and the results we have achieved so far. Due to its timeliness we have decided to use the Internet of Things as a platform on which to trial our framework using smartphones in particular. This is relevant as smartphones have the properties that the IoT relies on and they are more readily available to experiment with. Although the theory of the framework itself will allow it to be implemented in various environments. In this paper we focus on the framework as a whole and how each of the components utilises the others.

A. Lightweight Forensics Application

In order to experiment with the classification mechanism we produced, we have used the Android Eclipse device emulator. One of the test classifications we developed was targeting SPAM messages to a smartphone. If the LFA follows the classification and detects the words "download", "install" and "http://" it automatically notifies the user of the detected possible SPAM message. We experimented with the classification method for a number of different types of example SPAM and social engineering messages and these were picked up by our classification method as expected whilst ignoring all normal social messages sent. This ensures our classification method is suitable for this framework.

Some classifications such as the SMS monitor may run continuously on the device, whereas other classifications may only run intermittently. In particular we have used the alarm

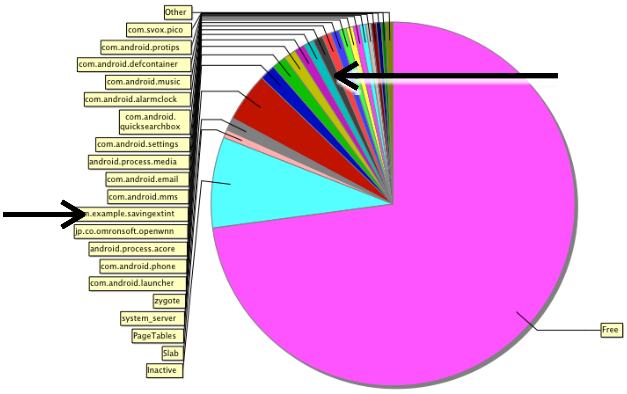
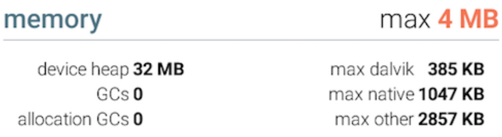


Fig. 4. Memory usage of Classification 3

feature in Android eclipse which runs a background service at regular intervals relying on an alarm to initiate it. This ensures the resources used are as low as possible. In an experiment we have created it checks the battery level every two hours, and if it has dramatically increased compared to the last time it has checked it will check the time of day it is. If it is night time and the phone is in idle mode, a trigger event will be created.

Figure 4 shows the memory usage of classification 3 which is the SMS monitor. The Android Eclipse DDMS information console provided the data. The majority of the memory shown is free memory. The section with the arrow pointing to it and the relevant label is classification 3. In order to experiment with this further we added on more classifications for the LFA to be looking for. In particular it monitored for incoming and outgoing calls, the state of the microphone and camera and also implemented the battery classification discussed earlier.

Whilst the modified classification 3 application was running on a physical device we again monitored memory it was using. The results can be seen in Figure 5. We are encouraged by the reduced memory it is using at it is only searching for minimal events due to the classification mechanism, even though it is monitoring for a number of items. We are also confident we can reduce this more, as it is still currently running on the screen of the device all the time.



memory max 4 MB

device heap **32 MB**
GCs **0**
allocation GCs **0**

max dalvik **385 KB**
max native **1047 KB**
max other **2857 KB**

Fig. 5. Memory Usage of the modified classification 3.

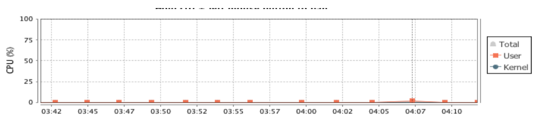
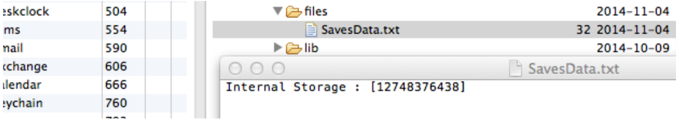


Fig. 6. CPU usage of the modified classification 3.



```
eskclock   504          ▼ files                     2014-11-04
ms         554              SavesData.txt        32 2014-11-04
nail       590          ▶ lib                       2014-10-09
cchange    606
lendar     666
eychain    760
```

SavesData.txt

```
Internal Storage : [12748376438]
```

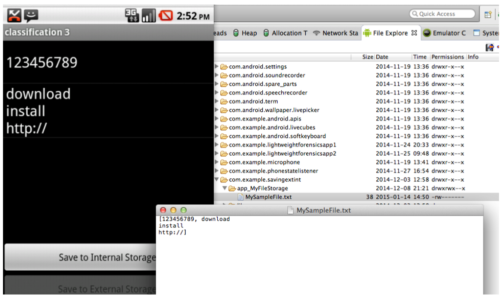
Fig. 7. Android Emulator saving a event log onto the device

The CPU usage for this app can be seen in Figure 6, throughout running the app and purposely triggering a number of the events the CPU % stays below 10%. As discussed earlier some of the classifications may continue to run all the time, whilst others may only run and specified points during the day or night. This will ultimately help to keep the resources used to a minimum.

In order to communicate this event report with the CSM we needed an output method. We initially instructed the LFA to save the number and text which triggered the classification to a .txt or .csv file internally on the device. This can be seen in figure 7 using an Android emulator device. This represents the report and log created by the LFA that would be sent to the CSM. We have implemented this on the virtual device emulated by Eclipse. We have also implemented it on physical Android devices although we had to integrate a way to save it to an external SDCard. This is due to being unable to access files on the device directly, which proves a problem for the testing point of view but in a final LFA implementation would not remain an issue.

The trigger event report is an important feature as it must contain all the information the CSM requires in order to make it useful, whilst also ensuring it doesn't over complicate matters for the LFA causing it to use more resources and processing power trying to send it.

We have experimented using an output as .txt and .csv files, as these can then be imported into a database. The CSM will be able to use this feature to create a log of all the events, separate them into relevant classifications and look at them all as an overview to discover patterns in trigger event behaviour across the device. The CSM will then create a full report which can be reviewed by relevant specialists who can identify the malicious actions taking place on the device. The CSM may also decide to request a sample, for example coding of an app, the permissions it uses, what data it gathers so it can integrate this into the report and send it to specialised scanning software to identify known malicious software.



123456789

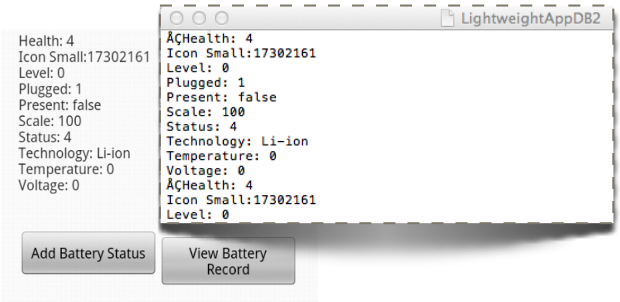
download
install
http://

```
                                         Size Date       Time  Permissions Info
▶ com.android.settings                        2014-11-19 13:36 drwxr-x--x
▶ com.android.soundrecorder                   2014-11-19 13:36 drwxr-x--x
▶ com.android.spare_parts                     2014-11-19 13:36 drwxr-x--x
▶ com.android.speechrecorder                  2014-11-19 13:36 drwxr-x--x
▶ com.android.term                            2014-11-19 13:36 drwxr-x--x
▶ com.android.wallpaper.livepicker            2014-11-19 13:36 drwxr-x--x
▶ com.example.android.apis                    2014-11-19 13:36 drwxr-x--x
▶ com.example.android.livecubes               2014-11-19 13:36 drwxr-x--x
▶ com.example.android.softkeyboard            2014-11-19 13:36 drwxr-x--x
▶ com.example.lightweightforensicsapp1        2014-11-24 20:33 drwxr-x--x
▶ com.example.lightweightforensicsapp2        2014-11-25 09:48 drwxr-x--x
▶ com.example.microphone                      2014-11-19 13:41 drwxr-x--x
▶ com.example.phonestatelistener              2014-11-27 16:54 drwxr-x--x
▼ com.example.savingextint                    2014-12-03 12:58 drwxr-x--x
  ▼ app_MyFileStorage                         2014-12-08 21:21 drwxrwx--x
      MySampleFile.txt                     38 2015-01-14 14:50 -rw-------
```

MySampleFile.txt

```
[123456789, download
install
http://]
```

Fig. 8. The output file received from classification 3.



```
Health: 4
Icon Small:17302161
Level: 0
Plugged: 1
Present: false
Scale: 100
Status: 4
Technology: Li-ion
Temperature: 0
Voltage: 0
```

```
ÂÇHealth: 4
Icon Small:17302161
Level: 0
Plugged: 1
Present: false
Scale: 100
Status: 4
Technology: Li-ion
Temperature: 0
Voltage: 0
ÂÇHealth: 4
Icon Small:17302161
Level: 0
```

Fig. 9. Saving Data in an SQLite

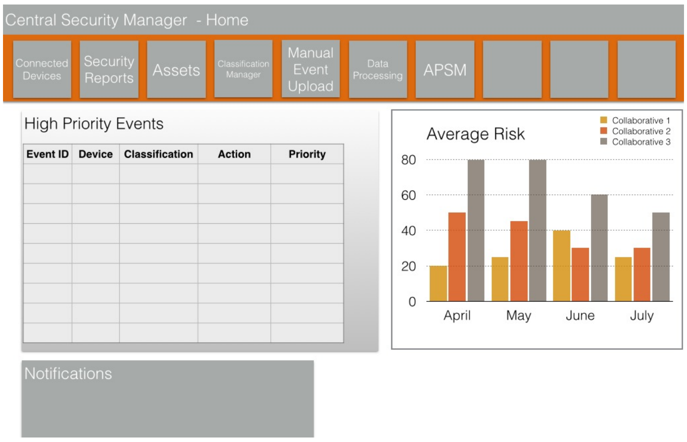
Figure 8 shows the output file we have received from classification 3. At the time it only shows the output to include the phone number it was received from and also the event that triggered the LFA to react to it.

In further experiments we decided on using SQLite files to save the events detected on the device. This can be seen in figure 9 using the battery classification. This decision was made as it allows us to then add further data in as required in the same file and makes it easier for the CSM to read.

B. Central Security Manager

Figure 10 shows the CSM user interface we have implemented. It is currently running on a local web server in order to experiment and test it. During the creation of it we have decided on a number of necessary features it should have. These are explained below.

The interface allows the network administrator to monitor everything from one console. Anything urgent they will need to view on another tab will be displayed here. For high priority situations the security manager will also be messaged via email or SMS to ensure they don't miss the notifications. As shown in figure 10 there is a list of the most recent high priority events, notifications which includes the most recent reports or decisions the admin needs to be aware of and finally a graph showing the average risk of three separate collaborative networks. This will change depending on how many collaborative networks are linked with the CSM. The admin can use this to track the devices risk over time.



Event ID	Device	Classification	Action	Priority

Fig. 10. CSM user interface

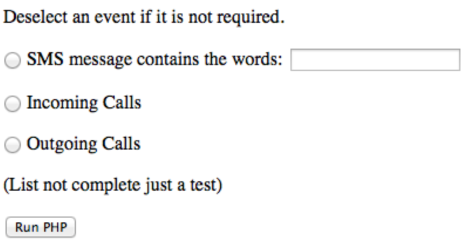
Connected Devices - This panel allows the administrator to monitor what is happening on the devices. The CSM will automatically collect the SQLite files representing events from separate devices and add them to their own individual records after processing the data and creating any necessary reports. The admin is also given the option to check the device history and also the performance of the device. This may allow problems such as battery deterioration to be picked up early.

Assets - In order to enhance current security we are utilising the outputs of other security software to further provide data for our work. An IDS could aid the detection of unknown devices to the CSM, and also confirm the number of known devices to ensure all devices are accounted for as well as alerting the CSM to potentially malicious traffic. The anti-virus and anti-malware software could notify the CSM of any new or existing threats so the CSM can react to it by checking its own devices. The firewall can also contribute by notifying the CSM of blocked traffic, and also monitoring the inbound, outbound and internal traffic so the CSM can pick up on suspicious activities such as DDoS early. The CSM can respond to the external inputs by creating new classifications to be run by individual LFAs in order to further enhance its own and existing security.

Data Processing - We have implemented a testing web server which will allow us to upload an SQLite file. The web server will then display this on the screen. The SQLite file we have simulated would represent one sent by the LFA and Col components after they detect an event. When the CSM receives the file, it will process what is potentially happening in the report and decide which component, if any, it should be forwarded to. It will then add the report to the individual devices own records.

Classification Manager - The classification manager will allow the administrator to view and edit the current classifications available as well as add a new classification based on any knowledge they have learned. The classification manager will also give the administrator the option to view the automatic classifications created by the CSM in response to reports from external inputs.

We have identified the need for a mechanism to ensure administrator can create a new classification as quickly as possible. The CSM needs to translate what the user wants and turn this into something the LFA can listen and respond to. Initially as a proof of concept we have created a protocol in which the CSM will contain a bank of code of possible actions the LFA could do. This is represented on a user interface to an



Deselect an event if it is not required.

○ SMS message contains the words: ▭

○ Incoming Calls

○ Outgoing Calls

(List not complete just a test)

Fig. 11. Manual classification implementation

administrator as a simple list of the names of the actions. The admin selects the tools they want the LFA to run and starts the process. This can be seen in figure 11 which shows a simple version of the form filled in by the administrator, in this case the user will be selecting to run SMS monitor, incoming calls or outgoing calls. We have implemented this using a MAMP web server. Once the admin selects the relevant events or classifications and selects the 'Run PHP' button, the CSM then matches the selected items to those items in its code bank. Once it has all the events and their relevant classifications it will create the code necessary to send to the LFA.

We want the administrator to be able to add new classifications quickly to make the overall protocol more effective. In order to do this any event that is possible to monitor on a specific device will be added to the CSMs code bank. The admin can then select various classification events and add them to a classification to ensure the priorities are followed and to avoid using too many resources.

C. Collaborative Component

In order to test how the collaborative component can further enhance security, we created a classification which ran across two Raspberry Pi devices. If one device detects motion in a room it will communicate this to the other device. If the second device then also detects motion in a separate room it takes a picture when the motion is active and then emails it to the security administrator. Although a relatively simple test it proves the devices benefit from working together.

VI. Discussion and Further Works

In this paper we have explored alternatives to current security and complement and extend existing security in the form of a novel framework. A combination of which we believe will be a significant step to enhance the security of our devices from malicious actions in future networking. This work has successfully set out to prove that the theory of our solution is feasible within a series of experiments and prototype components and will complement existing security. Our results in this work are very encouraging. The framework can be integrated onto a number of mobile devices within a number of different environments.

The framework is novel as a whole due to the way in which the components interact and bring security successfully into a new paradigm. It allows us to continuously run security at a faster pace across multiple devices. Unlike other solutions it uses many existing security applications and brings them together along with some of its own mechanisms to build up on the security. Using forensics as an inspiration it can be used to detect known and unknown existing vulnerabilities, threats, attacks and even device malfunctions. Even though in this paper we have focused heavily on the smartphone, the same mechanisms would apply on other devices such as the smart watch, laptops, computers tablets and housing appliances in smart homes.

Current and future networks are moving towards heavy processing and data storage on the network, and smaller devices on the edge of the network. These devices can include a range of types such as smartphones, desktop computers, laptops, smart watches, smart fridges, smart TVs and even smart houses. Our framework is a perfect candidate to be run on this type of network due to its flexibility. It will run on a number of different networks, such as Cloud, IoT, home, enterprise and many more. Proving dynamic the framework can be implemented in different ways to suit the network but ultimately each component will still be present for the same reason using the same initial theory. An advantage of this work over others is the ability to detect a wide range of problems, not just specific issues.

In the future we are continuing our work with the LFA in order to develop some higher level classifications which can be tested to their full capabilities on physical devices. We will also develop the collaborative approach to use resource sharing and authentication to ensure the devices that communicate with each other are trusted as well as trusted monitoring. We are currently experimenting with the mini computer 'Raspberry Pi' in order to implement this. Our work with the CSM will be continued, specifically focusing on how it can utilise the data it receives and create informative reports that can be used by security administrators or other components alike.

References

1. Reese, L.F., "Challenges faced today by computer security practitioners," *Computer Security Applications Conference, 1989., Fifth Annual* , vol., no., pp.143,, 4-8 Dec 1989

2. Householder, A.; Houle, K.; Dougherty, C., "Computer attack trends challenge Internet security," *Computer* , vol.35, no.4, pp.5,7, Apr 2002

3. H. Brumfitt and R. Askwith, "Mobile Devices: In-Network Anti-Malware Analysis," *The 14th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, 2013.

4. Antoon W. Rufi, "Introduction to Network Security," in *Network Security*, 1st ed. Indianapolis, USA: Cisco Press, 2007, ch. 1, sec. 2, pp. 2-21.

5. A. Arabo and B. Pranggono, "Mobile Malware and Smart Device Security: Trends, Challenges and Solutions," *19th International Conference on Control Systems and Computer Science*, pp. 526–531, May 2013.

6. L. Morgan (2014, December 23rd). "List of cyber attacks and data breaches in 2014". [Online]. Available: http://www.itgovernance.co.uk/blog/list-of-the-hacks-and-breaches-in-2014/

7 P. N. Mahalle, N. R. Prasad, and R. Prasad, "Novel Threshold Cryptography-based Group Authentication (TCGA) Scheme for the Internet of Things (IoT)." *IEEE ANTS 2013 Seventh IEEE International Conference on Advanced Networks and Telecommunication Systems*, 2013.

8 Yong Wang; Vangury, K.; Nikolai, J., "MobileGuardian: A security policy enforcement framework for mobile devices," *Collaboration Technologies and Systems (CTS), 2014 International Conference on* , vol., no., pp.197,202, 19-23 May 2014.

9 Jara, A.J.; Zamora, M.A.; Skarmeta, A.F.G., "An Architecture Based on Internet of Things to Support Mobility and Security in Medical Environments," *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE* , vol., no., pp.1,5, 9-12 Jan. 2010

10 Rahman, F.; Gani, M.O.; Ahsan, G.M.T.; Ahamed, S.I., "Seeing Beyond Visibility: A Four Way Fusion of User Authentication for Efficient Usable Security on Mobile Devices," *Software Security and Reliability-Companion (SERE-C), 2014 IEEE Eighth International Conference on* , vol., no., pp. 121,129, June 30 2014-July 2 2014.