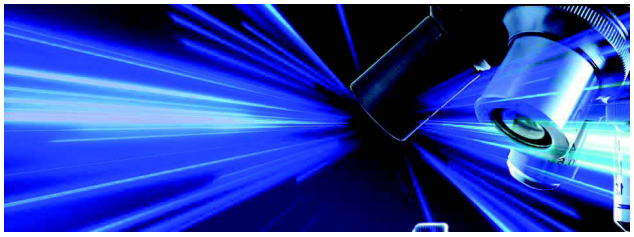
Proceedings

LASER 2016
Learning from Authoritative Security Experiment Results



Proceedings

LASER 2016
Learning from Authoritative Security Experiment Results

San Jose, CA, USA
May 26, 2016

ISBN 978-1-931971-35-5

Table of Contents

Program

Program Committee

Matt Bishop (University of California, Davis)
Rainer Böhme (University of Innsbruck, Austria)
Jed Crandall (University of New Mexico)
Serge Egelman (University of California, Berkeley and International Computer Science Institute)
Roya Ensafi (Princeton University)
Adrienne Porter Felt (Google)
Cormac Herley (Microsoft Research)
Chris Kanich (University of Illinois, Chicago)
David Manz (Pacific Northwest National Laboratory)
Damon McCoy (New York University)
Daniela Oliveira (University of Florida)
Alina Oprea (RSA Laboratories)
Vern Paxson (University of California, Berkeley and International Computer Science Institute)
Kami Vaniea (University of Edinburgh)
Robert Watson (Cambridge University)

Organizing Committee

Terry Benzel (ISI), General Chair
Sean Peisert (University of California, Davis, and Lawrence Berkeley National Lab), Program Chair
David Balenson (SRI International), Funding/Local Arrangements/Scholarships
Sami Saydjari (Cyber Defense Agency), Publicity
Laura S. Tinnel (SRI International), Publications/Web/IT Services
Carrie Gates (Dell), Advisor
Greg Shannon (CMU/CERT), Advisor

Workshop Sponsors

Message from the General Chair

Welcome to the 2016 Workshop on Learning from Authoritative Security Experiment Results.

Each year, LASER focuses on an aspect of experimentation in cyber security. The 2016 workshop focus was on cyber security experimentation methods and results that demonstrate approaches to increasing the repeatability and archiving of experiments, methods, results, and data.

The event was structured as a workshop with invited talks and a variety of guided group discussions in order to best meet the overall workshop goals.

LASER 2016 sought research papers exemplifying the practice of science in cyber security, whether the results were positive or negative. Papers documenting experiments with a well-reasoned hypothesis, a rigorous experimental methodology for testing that hypothesis, and results that proved, disproved, or failed to prove the hypothesis were sought.

This year, many of the papers and talks for the 2016 LASER Workshop included aspects of human-technology research, user studies, and usability. This theme was highlighted in the keynote talk, **"**Understanding the Cognitive Science of Cyber Security," by Nancy Cooke from Arizona State University.

We received 13 submissions, which were each reviewed by at least 3 members of the Program Committee. The Program Committee accepted 6 full papers, which they believed embodied the workshop spirit and focus of LASER 2016.

This year, the LASER Workshop was held as one of the IEEE Security and Privacy Workshops in conjunction with the IEEE Symposium on Security and Privacy.

LASER recognizes that the future of cybersecurity lies with the next generation of researchers. As such, LASER sponsors students who are working to become researchers to attend and participate in the workshop. In 2016, three students received full sponsorship.

On behalf of LASER 2016, I wish to thank the many people who made this workshop possible:

- Our program chairs, who worked diligently to put together a strong technical program that would benefit the community.
- The authors who submitted papers to this workshop.
- The members of the Program Committee, who carefully reviewed the submissions and participated in paper discussions.
- Our organizing committee, who provided guidance and donated their time to handle publicity, logistics, publications, and finances.
- The National Science Foundation, the IEEE Technical Committee, and our other sponsors, who provided the funding and facilities necessary to make the workshop a reality.
- The attendees, without whom there would be no workshop at all.

We look forward to meeting everyone at LASER 2017!

Terry Benzel, *USC Information Sciences Institute*
LASER 2016 General Chair

Kharon dataset: Android malware under a microscope

N. Kiss
EPI CIDRE
CentraleSupelec, Inria, Univ. Rennes 1, CNRS,
F-35065 Rennes, France

J.-F. Lalande
INSA Centre Val de Loire
Univ. Orléans, LIFO EA 4022,
F-18020 Bourges, France

M. Leslous, V. Viet Triem Tong
EPI CIDRE
CentraleSupelec, Inria, Univ. Rennes 1, CNRS,
F-35065 Rennes, France

Abstract

Background – This study is related to the understanding of Android malware that now populate smartphone's markets. **Aim** – Our main objective is to help other malware researchers to better understand how malware works. Additionally, we aim at supporting the reproducibility of experiments analyzing malware samples: such a collection should improve the comparison of new detection or analysis methods. **Methodology** – In order to achieve these goals, we describe here an Android malware collection called Kharon. This collection gives as much as possible a representation of the diversity of malware types. With such a dataset, we manually dissected each malware by reversing their code. We run them in a controlled and monitored real smartphone in order to extract their precise behavior. We also summarized their behavior using a graph representations of the information flows induced by an execution. With such a process, we obtained a precise knowledge of their malicious code and actions. **Results and conclusions** – Researchers can figure out the engineering efforts of malware developers and understand their programming patterns. Another important result of this study is that most of malware now include triggering techniques that delay and hide their malicious activities. We also think that this collection can initiate a reference test set for future research works.

1 Introduction

Android malware have become a very active research subject in the last years. Inevitably, all new propositions of detection, analysis, classification or remediation of malware must deal with their own evaluation. This evaluation will rely on a set of "malicious indicators" that have to be detected/analyzed/classified as bad and a set of "legitimate indicators" that have to be ignored by the evaluation method. Designing a set of "good things" appears simple but on the contrary, for precise evaluation, the set of "bad things" should be perfectly understood. We claim here that rigorous experiments have to rely on malware samples totally reversed.

Building an understandable dataset to be used for dynamic analysis is a difficult challenge. Indeed, an automatic methodology for reverse engineering a malware does not exist. First, no mature reverse engineering tool has been developed for Android that would be comparable to the ones used for x86 malware. Second, each malware is different and finding automatically the malicious code by statically analyzing the bytecode is a very difficult task because this code is mixed up with benign code. It requires a human expertise to extract relevant parts of the code. Finally, most advanced malware now include countermeasures to avoid to trigger their malicious behavior at first run and in emulated environments. Thus, an additional expertise is required to understand the special events and conditions the malware is awaiting.

Thus, building an understandable malware dataset requires a huge amount of work. We made this effort for evaluating our previous works [1] and we propose here to make our training dataset well documented in order to initiate the construction of a reference dataset of Android malware. Our goal is to build a well documented set of malware that researchers can use to conduct reproducible experiments. This dataset tries to represent most of the possible know types of malware that can be found. When choosing a malware for representing a type, we excluded the malware that are too obfuscated or encrypted to be reversed engineered in a reasonable time.

The contributions of the paper are:

1. A precise description of the internals of 7 malware samples i.e. how each malware attacks the operating system, how it interacts with external servers and the effects from the user perspective;

2. A graphical view of the induced information flows when the malware is successfully executed;

3. Instructions on how to trigger the malware in order to make reproducible the attacks operated by each malware of the dataset. These instructions are essential for conducting experimental evaluation of methodologies that analyze dynamic events.

In the following, we give an overview of existing Android malware datasets and present online services dedicated to malware analysis. In Section 3, we put seven malware under a microscope and give a precise description of each of them. Section 4 concludes this article.

2 Related works

2.1 Android security basics

Android security relies on standard Unix and Java security paradigms that are the base of standard Linux distributions. Android processes are isolated from each others using different Unix process ids but applications can still communicate between each other, by using so called *Intents* that transport exchanged information. Application are executed by a virtual machine or compiled by a Ahead-of-time compiler. Both mechanisms include runtime checks for implementing security and the most important security checks are guaranteed by the Linux kernel itself. For example, the access to the network is provided using a dedicated *inet* group.

A special file, called the Manifest, declares the software components that are the possible entry points for the application. The most important components can be: an *Activity* i.e. a set of graphical components for composing a screen of an application; a *Service* that can be run in or outside the main process of the application and has no graphical representation; a *BroadcastReceiver* that executes the declared callback when the application receives information. For starting or making these components communicate, *Intent*s are Java Objects that provide facilities to transport data. Some pre-defined *Intent*s encode some system events. For instance, the *Intent BOOT_COMPLETED* notifies applications that the smartphone has finished the boot process. As malware need to communicate, we often observe the use of *Intent*s.

The security policy of an application is expressed by the permissions declared in the Manifest. Permissions can protect resources (network, data, sensors, etc.) or system data or components (list of processes, ability to keep the smartphone awake, etc.). Other advanced security mechanisms can be found in recent Android versions such as checking the boot sequence integrity or the use of SELinux for enforcing mandatory policies at kernel level.

2.2 Malware datasets

One of the most known dataset, the Genome Project, has been used by Zhou et al. in 2012 to present an overview of Android malware [19]. The dataset is made of 1260 malware samples belonging to 49 malware families. The analysis was focused on four features of Android malware: how they infect users' device, their malicious intent, the techniques they use to avoid detection and how the attacks are triggered. The last feature is the most interesting for us as we want to provide a dataset that can be easily used by people working on dynamic analysis of Android malware. According to Zhou et al. analysis, Android malware can register for system events to launch their attack e.g. the `BOOT_COMPLETED` event sent when the smartphone is up. In addition to system events, some malware directly hijacks the main activity or the handler of the user interface components.

Unfortunately, the exact condition required to execute the malicious code is never provided by the paper's authors. For example, for the case of DroidKungFu1, we know that the malicious code can be launched at boot time but there is no indication about the time bomb used to schedule the execution of the malware. Indeed, DroidKungFu1 executes its malicious code only when 240 minutes have passed and this condition is checked by reading a specific value in the application preferences. Without this information, a dynamic analysis fails to observe interesting behaviors.

In [4], Arzt et al. present FlowDroid, a static taint analysis tool for Android applications. The goal of FlowDroid is to detect data leaks in Android applications using static analysis. To evaluate their tool, Arzt et al. have developed DroidBench[1], a set of applications implementing different types of data leakage thanks to implicit information flows, use of callbacks and reflections. Applications in this dataset are classified according to the technique they use to leak data and contain a description of the leak performed by the application. For each application, the source code for performing the leak is given, which makes result comparison and evaluation easy. At the time of writing, the DroidBench repository contains 120 applications of which APK and source code are both available. As it only performs a data leak, the source code is really minimalist. Unfortunately, the applications in DroidBench are not real malware samples and are only meant to evaluate dynamic and static analysis tools. They do not provide a dataset with the complexity of real malware in which benign code is mixed with malicious code.

Contagio dataset is a public collection of Android malware samples [15]. It was created in 2011 and is regularly updated, which makes it one of the most up to date public dataset. Each contribution is published as an article on the blog associated to the dataset with a link to

download the samples, a description, and an external link generally to a VirusTotal report. The static analysis part of the report seems to be done with Androguard and provides different information such as the required permissions, the components, the use of reflection, cryptography, etc. The dynamic analysis part lists the observed behavior during the execution: started services, accessed files, use of sensitive functions and connection to remote servers. Such information gives an insight on the nature of the application but is useless to determine how to launch the malicious code of a malware sample.

2.3 Online services

Some analysis services are provided online for commercial or research purposes. They are mainly developed to detect Android malware and potentially harmful applications but can also give a better understanding of an application.

One of them, Verify Apps, is the service used by Google to scan applications submitted on Google Play and applications installed on users' devices. Google does not provide any technical detail on their service but according to their report on Android security for 2014 [8], their tool uses a mix of static analysis and dynamic analysis. The goal of the analysis is to extract multiple features of the application and decide if it is potentially harmful by comparing these features with the ones used by known malware. For instance, the service compares the developer's signature with known signatures that are associated with malicious developers or malicious applications. The report provided by Google gives an insight on the type of security threats but lacks details on how these threats are executed. Unfortunately, the results of the analysis are not publicly available which makes the service not useful for research purposes.

Andrubis [12] is an online service that analyzes Android applications statically and dynamically to detect malicious behaviors using a combination of TaintDroid [7], Androguard, apktool and have analyzed more than 1,000,000 applications. Lastly, VirusTotal is an online scanning platform that uses 54 antiviruses and 61 online scan engines to perform analysis on files uploaded on its web page or sent by email. It uses several tools to perform the analysis, such as Androguard to disassemble and decompile APK packages, and Cuckoo sandbox to dynamically analyze an execution.

We claim that these platforms give very basic information and are not sufficient to understand deeply malware. We believe that every research team conduct apart their own reverse analysis. It is a huge amount of work which is often redone and thus has to be gathered and published. Thus, the description of a malware dataset is a complementary approach to online analysis tools.

Table 1: Malware of the Kharon dataset

Malware	Description SHA 256 hash value	Known Samples
BadNews [16]	Remote administration tool (Contagio) 2ee72413370c543347a0847d71882373c1a78-a1561ac4faa39a73e4215bb2c3b	15
SimpleLocker [13]	Ransomware (Contagio) 8a918c3aa53ccd89aaa102a235def5dcffa04-7e75097c1ded2dd2363bae7cf97	1
DroidKungFu [10]	Remote admin. tool (Genome project) 54f3c7f4a79184886e8a85a743f31743a0218-ae9cc2be2a5e72c6ede33a4e66e	34
MobiDash [6]	Agressive adware (Koodous) b41d8296242c6395eee9e5aa7b2c626a2-08a7acce979bc37f6cb7ec5e777665a	4
SaveMe [11]	Spyware (Contagio) 919a015245f045a8da7652cefac26e71808b2-2635c6f3217fd1f0debd61d4330	1
WipeLocker [5]	Data eraser (Contagio) f75678b7e7fa2ed0f0d2999800f2a6a66c717-ef76b33a7432f1ca3435b4831e0	1
Cajino [18]	Spyware (Contagio) 31801dfbd7db343b1f7de70737dbab2c5c664-63ceb84ed7eeab8872e9629199	4

3 Seven malware under a microscope

In this section, we present a detailed analysis of seven malware. We randomlly studied a lot of malware (more than 30) and selected the ones that were not too obfuscated or using ciphering techniques. We choosed recent ones that have been known to have been widespread on user's smartphones. These seven malware cover most of the known types of malware [19]: Aggressive adware, Fee paying services malicious usage, Ransomware, Remote Administration Tool, Spyware and Data Eraser. When studying each malware candidate for representing a type, we excluded the malware that are too obfuscated or encrypted to be reversed engineered in a reasonable time. We followed the advices of Rossow et al. [17] in order to support any future experiments: the dataset is balanced, cleaned and studied in a controlled sandbox (*correctness*); the experiment setup and malware list is documented (*transparency*); malware are executed in a real smartphone and sufficiently stimulated (*realism*); chosen malware have no network spread capabilities (*safety*).

For each malware presented in Table 1, we indicate its provenance in order to help researchers to rebuild the dataset[2]. We conducted a two step analysis in order to precisely describe their malicious code and their triggering condition. In a first part, we have manually reversed the bytecode and inspected it. This static analysis helps us to locate where the malicious code is and learn how it can be triggered. In a second part we did a dynamic analysis, triggered the previously identified malicious code and thus monitored all the malicious be-

haviors. We performed the experiment on a Nexus S with Android 4.0 *Ice Cream Sandwich* to which we added AndroBlare (further details below). We rooted our device and installed the Superuser[3] application if the application requires root privileges. Our monitoring process consists in dynamically tracking where information belonging to the analyzed sample spread during its execution and then building what we call a System Flow Graph to observe the malicious behavior of Android malware [2]. The information flow tracking is done thanks to AndroBlare[4], a tool that tracks at system level the information flow between system objects such as files, processes and sockets. The produced directed graph represents the observed information flows: it is a compact and human-readable representation of the observed malware activities captured by AndroBlare. The vertices are the information containers such as files and the edges are the information flows observed between these information containers.

3.1 BadNews, a remote administration tool

Badnews [16] is a remote administration tool discovered in April 2013. Its malicious final charge depends on commands received from a remote server. The malicious code is located in the package *com.mobidisplay.advertsv1*. Its behavior can be divided into three distinguished stages:

Stage 1: Malicious service setup and sensitive data recovery. Badnews starts at the reception of the `BOOT_COMPLETED` intent or the `PHONE_STATE` intent. When one of these intents is received, the service *AdvService* is started. On creation, this service collects information about the device such as the IMEI, the device model, the phone number and the network operator. Finally, this service sets up an alarm manager that is in charge of broadcasting an intent for the receiver *AReceiver*. This receiver will restart *AdvService* every four hours with an intent containing an extra data named *update* and set to true.

```
final AlarmManager aM = this.getApplicationContext().
      getSystemService("alarm");
final PendingIntent broadcast = PendingIntent.getBroadcast((
      Context)this, 0, new Intent(this, AReceiver.class),
      134217728);
aM.cancel(broadcast);
final long elapsedRealtime = SystemClock.elapsedRealtime();
aM.setRepeating(3, elapsedRealtime, 14400000L, broadcast);
```

Stage 2: Notify the C&C server of the availability of the device. Badnews transforms the device into a slave of a C&C server located at *http://xxxplay.net/api/adv.php*[5]. When *AdvService* is restarted with the extra data *update* set to true, it creates a thread which executes a function named *getUpdate()*. This function contacts the server and begins with sending an HTTP post request with the sensitive information collected on creation.

Stage 3: Execute the service order. The function *getUpdate()* then receives an answer from the server, which contains one of the following orders: 1) Open an URL; 2) Create a notification with an URL to open; 3) Install a shortcut that will open an URL; 4) Download and install an APK file 5) Create a notification with an APK file to download and install 6) Install a shortcut that will download an APK; 7) Update the primary or secondary server address. The APK files that might be installed are potentially malicious. During our observations, the server sent a malicious version of Doodle Jump and a fake version of Adobe Flash that seems to be a game.

Triggering Condition. As the malware requires a server to obey commands, we built a fake server and forged the commands. For example, for implementing a fake install command of another malware (*malware2.apk*), we create a file *index.html* containing:

```
{"status" : "install",
 "sound" : 0,
 "vibro" : 0,
 "apkname" : "Malware2",
 "url" : "http://192.168.0.10/malware2.apk"}
```

where *192.168.0.10* is the address of the local computer. We serve this file using a python web server. For substituting the server url in badnews.apk, we unpack the APK, substitute *http://xxxplay.net/api/adv.php* by *192.168.0.10/index.html*, repack it again, and sign the new APK:

```
$ apktool d badnews.apk # Then edit the smali files
$ apktool b badnews -o new_badnews.apk
$ jarsigner -verbose -keystore ~/.android/debug.keystore -
      storepass android -keypass android new_badnews.apk
      androiddebugkey
```

Then we install the new APK and force the service to avoid waiting 4 hours:

```
$ adb install new_badnews.apk
$ adb shell am startservice ru.blogspot.playsib.savageknife/com.
      mobidisplay.advertsv1.AdvService -ez update 1
```

3.2 SimpleLocker, a ransomware

Simplelocker [13] is a ransomware discovered in 2014. It encrypts user's multimedia files stored in the SD card. The original files are deleted and the malware asks a ransom to decrypt the files. Our sample displays instructions in Russian. Simplelocker communicates with a server hidden behind a Tor network to receive orders, for example the payment confirmation.

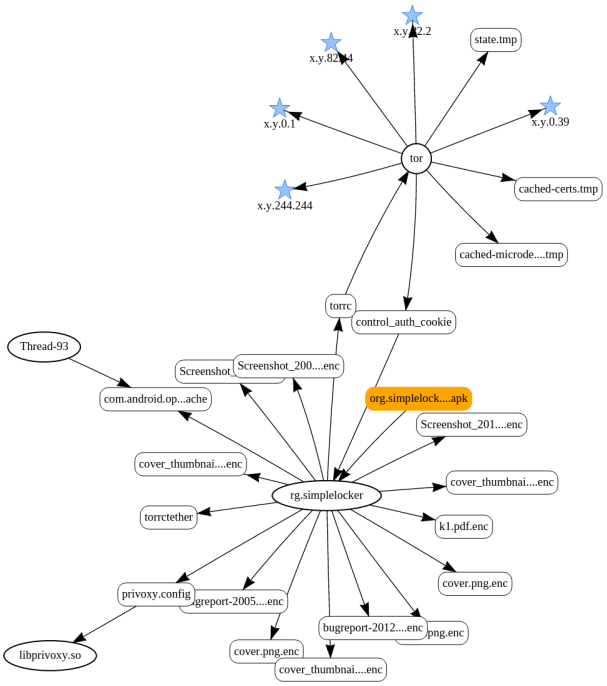


Figure 1: Information flows induced by an execution of SimpleLocker

Simplelocker relies on the execution of three main independent processes. First, *rg.simplelocker* runs the graphical interface, the main service and the different repetitive tasks. Second, *libprivoxy.so* and *tor* are two processes that give access to the Tor network.

Stage 1: Malicious code execution. SimpleLocker waits for the `BOOT_COMPLETED` intent. When it occurs, it starts a service located in the *MainService* class. Starting the main activity with the launcher also starts the service. The service takes a *WakeLock* on the phone in order to get the device running the malware even if the screen goes off. Then, it schedules two repetitive task executors (*MainService$3* and *MainService$4*) and launches a new thread (*MainService$5*). All these jobs are executed in the main process *rg.simplelocker*.

Stage 2: Communication with a server via Tor. A task executor *MainService$3*, launched every 180 seconds, sends an intent *TOR_SERVICE* to start the *TorService* class. If Tor is already up, the *TorSender* class is called to send the IMEI of the phone using the service. The *TorService* class is a huge class that setups executables: it copies and gives executable permission to the files *libprivoxy.so* and *libtor.so* that come from the APK. The *libprivoxy.so* process is executed calling from the *MainService class* as shown below:

```
final String[] array = { String.valueOf(this.filePrivoxy.
      getAbsolutePath()) + " " + new File(this.appBinHome, "
      privoxy.config").getAbsolutePath() + " &" };
TorServiceUtils.doShellCommand(array, sb, false, false);
```

The *libprivoxy.so* process listens for HTTP requests on the port 9050. It is an HTTP proxy that filters and cleans the request generated and received by the *tor* client.

Stage 3: User's data encryption. In the *MainService$5* thread, the malware encrypts all the multimedia files and deletes the original ones:

```
for (final String s : this.filesToEncrypt) {
  aesCrypt.encrypt(s, String.valueOf(s) + ".enc");
  new File(s).delete(); }
```

The used algorithm is AES in CBC mode with PKCS#7 padding. The encryption key is a constant in the code: we were able to generate a modified version of this malware where we have forced the decryption of the files.

The repetitive task *MainService$4*, checks in the *SharedPreferences* the value *DISABLE_LOCKER*: if set, the malware knows that the victim has paid. If not, it restarts the *Main* activity that displays a fullscreen Russian message informing the user that its files have been encrypted and asking for a ransom.

Triggering Condition. To trigger the malware, launch the application or reboot the device.

Information flow observations. Figure 1 shows that SimpleLocker is constituted of four independent processes (ellipses). The main process named *rg.simplelocker* writes the encrypted version of the multimedia files (`*.enc`). The process named *tor* is the process that communicates through the Tor network, using five sockets (stars). Four of them are nodes of the Tor circuit used to reach the server and the fifth is the interface used to send and receive messages. The *libprivoxy.so* process is the HTTP proxy used in combination with Tor.

3.3 DroidKungFu1, a remote admin tool

DroidKungFu1 is a malware discovered in the middle of 2011 that is able to install an application without any notification to the user. We have included this malware in our dataset because it is a well known malware that presents interesting features. We do not give a lot of details about it and we refer the reader to [10, 3].

The malicious code of the malware is included into the package *com.google.ssearch* that contains four classes. The most important class is *SearchService.class*. The malware also comes with 4 noteworthy assets : *gjsvro*, an encrypted version of the *udev* exploit, *ratc*, an encrypted version of the exploit *Rage Against The Cage*, *legacy*, an

APK file that contains a fake Google Search application, and *killall*, a *ratc* wrapper.

Stage 1: Setup of a countdown. DroidKungFu waits the `BOOT_COMPLETED` intent to start the service *SearchService*. When the service starts for the first time, it writes the current time in an XML file `stimestamps.xml` and stops itself. Every time *SearchService* is restarted, it checks if the elapsed time between the time of the restart and the time of `stimestamps.xml` exceeds four hours.

Stage 2: Installation of a fake Google Search app. When the period of four hours has expired, the malware collects sensitive information about the device (the IMEI, the device model, the phone number, the SDK version, memory size, network information) and tries to use the *Exploid* or *RATC* exploits. If it fails, it tries to use the *su* binary to become root. Then it extracts an APK from the asset *legacy*. This APK file is placed into the directory `/system/app` and further detected by `system_server` as a new application to be installed.

Stage 3: Executing the C&C server commands. Then, the malware or the fake Google Search app can receive commands from a remote server. This way, if the originating infected app is removed, the malware can still be able to receive commands through the fake Google Search app. The commands can be: install or delete any package, start an application or open a web page.

Triggering Condition. To trigger this malware, install the application, launch it once and reboot the phone. Then you need to execute the following command:

```
adb pull /data/data/com.allen.mp/shared_prefs/sstimestamp.
    xml
```

Modify the value *start* to 1 and push back the file in the phone. After that, just reboot the phone again.

3.4 MobiDash, an adware

MobiDash [6] is an adware discovered in January 2015. Hidden behind a functional card game, it displays unwanted ads each time the user unlocks the screen. To evade dynamic analysis tools, the malware waits several days before executing its malicious code. For that purpose the malware uses three internal states, namely *None*, *Waiting* and *WaitingCompleted*. The default state is *None*. The malware switches from *None* to *Waiting* when rebooted and reaches the state *WaitingCompleted* after a fixed countdown. Finally, it starts to display ads.

Stage 1: Bootstrapping the configuration. When the application is launched for the first time, the activity *com.cardgame.durak.activities.ActivityStart* is created and it calls the *InitAds()* function from the *MyAdActivity* class. This triggers a bootstrap procedure in which the file `res/raw/ads_settings.json` is read. This file contains information about the malware configuration, and in particular, contains the server to be contacted and the time to wait before triggering (called *OverappStartDelaySeconds*). In our sample, the server is `http://xxx.mads.bz`[6] and the delay is 24 hours. All these parameters are then saved in the *SharedPreferences* and the malware has reached the state *None*.

Stage 2: From state *None* to *Waiting*. Once the device is rebooted, the `BOOT_COMPLETED` intent is received by the *DisplayCheckRebootReceiver* and it triggers the *ping()* function from the *AdsOverappRunner* class. This function checks the internal state of the malware and executes a specific function for each case.

```
final AdsOverappRunner.State state = getState(context);
switch (
$SWITCH_TABLE$mobi$dash$overapp$AdsOverappRunner$State
      ()[state.ordinal()]) {
        case 1: {
                startWait(context);
                break;
        }
        case 2: {
                checkForCompleted(context);
                break;
        }
        case 3: {
                startAds(context);
                break;
        }
}
```

If the state is *None*, it calls the *startWait()* function which changes the internal state into *Waiting*, saves the current time in the *SharedPreferences* and setups two alarms. The first (resp. second) alarm is used to retrigger the *DisplayCheckRebootReceiver* every 15 minutes (resp. 24 hours).

```
protected static void startWait(final Context context) {
        setState(context, AdsOverappRunner.State.Waiting);
        setWaitStartTime(context, System.currentTimeMillis());
        DisplayCheckRebootReceiver.setupPingAlarms(context);
        DisplayCheckRebootReceiver.setupPingAlarmOne(context
              , (long)(AdsExtras.getOverappStartDelaySeconds
              ()*1000+1000)); }
```

Stage 3: From state *Waiting* to *WaitingCompleted*. The next call to *ping()* (with the *Waiting* state) will execute the *checkForCompleted()* function. This function checks if the delay has expired, changes the state to *WaitingCompleted* and calls the *startAds()* function. *startAds()* starts the service *DisplayCheckService* that request ads to the server and display them. Additionally,

the service sets up an alarm, as done in *startWait()*, in order to restart itself every 15 minutes. It also dynamically registers two receivers:

```
protected void setupUserPresent() {
       this.registerReceiver(this.screenOffReceiver, new
            IntentFilter("android.intent.action.SCREEN_OFF")
            );
       this.registerReceiver(this.userPresentReceiver, new
            IntentFilter("android.intent.action.
            USER_PRESENT")); }
```

The first receiver requests new ads each time the screen turns off by calling the *requestAds()* function. The second receiver displays an ad each time the user unlocks the screen by calling the *showLink()* function.

Additional features. When reversing the malware, we observed that the class *HomepageInjector* changes the browser homepage and the class *AdsShortcutUtils* installs launcher shortcuts. During our observations, none of these features have been activated.

We also observed that our malware sample contains a lot of different lawful Advertising Service SDK: *AdBuddiz*, *AdMob*, *Flurry*, *MoPub*, *Chartboost*, *PlayHaven*, *TapIt* and *Moarbile*. Nevertheless, the malware main activity (*ActivityMain$11*) only uses *AdBuddiz*, *AdMob* and *Chartboost*. To finish, log files about all the downloaded malicious ads are stored in the folder `data/data/com.cardgame.durak/files/mobi.dash.history/active/`. These logs contain information such as the requests to the server.

Triggering Condition. First, the application must be launched a first time and the device must be rebooted in order to reach the state *WaitingCompleted*. Then, by setting *waitStartTime* to 0 in the XML file of the directory `/data/data/com.cardgame.durak/shared_prefs/com.cardgame.durak_preferences.xml` and rebooting again, the malicious code is triggered. The smartphone must be rebooted promptly after modifying the file, for example by pushing it with adb, in order to avoid the malware to overwrite the modification.

Information flow observations. We give in Figure 2 the full graph of MobiDash as an example of a malware that generates a lot of system events. The main process `cardgame.durak` reads the file `ads_settings.json` to configure itself and connects to a large amount of IP addresses. Some of those IP are contacted by the originating game itself to retrieve fair ads and most of them are contacted by the malware to download malicious ads. The IP addresses shared between `cardgame.durak` and `android.browser` are connections opened when aggressive ads are displayed in fullscreen in a webview. We notice that the malware saves its history in a local directory, producing a lot of log files.

3.5 SaveMe, a spyware

SaveMe [11] is a spyware discovered in January 2015. It presents itself as a standalone application that is supposed to backup contacts and SMS messages. SaveMe seems to be a variant of another malware known as *SocialPath* [14]. The application has been available on *Google Play* before being removed.

Stage 1: Sensitive data recovery. When the application is launched, it asks to the user his name and phone number and saves these inputs in its local database *user_info4*. In background, the activity collects the device's MAC address, network operator name and ISO country code. Those information are then all sent to a master server, located at `http://xxxxmarketing.com`[7] (no longer available).

The visible part of the application offers features such as: add or delete a contact, save or restore your phonebook, save all your SMS messages and write a SOS message that will be sent to all your contacts in case your phone has been stolen. If you choose to save your messages, the application will save all the content of `content://sms/inbox` and `content://sms/sent` in its local database *user_info* and send it to the server.

Stage 2: Execute the master commands. In parallel, when the application is launched, a service named *CHECKUPD* is started (it also starts each time the device is rebooted). This service is used as a handshake between the device and the server. It executes three *AsyncTask* namely *sendmyinfos()*, *sendmystatus()* and *senddata()* for dialoging with the server. After those exchanges, the main service *GTSTSR* is executed. The purpose of this service is to contact the server in order to get commands to be executed. Depending on the answer given by the server, the service can perform different actions as detailed below.

First, it can send a text message to any number given by the server. We believe that this can be used for premium services as stated in [14].

```
if (GTSTSR.Mac.equals(this.address) && GTSTSR.
     Send_ESms.equals("SESHB")){
new update().var(this.address,"","SESFK","","","","","");
SmsManager.getDefault().sendTextMessage(GTSTSR.
     EXT_SMS, null, GTSTSR.SMS, null, null);
return; }
```

The service can also make a call by starting a service named *RC*. This service displays a *WebView* on the screen, probably to hide the call and makes a call to a potentially premium number given by the server [14].

```
Intent localIntent = new Intent("android.intent.action.CALL");
localIntent.setData(Uri.parse("tel:" + EXT_CALL));
intent.addFlags(268435456); intent.addFlags(4);
this.startActivity(intent);
```

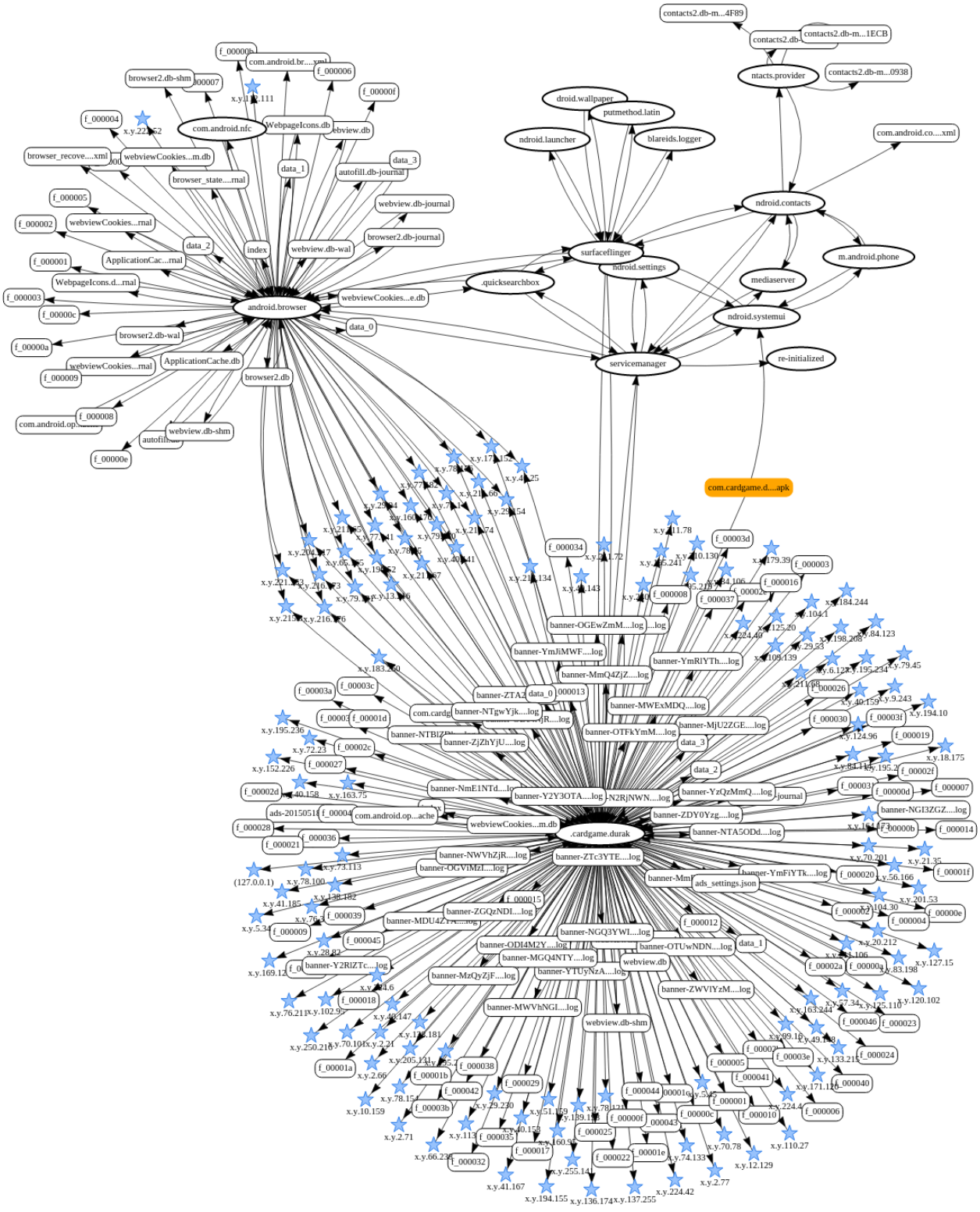


Figure 2: Information flows induced by an execution of MobiDash

After few moments, the service ends the call, removes the *WebView* and deletes the call in the call log by calling the function *DeleteNumFromCallLog()*.

```
final Uri parse = Uri.parse("content://call_log/calls");
contentResolver.delete(parse, "number=?", new String[]{s});
```

GTSTSR can also start a service named *CO* which will automatically fetch all the contacts of the victim and send them to the server. The main difference compared with the official feature of the application (except that there is no need to click on a button) is that *CO* will also steal contacts stored in the SIM card by reading `content://icc/adn`. Contacts are then stored in the database *user_info* before being sent.

The last feature provided by *GTSTSR* is the sending of text messages to victim's contacts by starting the service *SCHKMS*. The service checks the database *user_info*, picks one contact and sends him a message. This feature is used for spreading the malware via SMS containing a link [14]. Of course, the service deletes the SMS from the logs in order to hide it to the victim.

To finish with this malware, we observed a piece of code in the activity *pack* which allows the app to remove its icon from the launcher, in order to hide itself. This way, the victim may forget to uninstall the application. Nevertheless, this activity is never used in this sample.

```
this.getPackageManager().setComponentEnabledSetting(this.
    getComponentName(),
    COMPONENT_ENABLED_STATE_DISABLED,
    DONT_KILL_APP);
```

Triggering Condition. To trigger this malware it is sufficient to use the application icon or to reboot the device. Internet must be enabled for the malware to start.

3.6 WipeLocker, a blocker and data eraser

WipeLocker [5] is a malware discovered in September 2014. It blocks some social apps with a fullscreen hacking message and wipes off the SD card. It also sends SMS messages to victim's contacts. It might be an app for helping the sell of antivirus.

The malware presents itself as a fake *Angry Bird Transformers* game. Once the application is launched, the main activity performs three actions.

Stage 1: Starting the malicious service. The application first starts the service *IntentServiceClass* that can also be triggered by the `BOOT_COMPLETED` event. This service schedules the execution of *MyServices.getTopActivity()* every $0.5s$ and *MyServices.Async_sendSMS()* every $5s$. *getTopActivity()* checks the current foreground activity: if it is a social app like *Facebook*, *Hangouts* or *WhatsApp*, it displays a fullscreen image "*Obey or Be Hacked*", making impossible to use those apps. *MyServices.Async_sendSMS()* is an AsyncTask that sends a text message every $5s$ to all the victim's contacts: "*HEY!!! <contact_name> Elite has hacked you.Obey or be hacked*".

Stage 2: Activating the device admin features. The second action of the malware is to ask the user to activate the device administration features of the app [9]. If the user declines, the app will ask again, over and over, until the user accepts to do so. Administration features allow an application to perform sensitive operations such as wiping the device content or enforcing a password security policy. The file `res/xml/device_admin_sample.xml` declares the operations the application intends to handle. The content of this file is however empty, which means that the application will not handle any sensitive operations: the purpose of this stage is to make the app much harder to uninstall because device administrators cannot be uninstalled like normal apps. If the user accepts, the app closes itself and remove its icon from the launcher.

Stage 3: Wiping off the SD card. The last action performed by the malware is the deletion of all the files and directories of the external storage. Even if the user declined the device administration features, the function *wipeMemoryCard()* is called. This function uses *Environment.getExternalStorageDirectory()* to get the path of the external storage, and then calls *File.listFiles()* for iterating on files and deleting each of them.

Stage 4: Intercepting SMS. A last feature that comes with the malware is the interception of incoming SMS. It is simply a receiver named *SMSReceiver* that is triggered by the `SMS_RECEIVED` intent. When an SMS is received, the malware automatically answers to the sender with the message "*Elite has hacked you.Obey or be hacked*". The victim is not notified by the system about any incoming SMS because the receiver has a high priority (2147483647 in the manifest) and calls *abortBroadcast()* just after reading the message.

Triggering Condition. The icon launcher triggers all the features. A reboot of the device triggers the service.

3.7 Cajino, a spyware

Cajino is a spyware discovered in March 2015. Its particularity is to receive commands via *Baidu Cloud Push* messages. In addition to alternative markets, samples were downloadable on the *Google Play* store with more than 50.000 downloads.

Stage 1: Registration. The application must be launched at least one time. When it occurs, in the *onCreate()* function, a registration procedure of the *Baidu* API is executed in order to make the phone able to receive Push messages from the remote server.

```
if(!Utils.hasBind(this.getApplicationContext())){
        PushManager.startWork(this.getApplicationContext(),
        0, Utils.getMetaValue((Context)this, "api_key")); }
```

At the same time, the *MainActivity* displays an empty *WebView* and a dialog box pops up asking for an update with a "Yes" or "No" choice, with no code behind.

Stage 2: Receiving Push messages. The malware has a receiver named *PushMessageReceiver*. It can react to these intents broadcasted by *Baidu* services:

```
com.baidu.android.pushservice.action.MESSAGE
com.baidu.android.pushservice.action.RECEIVE
com.baidu.android.pushservice.action.notification.CLICK
```

When a Push message is received, *PushMessageReceiver* starts *BaiduUtils.getFile()* which checks if the device is concerned by the incoming message, and if so, starts *BaiduUtils.getIt()* to execute the right command. The commands are designed to: steal the contacts, steal the call logs, steal all SMS (inbox and sent), get the last known location of the device, steal sensitive data (IMEI, IMSI, phone number), list all data stored on the external storage. For each of these features, the malware first stores the results in files written into `/sdcard/DCIM/Camera/` before uploading them to the remote server. The malware can also send SMS to any phone number given by the server, upload to the server or delete any file stored on the external storage.

In some other versions, e.g. *ca.ji.no.method2*, more features are available. For example it can record the microphone with a *MediaRecorder* during a period of time given by the server:

```
BaiduUtils.recorder.prepare(); BaiduUtils.recorder.start();
Thread.sleep(int1 * 1000);
BaiduUtils.recorder.stop(); BaiduUtils.recorder.release();
```

It can also download an APK file into the directory `/sdcard/update/` and install it on the device:

```
private static void installApk(final Context context, String
      str) {
str = Environment.getExternalStorageDirectory()
        + "/update/update.apk";
final Intent intent = new Intent("android.intent.action.VIEW");
intent.addFlags(268435456);
intent.setDataAndType(Uri.fromFile(new File(str)),
        "application/vnd.android.package-archive");
context.startActivity(intent); }
```

The last feature of *Cajino* is a classical call to a number given by the server, not hidden from the user. That makes a total of 12 distinct features the malware can perform.

Triggering Condition. Launch the app to trigger the registration, then you need to wait for a Push message from the remote server. If you want to force the execution of a command, for example for listing the files of `/sdcard/`, send an intent with adb:

```
adb shell am broadcast -a com.baidu.android.pushservice.action
      .MESSAGE --es message_string "all list_file"
```

3.8 Dataset summary and usage

Table 2 gives an overview of the studied malware. For each of them, we recall their protection against dynamic analysis and give the main actions for defeating these protections. These remediation techniques will support the reproducibility of future research experiments.

We have used our dataset to evaluate the performances of GroddDroid [1], a tool for triggering malicious behaviors that targets suspicious methods. On four of them, the targeted methods were automatically triggered. On MobiDash, a method with benign code were targeted (false positive) and on SimpleLocker, GroddDroid had a crash. This example shows that documented dataset helps to measure if a proposed method works fine. Of course, for larger datasets, an other approach should be used to compute the false positive/negative results, but the use of Kharon dataset gives an opportunity to carefully check if a tool works as expected.

4 Conclusion

In this article, we have proposed to initiate the construction of a dataset of seven Android malware that illustrate as much as possible existing malware behaviors. These malware are recent, from 2011 to 2015. For all of them we detailed their expected behavior, isolated the malicious code and we observed their actions in a controlled smartphone. All these materials can be found online on the Kharon website.

An important result of this study is that these malware present a pool of techniques to hide themselves from dynamic analyzers. Thus, we explain how to trigger their malicious code in order to increase the reproducibility of research experiments that need malware execution.

We continue to supply the dataset and additional descriptions of malware can be read. We also propose to other actors of the community to enlarge this dataset. For that purpose, we encourage researchers to gather their experience by signaling us their own documentation about reversed android malware. We will be pleased to integrate any contribution. This way, we hope that this effort will bring new inputs for the research community and will become a reference dataset for precise and reproducible malware analysis.

Table 2: Malware dataset summary

Type	Name	Discovery	Protection against dynamic Analysis → *Remediation*	Details for reproducibility
Remote Admin Tool	Badnews	2013	Obeys to a remote server and delays the attack → *Modify the apk* → *Build a fake server*	Section 3.1
Ransomware	SimpleLocker	2014	Waits the reboot of the device → *send a BOOT_COMPLETED intent*	Section 3.2
Remote Admin Tool	DroidKungFu	2011	Delayed Attack → *Modify the value* `start` *to* `1` *in sstimestamp.xml*	Section 3.3
Adware	MobiDash	2015	Delayed Attack → *Launch the infected application, reboot the device and modify* `com.cardgame.durak_preferences.xml`	Section 3.4
Spyware	SaveMe	2015	Verifies the Internet access → *Enable Internet access and launch the application*	Section 3.5
Phone Blocker + Data Eraser	WipeLocker	2014	Delayed Attack → *Press the icon launcher and reboot the device*	Section 3.6
Spyware	Cajino	2015	Obeys to a remote server → *Simulate the remote server by sending an intent*	Section 3.7

Future works deal with comparing these seven malware with larger datasets in order to build automatic classification techniques. Moreover, for advanced malware that implement sophisticated protections such as obfuscation or ciphering, new investigations should be designed in order to link the static analysis of the code with dynamic analysis.

5 Most important malicious functions

In the following, we give the most 5 most important functions of each malware. It may help researchers to check that their experiment successfully executes the malicious code.

```
# Badnews

com.mobidisplay.advertsv1.AdvService.fillPostData()
com.mobidisplay.advertsv1.AdvService.onStartCommand(final
      Intent intent, final int n, final int n2)
com.mobidisplay.advertsv1.AdvService.startUpdater()
com.mobidisplay.advertsv1.AdvService.sendRequest(String string
      )
com.mobidisplay.advertsv1.AReceiver.onReceive(Context context
      , Intent intent)

# SimpleLocker

org.simplelocker.MainService.onCreate()
org.simplelocker.MainService$4.run()
org.simplelocker.TorSender.sendCheck(final Context context)
org.simplelocker.FilesEncryptor.encrypt()
org.simplelocker.AesCrypt.AesCrypt(final String s)

# DroidKungFu

com.google.ssearch.SearchService.onCreate()
com.google.ssearch.SearchService.updateInfo()
com.google.ssearch.SearchService.cpLegacyRes()
com.google.ssearch.Utils.decrypt(final byte[] input)
com.google.ssearch.Utils$PkgManager.deleteApp(final Context
      context, final String str)

# MobiDash

myutils/activity/MyAdActivity.InitAds(int n, ChartboostDelegate
      chartboostDelegate, int n2)
mobi/dash/overapp/AdsOverappRunners.ping(final Context
      context)
mobi/dash/overapp/AdsOverappRunners.startWait(final Context
      context)
mobi/dash/overapp/AdsOverappRunners.checkForCompleted(
      final Context context)
mobi/dash/overapp/DisplayCheckService.setupUserPresent()

# SaveMe

com.savemebeta.GTSTSR.CHECK()
com.savemebeta.RC.callnow()
com.savemebeta.LogUtility.DeleteNumFromCallLog(final
      ContentResolver contentResolver, final String s)
com.savemebeta.CO.allSIMContact()
com.savemebeta.SCHKMS.fetchContacts()

# WipeLocker

com.elite.MainActivity.onCreate(final Bundle bundle)
com.elite.MainActivity.wipeMemoryCard()
com.elite.MyServices.Async_sendSMS.doInBackground(Void ...
      arrvoid)
com.elite.MyServices.getTopActivity(final Context context)
com.elite.MainActivity.HideAppFromLauncher(final Context
      context)

# Cajino

ca.ji.no.method3.MainActivity.onCreate(Bundle bundle)
ca.ji.no.method3.BaiduUtils.getIt(final String s, final
      Context context)
ca.ji.no.method3.BaiduUtils.getLocation(final Context context
      , final String s)
ca.ji.no.method3.BaiduUtils.sendSMS(final String s, final
      String s2)
ca.ji.no.method2.BaiduUtils.installApk(final Context context,
      String string)
```

6 Availability

All malware descriptions and graphs can be accessed online at:

http://kharon.gforge.inria.fr/dataset

7 Acknowledgments

This work has received a French government support granted to the COMIN Labs excellence laboratory and managed by the National Research Agency in the "Investing for the Future" program under reference ANR-10-LABX-07-01.

We would like to thank the research engineers and engineering students of the Cyber Security Master of CentraleSupélec and Telecom Bretagne, who participated to the reverse engineering of the dataset. Our special thanks go to Radoniaina Andriatsimandefitra, Béatrice Bannier, Sylvain Bale, Etienne Charron, Loïc Cloatre, Marc Menu, Guillaume Savy.

References

[1] ABRAHAM, A., ANDRIATSIMANDEFITRA, R., BRUNELAT, A., LALANDE, J.-F., AND VIET TRIEM TONG, V. GroddDroid: a Gorilla for Triggering Malicious Behaviors. In *10th International Conference on Malicious and Unwanted Software* (Fajardo, Puerto Rico, oct 2015), IEEE Computer Society, pp. 119–127.

[2] ANDRIATSIMANDEFITRA, R., AND VIET TRIEM TONG, V. Capturing Android Malware Behaviour using System Flow Graph. In *8th International Conference on Network and System Security* (Xi'an, China, Oct. 2014), M. H. Au, B. Carminati, and C.-C. J. Kuo, Eds., Springer International Publishing, pp. 534–541.

[3] ARSENE, L. An android malware analysis: Droidkungfu, Nov. 2012. http://www.hotforsecurity.com/blog/an-android-malware-analysis-droidkungfu-4474.html.

[4] ARZT, S., RASTHOFER, S., FRITZ, C., BODDEN, E., BARTEL, A., KLEIN, J., LE TRAON, Y., OCTEAU, D., AND MCDANIEL, P. FlowDroid: Precise Context, Flow, Field, Object-sensitive and Lifecycle-aware Taint Analysis for Android Apps. In *ACM SIGPLAN Conference on Programming Language Design and Implementation* (Edinburgh, UK, jun 2014), vol. 49, ACM Press, pp. 259–269.

[5] CHRYSAIDOS, N. Android WipeLocker - Obey or be hacked, Sept. 2014. http://www.virqdroid.com/2014/09/android-wipelocker-obey-or-be-hacked.html.

[6] CHYTRY, F. Apps on google play pose as games and infect millions of users with adware, Feb. 2015. https://blog.avast.com/2015/02/03/apps-on-google-play-pose-as-games-and-infect-millions-of-users-with-adware/.

[7] ENCK, W., GILBERT, P., CHUN, B.-G., COX, L. P., JUNG, J., MCDANIEL, P., AND SHETH, A. N. TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones. In *9th USENIX Symposium on Operating Systems Design and Implementation* (Vancouver, BC, Canada, Oct. 2010), USENIX Association, pp. 393–407.

[8] GOOGLE. Android security 2014 year in review. https://static.googleusercontent.com/media/source.android.com/en//devices/tech/security/reports/Google_Android_Security_2014_Report_Final.pdf.

[9] GOOGLE. Device administration. https://developer.android.com/guide/topics/admin/device-admin.html.

[10] JIANG, X. Security alert: New sophisticated android malware droidkungfu found in alternative chinese app markets, May 2011. http://www.csc.ncsu.edu/faculty/jiang/DroidKungFu.html.

[11] LINDEN, J. The privacy tool that wasn't: SocialPath malware pretends to protect your data, then steals it, Jan. 2015. https://blog.lookout.com/blog/2015/01/06/socialpath/.

[12] LINDORFER, M., NEUGSCHWANDTNER, M., WEICHSELBAUM, L., FRATANTONIO, Y., VAN DER VEEN, V., AND PLATZER, C. Andrubis - 1,000,000 Apps Later: A View on Current Android Malware Behaviors. In *3rd International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security* (Wroclaw, Poland, Sept. 2014).

[13] LIPOVSKY, R. ESET analyzes first android file-encrypting, TOR-enabled ransomware, June 2014. http://www.welivesecurity.com/2014/06/04/simplocker/.

[14] NEMCOK, M. Warning: Mobile privacy tools "socialpath" and "save me" are malware, Jan. 2015. http://blog.lifars.com/2015/01/11/warning-mobile-privacy-tools-socialpath-and-save-me-are-malware/.

[15] PARKOUR, M. Contagio mobile, 2012. http://contagiominidump.blogspot.fr/.

[16] ROGERS, M. The bearer of BadNews, Mar. 2013. https://blog.lookout.com/blog/2013/04/19/the-bearer-of-badnews-malware-google-play/.

[17] ROSSOW, C., DIETRICH, C. J., GRIER, C., KREIBICH, C., PAXSON, V., POHLMANN, N., BOS, H., AND VAN STEEN, M. Prudent practices for designing malware experiments: Status quo and outlook. In *IEEE Symposium on Security and Privacy* (San Francisco Bay Area, CA, USA, may 2012), IEEE Computer Society, pp. 65–79.

[18] STEFANKO, L. Remote administration trojan using baidu cloud push service, Mar. 2015. http://b0n1.blogspot.fr/2015/03/remote-administration-trojan-using.html.

[19] ZHOU, Y., AND JIANG, X. Dissecting android malware: Characterization and evolution. In *IEEE Symposium on Security and Privacy* (San Francisco Bay Area, CA, USA, may 2012), IEEE Computer Society, pp. 95–109.

Notes

[1] https://github.com/secure-software-engineering/DroidBench

[2] We warn the readers that these samples have to be used for research purpose only. We also advise to carefully check the SHA256 hash of the studied malware samples and to manipulate them in a sandboxed environment. In particular, the manipulation of these malware impose to follow safety rules of your Institutional Review Boards.

[3] https://play.google.com/store/apps/details?id=com.noshufou.android.su

[4] https://www.blare-ids.org

[5] We intentionally anonymized this URL

[6] We intentionally anonymized this URL

[7] We intentionally anonymized the URL

The Effect of Repeated Login Prompts on Phishing Susceptibility

Peter Snyder and Chris Kanich

Department of Computer Science
University of Illinois at Chicago
{psnyde2,ckanich}@uic.edu

Michael K. Reiter

Department of Computer Science
University of North Carolina at Chapel Hill
reiter@cs.unc.edu

Abstract

Background. Understanding the human aspects of phishing susceptibility is an important component in building effective defenses. People type passwords so often that it is possible that this act makes each individual password less safe from phishing attacks.

Aim. This study investigated whether the act of re-authenticating to password-based login forms causes users to become less vigilant toward impostor sites, thus making them more susceptible to phishing attacks. Our goal was to determine whether users who type their passwords more often are more susceptible to phishing than users who type their passwords less often. If so, this result could lead to theoretically well-grounded best practices regarding login-session length limits and re-authentication practices.

Method. We built a custom browser extension which logs password entry events and has the capability of shortening session times for a treatment group of users. We recruited subjects from our local campus population, and had them run the extension for two months. After this time, we conducted a synthetic phishing attack on all research subjects, followed by a debriefing. Our research protocol was approved by the University's IRB.

Results. We failed to reject the null hypothesis. We found that login frequency has no noticeable effect on phishing susceptibility. Our high phishing success rate of 39.3% was likely a leading factor in this result.

Conclusions. This study confirmed prior research showing exceedingly high phishing success rates. We also observed that recruiting only in-person and campus-affiliated users greatly reduced our subject pool, and that the extension-based investigation method, while promising, faces significant challenges itself due to deployed extension-based malware defenses.

1 Introduction and Motivation

Of all cybersecurity attacks, phishing is perhaps most intimately tied to user decisions and behavior, rather than technical weaknesses of the platform on which it is perpetrated. Despite numerous studies both aimed at better understanding why and how people fall for phishing attacks [3, 4] and new systems to detect the sites themselves and protect users [9], the problem continues [6]. Thus, fully understanding the causes and effects of the phenomenon is a crucial component of successful defenses.

Fundamental to phishing as an attack is the user credential, which is most often a password. Passwords have been the subject of intense investigation along many dimensions; e.g., Bonneau et al. consider many of these dimensions and present a framework for comparing passwords with other authentication schemes [1].

Beyond the weaknesses of passwords discussed by Bonneau et al., the difficulties of password entry are exacerbated by the frequency with which users are asked to re-authenticate to sites. In the Internet's infancy, personal computers were commonly shared among different users, but software to enable efficient sharing via different profiles did not yet exist. Thus, automatically logging users out after some relatively short period of time became the de facto default security posture with respect to authenticated sessions. In the present day, however, browsing often takes place on single-user devices like smartphones, and even when not, streamlined browser profiles make separating different user accounts easy.

Several services like Facebook and Google have thus adopted a strategy where sessions remain logged in for an indeterminate amount of time.[1] However, to our knowledge there is no scholarship or best-practices document available which provides a breakdown of how successful this practice is, and it has not seen widespread adoption

[1] It remains to be seen whether such infrequent reauthentication might itself be harmful: if passwords are so rarely re-entered, the user runs the risk of forgetting it simply because they never use it.

on the rest of the web.

With respect to phishing defenses, this might indeed be a very good one: the longer a session remains in place on a given device, the less common the sight of the login screen (or even any login screen across all sites) is to the individual user. If login screens are less common, we hypothesize that this will cause the user to be more alert when logging in to services. While this may be the reasoning behind the long-lived sessions used by large Internet companies, being able to reproduce this effect in open scholarship would be an important step towards convincing more software authors and site owners to make these longer-lived sessions the default.

Specifically, financial services websites very often keep their session lengths limited to hours or even minutes. On one hand, this choice is an entirely reasonable: an online banking session left logged in on a shared or stolen device can cause immense damage for an individual. Conversely, being prompted for these passwords continually has the very real potential to make those users more susceptible to phishing attacks. When considering the threat models of phishing attacks and physical device takeover, the former can be perpetrated by anyone on the Internet rather than anyone with access to the device, potentially making the phishing concern far greater than the session length concern.

We claim that password re-entry frequency is not only a usability issue, but is also a security issue for all password-based login systems. Our hypothesis is that the more users are asked to authenticate with websites, the more they will experience security fatigue, and as a result become more susceptible to phishing attacks.

We further hypothesize that this effect is not local to the website. In other words, we expect that when users are prompted to frequently authenticate, they become security fatigued in general, on all websites they visit, not only on the specific sites that are prompting users for their credentials.

This study was built to test these hypothesises. We devised a methodology which includes a control group whose login frequency is unmodified, and a treatment group who have the length of their login session shortened, necessitating additional logins to simulate sites with short session timeouts. Unfortunately, we did not find statistical significance in our study, but we were able to replicate important results in phishing susceptibility and to find new but unremarkable results related to well crafted spear phishing attacks.

2 Methodology

This work was carried out in five stages, starting with the development of the software used to measure and manipulate the web browsing experience of the control group, and ending with informing the test participants about the experiment they participated in. This section describes each of these five sections in chronological order.

2.1 Web Browser Extension Development

Browser extensions are pieces of software that are installed in commodity web browsers to measure or modify the user's browsing experience. Though all recent versions of popular web browsers support the ability to write and install extensions, we limited our study to people using Firefox and Chrome.

Our study included two different extensions, one version for the control group that took measurements of the user's browsing activities, and another version for the experiment group that took the same measurements but also modified the browser to induce the user to (re)authenticate with popular websites more frequently than they normally would. Each of theses versions of the extension are described in greater detail in the following subsections.

When the user installed the extension in their browser, she was prompted to enter her email address. The extension then generated a random identifier for the user. The researchers never tied these two identifiers together; we were never able to associate a person's random identifier with her email address. These identifiers were used to identify users during different parts of the study. Finally, on installing the extension, users were randomly assigned to either the control or experiment group, with equal chance.

Throughout the experiment the extensions reported statistics to a central recording server. When the extension reported non-sensitive information to the recording server (such as how long a user has been participating in the study), the extension identified the user by her email address. Likewise, whenever the extension reported sensitive information to the server, the data was tied to the user's random identifier. This allowed us to track which users were still participating in the study, without tying any sensitive information to the participant.

2.1.1 Control Group Extension

The control group version of the extension did not modify the browsing experience at all; it only recorded information about the user's browsing activities. Each of these data points is described below.

User Participation: In order to determine which participants were staying active in the study (and to be able to remind participants if they were not staying active) the extension reported each active user's email address to the recording server once a day.

Pages Visited: In order to determine how regularly different study participants used the web, the extension recorded how many pages each participant visited each hour. The extension did not record *which* pages were visited, only a count of how many each hour. This information was periodically reported back to the recording server with the user's random identifier.

Passwords Entered: The extension also recorded how often users entered passwords on the web. Each time the user entered her password into a password field on the web, the extension recorded the URL of the page (used to determine what types of pages the user trusted with her password), a salted hash of the password (in order to determine if the user reused passwords, without revealing their passwords) and a NIST entropy measure of the password [5] (as a measure of password strength). These values were also periodically reported to the recording server, along with the user's random identifier.

2.1.2 Experiment Group Extension

The extension performed differently for users in the experiment group. In addition to recording the information discussed in Section 2.1.1, the experiment version of the extension also modified users' browsing sessions to cause them to need to authenticate with websites more frequently then they otherwise would.

- https://www.reddit.com
- https://www.facebook.com
- https://www.google.com
- https://mail.google.com
- https://www.tumblr.com
- https://twitter.com
- https://mail.yahoo.com
- https://www.pinterest.com

Figure 1: URLs of sites that participants in the experiment group were induced to reauthenticate with more frequently

We first selected eight popular sites where users needed to login to use the sites primary functionality, listed in Figure 1. When a user logs into one of these sites, the site sets a cookie in the user's browser. This cookie usually lasts for a long time: days, weeks or months. The site uses this information to identify the user to the site, so that the site knows who the user is and does not ask the user to re-login.

The experiment group version of the extension shortened the life span of these cookies to expire on average in 48 hours (some random variation, between plus-and-minus twelve-hours, was added into the cookie expiration times, to make it more difficult for experiment-group members to detect the manipulation). The net result of editing the expiration dates of the cookies is that users would need to re-login to these sites approximately every two days, instead of once a week or once a month.

2.2 Recruitment

Recruitment was conducted through university mailing lists of students, faculty and staff. A person was eligible to participate in the study if she 1) did most of her web browsing on a computer she could install software on; 2) was a student, faculty or staff member; 3) used Chrome or Firefox as her main browser; 4) spent some time on social media sites most days; 5) was at least 18 years old; and 6) was not currently incarcerated.

Participants were required to participate in the study for two months, during which they needed to use the computer with the extension installed at least once every three days. They were told that the study was about "measuring safe browsing practices online", but were not given any further detail about the purpose of the study. They were told that the extension would take anonymous measurements of their browsing habits, and that it would not harm their computer.

Participants were not told that some participants would have their cookies removed earlier than normal, and that they would need to log into sites more often. This deception was conducted with the review and permission of the IRB of the University of Illinois at Chicago.

2.3 Study Participation

Each participant who agreed to the above conditions arranged to meet with a member of the study to guide her through the process of installing the extension on her computer, confirm that she met the eligibility requirements, read and sign a consent form, and document her agreement to participate in the study.

Each participant was offered $30 in Amazon.com gift cards as compensation for her participation, receiving $15 on entering the study and the remaining $15 at the end of the study, if they followed all of the agreed-to terms.

During the study, participants operated their computers as normal, and carried out their typical browsing behaviors. We regularly checked to make sure that the extensions were functioning correctly and that the study participants were still using their browsers at least once every three days. In a few cases, we noticed that a study participant was not using her browser in accordance with

the study's terms. In such cases we contacted the participant by email to see if there was a technical problem, and in a few cases we removed participants from the study who were not able to meet the study's requirements.

2.4 Simulated Phishing Attack

At the end of the two-month study period, we sent an email out, telling the participants that the end of the study was approaching, and that they would need to make an appointment to have the software removed from their computers and be debriefed from the study.

We separately sent all email participants a fake phishing email. The email told all students that they should click on a link in the email to log into their university accounts, in order to complete a brief survey and qualify for the remaining $15 Amazon gift certificate. The email was constructed to only include content that an attacker would have access to and be able to forge.

Notably, the message was sent from a non-university account, which had never been used to interact with the study participants prior. Additionally, the link in the message that participants were asked to click on—and which claimed to be a link to the university's sign-on system—linked to a new domain that was not university owned and which had never been provided to participants before. Finally, when participants clicked on the link and were taken to the false, phishing, version of the university's sign-on system, they were asked to log into a domain that was also not university owned or affiliated with the school.

The fake, phishing version of the university's login page we constructed kept track of which study participants visited the page, how long they stayed on the page, if they entered a user name and password, and if they they submitted the form to attempt to log in.

Each participant who submitted values in the login form was asked to complete a survey. The survey attempted to assess whether she took standard precautions before submitting her university credentials by asking, among other questions, if she checked the URL before entering her user name and password, and if she noticed anything abnormal about the login page's URL.

To protect the participants, we did not record or transmit the entered user name or password over the network; we only recorded how many participants interacted with the page in the same manner one would interact with the true university login system, and if they trusted the false version of the page with their account credentials.

In order to avoid having study participants influence one another or inform each other about the deception in the study, we took care to not reveal the deception for one week, until all participants had a chance to receive and respond to the fake phishing email. If users did click on the link in the sent email, and submitted their credentials to our fake-sign-on page, the messages and web pages they received appeared identical to those they would receive from the true university login system.

2.5 Debriefing

One week after the fake phishing email was sent to study participants, all study participants were sent another email, this time from the university email account that had been corresponding with them throughout the study. The email asked participants to schedule a debriefing meeting with the research assistant. At this meeting, each participant was told about the true purpose of the study, given the chance to ask about the purpose, methods, or outcomes of the research, and provided with the remaining $15 in Amazon credit.

3 Results

We were able to recruit 101 study participants, 89 of which completed the study[2]. Of those who completed the study, 43 were in the control group, and 46 in the experiment group.

3.1 Phishing Susceptibility

Of the 43 participants in the control group, 17 (39.5%) clicked on the link in the phishing email, or otherwise visited the phishing page. All 17 of these control-group members entered some value into the password field on the fake university-login page and submitted the form. In the experiment group, 19 of the 49 (38.8%) participants visited the phishing page, and 18 of them entered some value into the password field and submitted the form. We were not able to find a statistically significant difference between the control and treatment groups.

Participants who submitted the login form were taken to a survey that asked about their participation in the study, if they encountered any technical problems with the browser extension, and if they would like to be notified of the study results when available. Most relevant to the question of phishing susceptibility, participants were asked if they noticed that the URL for the fake university-login form was different from the URL where they normally logged in. Of the 17 members of the control group who completed the survey, 5 (29.4%) stated that they noticed the URL was different, versus 6 of 18 experiment-group members (33.3%) who noticed that the URL was different.

	Control	Experiment
Mean	32.88	35.56
Min	0	0
Max	407	546
St Dev	88.08	64.06

Table 1: Basic statistics on the number of passwords entered by users in the control and experiment groups.

3.2 Password Entry

Finally, the data gathered during this work allowed us to make some measurements about how many passwords participants entered during the two month study, and on how many different domains they submitted passwords. For the eight domains we affected in the study, participants entered on average 34.2 passwords, with our extension inducing users in the test group authenticating more often than users in the control group. Users in the control group entered on average 32.9 passwords on the eight watched sites during the study, while users in the experiment group entered 35.6 passwords to the eight relevant domains.

More broadly, users entered on average 185.74 passwords during the two month study, and submitted passwords to 28.69 domains.

4 Related Work

This work sits alongside other research establishing the effectiveness of phishing attacks as a means of stealing user credentials. Dhamija et al. [2] found that a well constructed phishing page fooled 20 out of 22 test subjects, and that this vulnerability seemed unrelated to demographic or personal characteristics, such as age, sex, or number of hours of computer use. Jagatic [8] investigated how social connections can affect the success rate of phishing attacks, and the success rate of a phishing attack went from 16% of 94 target students to 72% of 487 targeted students when the phishing message was forged to appear to be from a friend or other social contact.

Other research has established that common anti-phishing indicators in browsers do a poor job of alerting users to fraud. Schechter et al. [10] found that factors like the absence of `https` encryption on web pages and missing user-selected site images did not dissuade users from entering their credentials (all 27 users submitted their credentials to the phishing site in the former case, and 23 out of 25 users still did so in the latter case). Similarly, Wu et al. [12] found that even with additional anti-phishing toolbars and utilities installed, 10 out of the 30 participants in their study were still successfully phished. Whalen and Inkpen [11] used an eye-tracking system to determine what security indicators users viewed, and found that unless specifically prompted, users rarely looked at the browser's security indicators. Jackson et al. [7] found that web-browser users did not understand the browser's anti-phishing security warnings, and thus that they offered no protection, unless they received specific training in understanding the browser's indicators.

[2]Seven members of the control group, and five members of the experiment group, exited the study mid-experiment.

5 Lessons Learned

While the core experiment failed to reject the null hypothesis, several of our observations confirm previous studies and can otherwise be useful to the community performing further security-based user studies.

Confirmed very high phishing success rate. The success rate for phishing is high. 40.4% of participants who received the phishing email submitted a password to a untrusted domain, and 97.2% of participants who clicked through the email and visited the fraudulent university-login page submitted their credentials.

This result is possibly due to the priming effect of our phishing strategy (i.e., our subjects were expecting an email regarding payment). However, other factors were also likely at play, including a correctly functioning `https` url, a benign yet similar hostname, and an incredibly low-volume campaign such that typical defenses to prevent deliver of phishing messages would not have been triggered.

Note that we did not investigate whether any of our users had phishing defenses turned on, either directly through their browser or through additional software such as browser extensions or anti-virus programs.

Recruitment challenges. We chose to recruit through our university's email channel for mass advertisement to all faculty, staff, and students. While our university is far more diverse than average in terms of race and socio-economic status, this was still likely to be a less representative group than the general population.

Our reason for recruiting in this manner was that we wanted to ensure a standard phishing experience for the study. Everyone was required to use the same campus single-sign-on infrastructure, which we also required participants to use when selecting a time to meet with us to enter the study. We required this in-person meeting to minimize fraud and to ensure that the extension was installed correctly.

When we recruited in this manner, we had a far lower response rate than we expected, especially given the reward structure for our study. We believe that attempting to minimize participant fraud via in-person meetings was

likely not an effective use of time; other methods of filtering out fraudulent users would likely have been superior.

Extension installation challenges. The proliferation of extension-based malware made it particularly difficult to successfully install the extension. This included turning off various anti-malware features in the browser (temporarily) to successfully install the extension on users' machines.

Asking users to install an extension with such expansive permissions is a lot to ask, even in an IRB-approved, monetarily compensated study. We minimized the amount of data collected and ensured that all data was anonymized and encrypted during storage and transmission. We explained this process in layman's terms during the installation of the extension at the in-person enrollment events.

5.1 Advice for studying phishing susceptibility

For anyone wishing to attempt an experiment like this one, we believe that a few changes would raise the likelihood of observing a correlation—if one exists—between login frequency and phishing susceptibility. First, we believe that targeting more websites (or even doing so in a site-agnostic fashion) would be beneficial, as well as allowing for the collection of information about password entry more broadly.

Second, we recommend conducting further research in a way that can control for different amounts of natural (i.e., pre-test) password re-entry. The best way to control for this would likely be to recruit more research subjects. Allowing participants to sign up remotely would be a boon in this respect. However, it might filter for more technically savvy users who are able to install extensions on their own.

The effect of saved passwords or password managers would be an interesting angle to investigate. These tools associate the saved logins credentials with specific sites, and so phishing sites would not be auto-filled. Controlling for this effect would be important, as this process can drastically reduce the number of passwords typed by a user over a given amount of time.

References

[1] Bonneau, J., Herley, C., Van Oorschot, P. C., and Stajano, F. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Security and Privacy (SP), 2012 IEEE Symposium on* (2012), IEEE, pp. 553–567.

[2] Dhamija, R., Tygar, J. D., and Hearst, M. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (2006), ACM, pp. 581–590.

[3] Downs, J. S., Holbrook, M. B., and Cranor, L. F. Decision strategies and susceptibility to phishing. In *Proceedings of the second symposium on Usable privacy and security* (2006), ACM, pp. 79–90.

[4] Egelman, S., Cranor, L. F., and Hong, J. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008), ACM, pp. 1065–1074.

[5] Guideline, N. E. A. Nist special publication 800-63 version 1.0. 2, 2006.

[6] Hong, J. The state of phishing attacks. *Communications of the ACM 55*, 1 (2012), 74–81.

[7] Jackson, C., Simon, D. R., Tan, D. S., and Barth, A. An evaluation of extended validation and picture-in-picture phishing attacks. In *Financial Cryptography and Data Security*. Springer, 2007, pp. 281–293.

[8] Jagatic, T., Johnson, N., Jakobsson, M., and Menczer, F. Phishing attacks using social networks. *Indiana University Human Subject Study*, 05–9892.

[9] Khonji, M., Iraqi, Y., and Jones, A. Phishing detection: a literature survey. *Communications Surveys & Tutorials, IEEE 15*, 4, 2091–2121.

[10] Schechter, S. E., Dhamija, R., Ozment, A., and Fischer, I. The emperor's new security indicators. In *Security and Privacy, 2007. SP'07. IEEE Symposium on* (2007), IEEE, pp. 51–65.

[11] Whalen, T., and Inkpen, K. M. Gathering evidence: use of visual security cues in web browsers. In *Proceedings of Graphics Interface 2005* (2005), Canadian Human-Computer Communications Society, pp. 137–144.

[12] Wu, M., Miller, R. C., and Garfinkel, S. L. Do security toolbars actually prevent phishing attacks? In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (2006), ACM, pp. 601–610.

A Appendix: Source Code

This section includes links and descriptions of the code used in the system. All code described here is managed in public git repositories.

- **Browser Extension**
  `git@github.com:snyderp/uic-phishing-extension.git`
  Javascript code used to build the Firefox and Chrome browser extensions.

- **Recording Server**
  `git@github.com:snyderp/bits-phishing-server.git`
  Python web server that records the information sent by each browser extension.

- **Phishing Server**
  `git@github.com:snyderp/bits-phishing-survey.git`
  Python web server that implements the simulated

phishing attack on the university's single sign-in system, along with the debriefing study.

- **Signup Server**
  `git@github.com:snyderp/bits-phishing-signup-server.git`
  Python web server that runs the system that publishes information about the study, allows users to signup for the study, and verifies that users meet the study participation requirements.

Towards robust experimental design for user studies in security and privacy

Kat Krol, Jonathan M. Spring, Simon Parkin and M. Angela Sasse
University College London

Abstract

Background: Human beings are an integral part of computer security, whether we actively participate or simply build the systems. Despite this importance, understanding users and their interaction with security is a blind spot for most security practitioners and designers.
Aim: Define principles for conducting experiments into usable security and privacy, to improve study robustness and usefulness.
Data: The authors' experiences conducting several research projects complemented with a literature survey.
Method: We extract principles based on relevance to the advancement of the state of the art. We then justify our choices by providing published experiments as cases of where the principles are and are not followed in practice to demonstrate the impact. Each principle is a discipline-specific instantiation of desirable experiment-design elements as previously established in the domain of philosophy of science.
Results: Five high-priority principles – (i) give participants a primary task; (ii) incorporate realistic risk; (iii) avoid priming the participants; (iv) perform double-blind experiments whenever possible and (v) think carefully about how meaning is assigned to the terms *threat model*, *security*, *privacy*, and *usability*.
Conclusion: The principles do not replace researcher acumen or experience, however they can provide a valuable service for facilitating evaluation, guiding younger researchers and students, and marking a baseline common language for discussing further improvements.

1 Introduction and aims

Security mechanisms are incorporated into IT systems to protect them or the information they contain. Protection can extend to the regular activities of users and the systems they interact with, relying on them to behave in a secure manner. In the past, humans have been referred to as the "weakest link" in security, where insecure actions – be they malicious or unintended – can jeopardise systems and expose them to threats [55]. Research from 1999 onwards [1, 64] has shown that the systems themselves can introduce security weaknesses, by being *unusable* and a *bad fit* to the tasks performed by users of those systems [53], in turn making secure behaviour difficult. Usability is then an important factor in the design and deployment of security mechanisms, where treating usability as an after-thought to be added to an existing system can instead impact security [65].

Particularly for complex IT systems, users tend to fail – rather than knowingly refuse – to comply with security expectations [33]. Potential reasons why users do not comply include: security-compliant behaviour demands too much of them, the need to comply is not obvious to the individual, or the definition of compliant behaviour is simply unworkable. In all of these cases, individuals may rationalise that behaving securely is not worth their time or effort if there are no perceived personal benefits [24]. The result of this rationalisation is often that users develop coping strategies to reduce the demands of security, work around security systems, or become disenfranchised with security if it continues to act as a distraction and a barrier [2]. For instance, users may rationalise when considering whether to maintain a written note of a password as a recall aid, even though humans are not adapted to such memory tasks.

The users' efforts to make use of the system, by rationalisation or otherwise, often conflicts with the security architects' efforts to secure the system [61]. User efforts to cope with the demands of security are only exacerbated by security architects who insist that users can be trained to perform security tasks (e.g., [6]), even though usable security research demonstrates that there are cases where this is untrue [9]. Individuals and groups develop their own alternatives to security if the security architecture does not accommodate the users' security needs. Often groups use their own approximation of what a secure

system should do while attempting to respect the need to behave securely [34]. Mandates and restricted systems further undermine users when security managers and designers do not understand the user [25, 52]. Thus the user aspect of the system cannot be avoided, ignored, or designed out by the security architect.

The body of evidence in security usability is growing both in general knowledge on how users make security and privacy choices and also for the use and challenges of specific technologies, such as encryption [64, 57, 49]. One notable development was the definition of "Grand Challenges" for achieving user-centred security in 2005 by Zurko [67], stating that for those developing secure systems:

> "The body of experience testing the usability of security both in the lab and in context will define the techniques and tools we need and can use. It will also generate a body of best practice we can begin to systematize in checklists and expert evaluations."

In this paper, we offer such a systematisation by producing a set of principles for user studies of both security and privacy. The principles complement each other, and are interdependent. Such a set of principles can support the development of robust study outcomes, comparison of results across user studies, and composition into a meaningful body of evidence centred around the users' security technology experience. We review a concise set of experiments studying user security technology use. The result of the review is five *principles* which can be reviewed in advance of performing a study or to help guide evaluation of past studies.

Zurko [67] states that explicit security mechanisms that are incomprehensible to users and which are not integrated with the task are not effective. There is then a need to capture end-user understanding of security and how security fits with their activities. Researchers who consider these principles will have a language to express assurance that their study is applicable, consistent, reliable, and should be believed. Furthermore, the principles assist in establishing relationships among outcomes of different studies on elements such as: identifying user needs, user risk profiles, impact of using particular technologies, and the impact of certain more-or-less controlled conditions of use. Findings can then be collated within specialised frameworks, where efforts are already underway in the research community – these include the "human in the loop" framework developed by Cranor in 2008 [13], and a repository of behavioural science findings as relate to IT-security [44].

The paper is arranged as follows: Section 2 describes background on general rules of research validity; Section 3 describes the process followed to derive the principles; Section 4 details the recommended study principles for examining the usability of security and privacy technologies. Discussion follows in Section 5, followed by Conclusions.

2 Background

Hatleback and Spring [22] identify and explain four desirable experiment design features by analogy with biology that should apply to experiments in computing and computing security, analogous to principles proposed specifically in malware research [51]. The four principles are:

Internal validity: The experiment is of "suitable scope to achieve the reported results" and is not "susceptible to systematic error" [22, p. 451]

External validity: The result of the experiment "is not solely an artifact of the laboratory setting" [22, p. 451]

Containment: No "confounds" in the results, and no experimental "effects are a threat to safety" of the participants, the environment, or society generally [22, p. 452]

Transparency: "there are no explanatory gaps in the experimental mechanism" and the explanatory "diagram for the experimental mechanism is complete" in that it covers all relevant entities and activities. [22, p. 452]

These four terms come from a background of experimental and quantitative research. In considering robust experimental principles in the junction between IT-security practice and behavioural sciences research, Pfleeger and Caputo [44] suggest that steps be taken to reduce confounding variables and biases, to then support transferability. Qualitative and case-study based research methods have analogous principles, although they are derived and ensured differently. The qualitative research methods tradition roughly makes the following translations. Validity [43] usually refers specifically to internal validity, whereas transferability (or generalisablity) [30] maps to external validity, trustworthiness [40] and credibility [29] map to transparency, and containment is expressed as ethics in research design and execution [11].

A famous principle in philosophy of science is falsifiability: the idea that good theories are those whose truth can be tested and hypothetically fail [45]. For Popper, a good experiment is taken to be one that tests an existing theory. Successful theories pass more tests than others. However, one must immediately ask how to design such an experiment, and what features it should have. Further, when two results conflict, we must evaluate the

strength of evidence provided by each experimental result [39, p. 146]. Our principles provide positive, constructive guidance for both these processes that the falsifiability concept does not directly supply. That is, these principles provide something of the "how" to experiment design.

3 Methodology

Our methodology for selecting which principles to discuss followed two phases. First, we oriented our search in broad strokes by the categories of desirable features introduced in the Section 2. These categories are derived from scientific investigation more broadly and enjoy general support across multiple disciplines, and thus provide a reasonable starting point. They are also explicitly very general, and so require more detailed specification for challenges common to our particular field of interest, user studies in security and privacy.

The second phase of our methodology was to evaluate and select principles based on our expertise and experience. We thought it important that studies should be realistic, bias as little as possible and use precise language and we structured these high-level goals into principles. The process of specifying them was as follows. First priority was that we had personal empirical experience designing studies that meet these principles and overcome the underlying challenge. Thus we prioritised principles for which we feel we have an adequate and accurate formulation. We also selected principles based on our perception of the importance to a high-quality study, where high-quality means one can explain adequately how it meets the four high-level rules of internal validity, external validity, containment, and transparency. These properties are not commensurate, that is they are not measured in equivalent units and thus are not directly comparable. Therefore we employed expert judgement to decide whether a certain drop in containment from one common challenge is more or less damaging to study quality than some certain drop in external validity, for example. We did not consider a rubric or other counting exercise to adequately help our principle selection process; the qualities are too contextual and rich to so easily put into bins.

Thus, one may reasonably disagree with our choice of principles. We would like to stress that the five principles we propose are what we have found important in our research and they might not be applicable to all types of studies and all research areas within usable security and privacy. As part of future work, our proposed principles should enable quantitative, empirical measurement of study outcome differences to further improve study design understanding. This experience-first strategy is more likely to produce an incomplete list, but our recommendations are more likely to be accurate. Secondly, we emphasise our set of principles is not meant to be exhaustive, but only accurate and high-value recommendations. From this point of view, additional principles of equally high quality are welcome from the rest of the community based on others' study design experience. At present, we prefer to be somewhat conservative in our coverage and confident the principles we recommend are accurate and useful.

4 Results: Principles

Our principles come from one of two angles, roughly those from security studies and those from user studies. More specifically, our principles arise from the following two challenges, (1) subject-matter-specific problems common to security generally interacting for the first time with techniques for exploring user experience, and (2) general challenges of human experience studies that have particularly pervasive or damaging impact when they arise in security usability. We propose five principles for robust experiment design; the first three are security subject-matter issues, and the second two are general user experience study problems with particular impact for the applicability of behavioural research outcomes in IT security [44].

- Give participants a **primary task**
- Ensure participants experience **realistic risk**
- **Avoid priming** the participants
- Perform experiments **double blind** whenever possible
- **Define** these elements precisely: threat model; security; privacy; usability

These five principles for robust studies in usable security are best viewed as subject-matter specific elaborations of desirable experiment-design elements for robust study design from across the sciences. Although in some important ways these principles are not new because they are based on existing ideas, in other important ways these recommendations are unique because they have been tailored to the specific challenges common to user studies in security and privacy.

The genesis of our principles is learning from experiments we have participated in or have read about; in parallel, the target impact of our proposed principles is to better learn from experiment. We have generated the principles by learning from shortcomings in our own experiment designs and those of others. However, in order to best learn from experiments in the future it would be prudent to follow the principles as recommendations or

heuristics for overcoming some of the most common errors that arise in usable security studies; thus our principles are open to future revision, addition, and amendment as warranted.

We use past work as a series of case studies to elicit these principles, and use analogy to existing literature in other fields as evidence that our conclusions are robust and not mere idiosyncrasies of the cases used. Usable security makes use of methods from both qualitative and quantitative research disciplines. A measurement study may be used on a subset of participants to examine the extent to which the values reported by the humans match objective values of behaviour captured by a sensor. For example, privacy studies have repeatedly identified a discrepancy between reported preferences and actual behaviours (e.g. [60, 28]). Therefore we make use of the study design principles from both traditions, where appropriate. Subsequently, we describe how each of the five principles relates to widely-accepted generic principles of good study design.

4.1 Primary task

By giving participants a primary task in a study we make sure they are put in a realistic situation. In real life, people use computers in order to accomplish some task, be it to send an email, make a purchase or search for information, and so security as a task is secondary to a main purpose.

In usable security research, Brostoff and Sasse [10] were the first to have a primary task in their study. In a 3-month field trial, 34 students used an authentication mechanism called Passfaces to access their course materials. Although users were positive about the idea behind Passfaces when asked about it, a 3-month trial of participants using Passfaces and passwords in practice painted a different picture. The results show that the frequency of logging in to the system dropped when Passfaces guarded access to the system; participants logged in with one third of the frequency when they authenticating using a grid of Passfaces rather than passwords. Since logging in using Passfaces took longer, more recent research of security behaviours (e.g., [63]) would in retrospect imply that participants decided that it was not worth their time to spend a minute logging in only for a few minutes of work on a system.

Giving participants a primary task while we study their security behaviour is related to two important features in usable security. First, users in the real world have a primary task which is interrupted by performing security tasks. Including a primary task makes sure the experiment simulates the real world accurately enough to be meaningful; this is a form of ensuring external validity or transferability. A primary task adds external validity in another way, namely we know from psychology that human attention and other mental resources are bounded [59], where such bounds can impact security [3]. Further to this, users would rather achieve their goal than be distracted by secondary tasks that divert their attention from the primary task [63]. Herley [19] urges security designers to be mindful of how much security-related effort is demanded of users, and to use what is available to them wisely. Having the full mental resources of a participant available for a security task in a study setting does not necessarily translate to that person wanting – or being able – to devote their mental resources solely to security in a more realistic setting.

Giving participants a primary task in a study is not always appropriate. For example, user testing of a new authentication mechanism is a multi-stage process from requirements gathering to evaluation post-adoption. At one of the early stages, it is advisable to conduct a performance study with users to assess if a security-related task is achievable. For example, it would be confirmed whether it is possible to read and enter the digits from an RSA token into an entry field within the defined expiry window for a generated series of digits. In research on CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart), many studies have focused on establishing if a paricular type of CAPTCHA is decipherable (e.g., [20]). Ideally such performance evaluations would be complemented with studies to assess user acceptance and suitability in real-life interactions to assess whether solutions are viable in genuine user populations.

Creating a primary task is difficult, and requires time and effort. Preibusch [47] provided a guide on how to study consumers' privacy choices in the tradition of behavioural economics and advocated using real-world shopping scenarios as a main approach. Researchers should create real shopping scenarios to study privacy choices; in this instance real means the participant can browse in an online shop, buy and pay for goods, and receive them. Studies in privacy have used primary tasks including purchasing gourmet goods [46], DVDs [5] and cinema tickets [31]. Some examples of primary tasks used in security studies include asking participants to buy goods from online retailers (e.g., [4]) and evaluate a tool for summarising academic articles [36]. While all experiments mentioned above were confined to a university laboratory, researchers are also increasingly conducting security experiments in the wild. A notable example is a field experiment by Felt et al. [17] which tested six proposed SSL warnings in Google Chrome and recorded 130,754 user reactions. The research has superior methodology since the behaviour of actual users is recorded as they are going about their daily online activities, and there is arguably no better primary task than

the actual one that a user naturally chooses to do themselves. However using such superior methodology is not within reach of most academics and more collaborations between industry and academia are necessary to make such studies possible.

4.2 Realistic risk

Like the importance of a primary task, a realistic risk is part of the design principle of providing a realistic task environment for study participants, because the potential for real consequences is part of a realistic experience of security. Enumeration of the risks of usability failure is also important to the design of secure systems [67]. Participants are under 'realistic risk' when they perceive there is a threat to the security and privacy of their own information.

An experiment should introduce realistic risk to participants because people behave differently if they know a situation is a simulation. Lack of a realistic risk threatens external validity, where this threat stems from the fact that participants' perception of risk is one of the things (implicitly) being tested, and it changes if participants know a situation is a simulation. In this sense, lack of realistic risk causes the experimental results to be solely an artifact of the laboratory setting, with no adequate analog in the real world, and so transferability or external validity is undermined. Participant risk perception variability also represents a threat to the internal validity of a study when participants are exposed to different perceived risks without measuring, controlling, or monitoring those differences during the study.

Studies have introduced realistic risk to participants in different ways. Schechter et al. [54] (described in more detail below) asked a group of their participants to log in with their actual credentials to online banking. In a study by Beresford et al. [5], participants were purchasing DVDs and entered their own details to complete the purchase and have the products shipped.

The impact of using participants' actual credentials has been tested directly. Schechter et al. [54] tasked their participants with performing different online banking tasks, and manipulated a range of different website authentication measures such as HTTPS indicators and website authentication images. A group of participants in their experiment used their actual credentials while others role-played with simulated credentials. The researchers found that those participants who used their own credentials in the experiment behaved more securely than those using credentials provided for them.

In a study by Krol et al. [36], participants brought their own laptops to the laboratory and if they downloaded a file despite a security warning, it could have potentially infected their own computer with a virus. In interviews afterwards, a few participants stressed that if they have to download something from an untrusted source, they would do it on a public shared computer in order not to put their own machine in jeopardy. However, owning the laptop is not the only element of realism perceived by participants as 29 out of 120 participants said they considered the laboratory a trusted environment and assumed that the researchers checked the files beforehand and would not let them download something malicious. This fact highlights the need for continued assessment of users' perceptions of risk, both before and after studies, to improve researchers' interpretation of results and understanding of user attitudes.

Obviously inserting a realistic risk into a study protocol causes an interesting trade-off with containment, however, that must be addressed through the institutional review board (IRB) process. If the IRB is unfamiliar with the relevant technologies, the Menlo report [14] provides a framework for deciding whether the study poses too much of a threat to prospective participants. The Menlo report elaborates four principles for information and computer technology (ICT) research: respect for persons, beneficence, justice, and respect for law and public interest. These principles are based on ethics in biomedical studies but are thoroughly adapted for the ICT context.

The challenge of creating an ethically sound study with a realistic risk may lead a researcher to opt for sacrificing the external validity and ignoring this principle. Usually when a study sacrifices external validity it is to gain internal validity, losing representativeness in order to more carefully control the effects being studied. When internal and external validity are exchanged in this way, it immediately suggests a family of studies, some with strict controls and some descriptions of the real world and a gradient of more or less controlled studies in between, that could be synthesised in order to provide appropriate explanation for the phenomena. Relaxing the principle of a realistic risk to the participant does not provide such an exchange; if anything it negatively impacts both internal and external validity. The research into security usability has up to this point done a great deal of work in identifying factors which influence security behaviour – increasingly research is finding that the properties or severity of these factors can encourage a particular response to security. Individual utility of security can be influenced by factors personal to them; for example, the complexity and number of passwords that a person must manage can – once it reaches a perceived 'limit' – encourage the reuse of passwords or a reliance on recall aids such as written notes [2, 24].

4.3 Avoidance of priming

Priming is exposing participants to some information that might influence how they subsequently behave in the experiment. Non-priming the participant helps avoid biases such as demand characteristics where the participant gives answers based on what they believe the experimenter expects of them. Non-priming is an issue of internal validity, but also containment if the researcher comes into possession of personal or otherwise sensitive information. Non-priming can be achieved by simply not telling participants much about the purpose of the study, it can range from keeping the study description general to actively telling lies to participants. A common way to avoid priming is to deceive participants about the actual purpose of the study. Deception has been used in our field of research; Egelman et al. [15] advocate deception for user studies in security and privacy to produce generalisable findings. Krol et al. [36] told their participants they were examining a summary tool for academic papers where in reality they studied participants' reactions to download warnings.

Again ethical questions arise from the fact that participants are lied to. Psychology has traditionally dealt with this dilemma by requiring researchers to debrief participants at the end of the study and tell them what the actual purpose of the research was. However researchers have warned about potential negative consequences that might arise from deception. Horton et al. [27] emphasise that using deception can make participants distrust researchers in future studies. Researchers in the field of economics tend to avoid deception altogether as this could falsify the research results [26].

4.4 Double blind

In a double blind experiment, both the participant and the person executing the experiment do not know details of the study – this limits the capacity for either party to influence the study outcomes through knowledge of the study design itself. Traditionally used in medicine [50], the person executing the experiment would not be informed as to whether a patient is receiving an active medicine or a placebo. In this way, the designers of a medical trial hope to avoid a situation where an experimenter administering medicine treats the subject differently or influences the results in any other way. Double blind experiment design can improve internal validity and containment by preventing accidental transmission of biases, research goals, or sensitive information between the researcher and the participants.

To the best of our knowledge, experimental procedures using double blind have been used only once so far in usable security and privacy research. Malheiros et al. [41] studied individuals' willingness to disclose information in a credit card application. They employed three undergraduate psychology students to conduct experimental sessions. The students were told that the study was exploring individuals' willingness to disclose different types of information on a loan application for an actual loan provider. In reality, the study was looking at participants' privacy perceptions.

As previously, there are ethical considerations with not telling the entire truth not only to participants but also the person executing the experiment. Running a useful double-blind experiment introduces challenges to the experiment design. For example, if the person who executes the experiment is unaware of the purpose of the study, they for example cannot ask specific questions in response to the participant's behaviour, which may be valuable to adequately interpret the participant's behaviour in situ. Debriefing at the end of the study session might not be possible as the experimenter is not adequately prepared to do so themselves, and another researcher would need to be present to debrief the participant. Such an approach would further require that the experimenter be debriefed at the end of the study.

4.5 Define: threat model, security, privacy, usability

There are two important ways in which the researcher must carefully attend to how meaning is assigned to terms during explanation and during execution. Firstly, terms must have precise and well-defined meanings when articulating the design, protocol and results of an experiment to colleagues; secondly and more subtly, the researcher should be careful not to bias participants by priming them with definitions provided during the course of a study. In the first case, being clear and consistent with definitions during experiment design and execution improves internal validity. This reduces the chances for error or imprecision that would lead to systemic design flaws, as would result from confusion of similar concepts that are actually distinct at the detail level of experimental examination. Clear definitions improve transparency, trustworthiness, and credibility when describing and explaining an experiment. In the more subtle case, it is generally desirable that the researcher not provide any definitions of terms to the subject participants, to avoid biasing the participants' answers. This sense of attention to definitions overlaps heavily with the avoidance of priming, discussed in more detail in Section 4.3.

The terms we find to be most commonly impacted by definitional problems are *threat model*, *security*, *privacy*, and *usability*. These words are central to all research in the field, so it is both unsurprising and troubling that the terms are hard to define. Definitional disputes about the

term information continue in information science, for example [66]. The difficulty is unsurprising because research in any field can be interpreted as wrestling with creating precise agreement for defining the terms and relations among them that adequately describe the mechanisms under study. The lack of definition is simultaneously troubling because lack of specificity prevents a genuine discussion about the merits of competing definitions to capture the mechanism adequately and instead hides behind ambiguity. Researchers should consider and contrast different terms in forming their own understanding to promote their ability to support study participants in articulating their own perspectives.

When articulating the design, protocol, and results of studies, researchers should take as a starting point the most widely agreed upon definitions. Shared definitions are critically important to a well-functioning research culture and community because without shared definitions we cannot genuinely compare results among studies. Appropriate international standards bodies include IETF (Internet Engineering Task Force), IAB (Internet Architecture Board), ISO (International Organization for Standardization), and IEEE (Institute of Electrical and Electronics Engineers). If these starting points are insufficient, then the researcher has a firm point of departure to explain why this is so; however, redefinition must be careful and must ensure usage does not slide between different definitions of a term without noting so doing. The security glossary from the IETF is an informational document that provides an excellent starting point [58]. Departures from definitions should be clear and justified. Therefore it is worth excerpting from the RFC for each of the terms to discuss common departure points.

Threat is "a potential for violation of security, which exists when there is an entity, circumstance, capability, action, or event that could cause harm" [58, p. 303]. Note that this does not define *threat model*, which is the set of threats and countermeasures considered relevant to the system at hand. Considering the relevant set of threats is essential for the external validity of a study.

Security is "a system condition that results from the establishment and maintenance of measures to protect the system," where the measures taken are suggested as deterrence, avoidance, prevention, detection, recovery, and correction [58, p. 263]. Studies often must contribute to a specific aspect of security, as it covers a broad range of activities. Authors would do well to specify which measures or aspects of the system condition of security on which their study focuses.

Privacy is "the right of an entity (normally a person), acting in its own behalf, to determine the degree to which it will interact with its environment" [58, p. 231]. This term has a particularly rich history of being difficult to define cleanly. For a comprehensive overview of the different definitions of privacy see the work of Gürses [21].

Usability is not directly defined by the IETF, however it is referenced as one of the two requirements for the availability pillar of the classic confidentiality-integrity-availability triad. Availability is "the property of a system or a system resource being… usable… upon demand…" [58, p. 29]. This supports the idea that, if availability is a requirement, an unusable system cannot be secure. Meanwhile, the failure of the standards to have even an informational definition of usability while giving it such a prominent position serves to highlight the importance of research in usable security. The usable security community cannot contribute to filling this gap for the Internet community as a whole if we are not clear about our own definitions.

A definition of *usability* is provided in the ISO 9241-11 standard for "office work with visual display terminals", as the *"Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use"*. In studies, the inclusion of a primary task then provides an approximation of the context of use, against which to measure these qualities.

Although researchers need to be clear when communicating their definitions to peers, while conducting studies the researcher should not provide definitions to the participants when participant perceptions of these terms are being studied. Providing or sanctioning responses threatens the study because it injects a systemic error in the form of the researcher's pre-conceived definitions, threatening internal validity. Methods for avoiding even accidental transfer of ideas from the researcher to the participant are discussed in Section 4.3 and Section 4.4 on avoiding priming and performing double blind experiments, respectively.

We are often studying the definitions, because security and privacy mean different things to different people. The gap between security architect and user definitions of security is demonstrated by an example study on CAPTCHAs. Krol et al. [37] asked participants to make purchases on a ticket-selling website and part of the check-out process was to solve a CAPTCHA. After the purchase, participants were interviewed about their experience. In the security community, the security of CAPTCHAs is considered in terms of them being solvable by humans and not by robots. This is to protect the system from automated attacks leading to for example unavailability of the service to actual users. In the study, when participants mentioned security they did not speak about how well the CAPTCHAs protected the services but worried about the security of their own accounts and personal data.

Defining basic terms can be difficult as their meaning is often contextual. The challenge is to be precise, con-

sistent and open to discussion with others.

4.6 Additional considerations

There are two areas which we treat as additional considerations, which are important but for which we have no direct recommendations: sampling bias, and the impact of current events on participants' perception and comprehension of security.

4.6.1 Sampling bias

An important consideration that we did not include in the principles is *sampling bias*. Sampling bias is, roughly, when the sample studied in an experiment is not relevantly representative of the population to which the researcher generalises their results. Therefore it is a type of threat to external validity. Sampling bias is a common scientific problem which has been studied in both psychology and information security, and thus it should not be surprising that user studies in security and privacy also contend with sampling bias.

In the larger fields of psychology and security, sampling bias has been studied in different ways. Many psychology research studies have had undergraduate students as participants, where a study with a mean participant age of 19 is not uncommon. Further, studies in psychology often rely on participants drawn from Western, educated, industrialised, rich and democratic (WEIRD) societies. Heinrich et al. [23] showed that these participants are not representative of all humans and are often outliers. In psychology, it often appears to be the richness or complexity of human individuals or systemic cultural differences that drives sample bias concerns. In information security, sampling bias is more often treated as an artifact of the sensor choices or as an artifact intentionally inserted by the adversaries being studied [42]. Sampling bias in information security may be assessed by technical measures with individual components which compare the whole sample to the population of available properties, such as total viable IP addresses [62]. However, like psychology, the argument for what qualifies a sample as sufficiently or relevantly unbiased must be made on a case-by-case basis.

In the field of usable security, sampling bias has already been discussed in at least two ways. Firstly, there has been a discussion as to whether samples drawn from crowd-sourcing platforms are representative of the wider population [32, 56]. Secondly, some studies have focused on the security and privacy of hitherto understudied populations. For example, Elliott and Sinclair Brody [16] studied the security and privacy needs of Afro-American New Yorkers. Bonneau and Xu [7] studied how character encoding can influence the choice of passwords for English, Chinese, Hebrew and Spanish speakers.

In our own studies, we have used pre-screening to maximise diversity of samples to include users of different age groups, gender and educational backgrounds (e.g., [38]). Pre-screening is our best effort to match our study sample to the population who uses the technology we are studying. In many cases, we do not know what subset of the whole human population is actually our target population of users, which makes targeting the correct sample particularly difficult. We have not had participants from non-WEIRD population groups in our own studies, thus we feel less qualified to talk about how to address this. How usable security is different from other disciplines in this respect requires further investigation, which is why we have not listed formal recommendations in this regard as we have for our five principles. Our general advice is to be mindful of the bias of one's research sample and not to frame any results as if they were universally applicable, but rather to frame results as applicable only to the population group(s) studied.

4.6.2 Current events

Participants often need to be considered in their social context, rather than as isolated individuals. While this tension is common between psychology and sociology, it is particularly important in security and privacy user studies because the social backdrop of information security has been changing so rapidly over the last two decades.

In Section 4.5, we elaborated on definitions of threat model, security, privacy and usability as a means to support dialogue with study participants as to what these concepts mean to them. Similarly, participants will develop an understanding of these concepts from the environment around them. Rader and Wash [48] found that people develop knowledge of security from both incidental and informal sources. Those who proactively learnt about security would learn – through sources such as news articles – about how to protect themselves from a range of attacks; others would learn incidentally by way of stories from others, sharing ideas primarily about the kinds of people who might attack them. Participants' perspectives about security may then be shaped by current events as they are documented and discussed with others. This may then require researchers to in some way be mindful of the current events around the time of a study – participants' perceptions may not be stable over time (and distinct studies) as the outside world changes.

5 Discussion

In this section, we discuss how researchers and practitioners could apply these principles. The principles do not replace researcher acumen or experience, however they provide a valuable service for facilitating evaluation, guiding younger researchers and students, and marking a baseline common language for discussing further improvements.

Our list of recommendations does not trivialise the difficulty of the research to be done. There is no replacement for the researcher's experience and skill; yet best principles to check for when designing, executing, or evaluating an experiment help in other critical ways. In weighing the advantages and disadvantages of checklists as a component of repeated procedures, Klein [35] notes that checklists should not supplant expertise but can be used to break complex procedures into repeatable steps. Most surgeries use checklists, but this improves patient outcomes only when hospital staff are properly trained and understand the checklist [12], unsurprisingly. Similarly, our mere process of listing these principles is not sufficient to improve research outcomes.

It may be that researchers in the field of usable security and privacy combine experiment tools to respond to the principles. Bravo-Lillo et al. [8] for instance have developed a reusable research ethics framework. Ferreira et al. [18] use a formal modelling technique to define technical and social threats as a precursor to designing and running experiments which involve human participants – such an approach may be applied to define the *threat model* for a study.

Security practitioners and developers of automated IT systems may want to account for the user when building security mechanisms that require human interaction – research that considers the principles can be more readily applied within a repeatable framework as advocated by Cranor [13]. Study of security alongside a primary task can identify communication impediments; realistic risk can characterise personal variables; clearly-articulated threat models can convey how behaviour and security mechanisms under evaluation respond to anticipated attacks.

6 Conclusions and future work

The five principles presented here provide an excellent example of learning from past experiments in order to produce incrementally better experimental designs going forward. Although we do not claim that the principles are exhaustive, they provide a fruitful starting point for reflecting on experimental design principles within the specific subfield of usable security and privacy research. The principles of primary tasks, realistic risks, avoiding priming, conducting double blind studies, and defining terms are reasonably intuitive from surveying the literature and have demonstrated benefits.

We recommend that anyone designing an experiment in usable security and privacy considers these principles carefully. If, after consideration, the researchers decide one or more principles do not apply to their study design, we simply recommend that they explain why when reporting their studies. This also serves to more concretely define the validity of subsequent study findings relative to the work of others in the field and in the wider world of security practice.

The work of describing principles that are important to experiments and other structured observations within a field is never done. The process is iterative; as helpful principles are applied more widely in new studies, new challenges will arise as old best principles are mastered. To facilitate such advancement of the field, future work should continually analyse the trade-offs between internal validity and external validity and the challenges of providing transparency and containment. With an eye keen to these potential problems, we can catalogue both further study designs and their impacts upon the capacity to capture user experiences of security technologies.

Acknowledgments

Kat Krol is supported by a Secure Usability Fellowship from the Open Technology Fund and Simply Secure. Jonathan M. Spring is supported by University College London's Overseas Research Scholarship & Graduate Research Scholarship. Simon Parkin is supported by UK EPSRC, grant nr. EP/K006517/1 ("Productive Security"). The authors also wish to thank the LASER Workshop's Program Committee, their paper shepherd and attendees of the workshop for their support in preparing the paper. Steve Dodier-Lazaro and Sören Preibusch provided useful feedback on earlier drafts of the paper.

References

[1] Adams, A., and Sasse, M. A. Users are not the enemy. *Communications of the ACM 42* (1999), 40–46.

[2] Beautement, A., Sasse, M. A., and Wonham, M. The compliance budget: Managing security behaviour in organisations. In *Proceedings of the 2008 workshop on New security paradigms* (2009), ACM, pp. 47–58.

[3] Benenson, Z., Lenzini, G., Oliveira, D., Parkin, S., and Uebelacker, S. Maybe poor Johnny really cannot encrypt – The case for a complexity theory for usable security. In *New Security Paradigms Workshop (NSPW'15)* (2015).

[4] Berendt, B., Günther, O., and Spiekermann, S. Privacy in e-commerce: stated preferences vs. actual behavior. *Communications of the ACM 48*, 4 (2005), 101–106.

[5] BERESFORD, A. R., KÜBLER, D., AND PREIBUSCH, S. Unwillingness to pay for privacy: A field experiment. *Economics Letters 117*, 1 (2012), 25–27.

[6] BONNEAU, J., AND SCHECHTER, S. Towards reliable storage of 56-bit secrets in human memory. In *Proc. USENIX Security* (2014).

[7] BONNEAU, J., AND XU, R. Of contraseñas, sysmawt, and mìmă: Character encoding issues for web passwords. In *Web 2.0 Security & Privacy* (May 2012).

[8] BRAVO-LILLO, C., EGELMAN, S., HERLEY, C., SCHECHTER, S., AND TSAI, J. You needn't build that: Reusable ethics-compliance infrastructure for human subjects research. In *Cybersecurity Research Ethics Dialog & Strategy Workshop* (2013).

[9] BROSTOFF, S., INGLESANT, P., AND SASSE, M. A. Evaluating the usability and security of a graphical one-time PIN system. In *BCS Interaction Specialist Group Conference* (2010), British Computer Society, pp. 88–97.

[10] BROSTOFF, S., AND SASSE, M. A. Are Passfaces more usable than passwords? A field trial investigation. *People and Computers* (2000), 405–424.

[11] CHEEK, J. Research design. In *The Sage Encyclopedia of Qualitative Research Methods*, L. M. Given, Ed. SAGE Publications, 2008, pp. 762–764.

[12] CONLEY, D. M., SINGER, S. J., EDMONDSON, L., BERRY, W. R., AND GAWANDE, A. A. Effective surgical safety checklist implementation. *Journal of the American College of Surgeons 212*, 5 (2011), 873–879.

[13] CRANOR, L. F. A framework for reasoning about the human in the loop. *UPSEC 8* (2008), 1–15.

[14] DITTRICH, D., AND KENNEALLY, E. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. *US Department of Homeland Security* (December 2011).

[15] EGELMAN, S., TSAI, J. Y., AND CRANOR, L. F. Tell me lies: A methodology for scientifically rigorous security user studies. In *Workshop on Studying Online Behaviour at CHI'10* (2010), ACM.

[16] ELLIOTT, A., AND BRODY SINCLAIR, S. S. Design Implications of Lived Surveillance in New York. In *Workshop on Everyday Surveillance at CHI'16* (2016).

[17] FELT, A. P., REEDER, R. W., ALMUHIMEDI, H., AND CONSOLVO, S. Experimenting at scale with Google Chrome's SSL warning. In *SIGCHI Conference on Human Factors in Computing Systems* (2014), pp. 2667–2670.

[18] FERREIRA, A., HUYNEN, J.-L., KOENIG, V., AND LENZINI, G. A conceptual framework to study socio-technical security. In *Human Aspects of Information Security, Privacy, and Trust*. Springer, 2014, pp. 318–329.

[19] FLORÊNCIO, D., HERLEY, C., AND VAN OORSCHOT, P. C. Password portfolios and the finite-effort user: Sustainably managing large numbers of accounts. In *USENIX Security* (2014).

[20] FUJITA, M., IKEYA, Y., KANI, J., AND NISHIGAKI, M. Chimera captcha: A proposal of captcha using strangeness in merged objects. In *Human Aspects of Information Security, Privacy, and Trust (HAS 2015), HCI International 2015*. Springer, 2015, pp. 48–58.

[21] GÜRSES, S. *Multilateral Privacy Requirements Analysis in Online Social Network Services*. PhD thesis, K.U. Leuven, 2010.

[22] HATLEBACK, E., AND SPRING, J. M. Exploring a mechanistic approach to experimentation in computing. *Philosophy & Technology 27*, 3 (2014), 441–459.

[23] HENRICH, J., HEINE, S. J., AND NORENZAYAN, A. The weirdest people in the world? *Behavioral and brain sciences 33*, 2-3 (2010), 61–83.

[24] HERLEY, C. So long, and no thanks for the externalities: The rational rejection of security advice by users. In *New Security Paradigms Workshop (NSPW'09)* (2009), ACM, pp. 133–144.

[25] HERLEY, C. More is not the answer. *IEEE Security & Privacy 12*, 1 (2014), 14–19.

[26] HERTWIG, R., AND ORTMANN, A. Experimental practices in economics: A methodological challenge for psychologists? *Behavioral and Brain Sciences 24*, 03 (2001), 383–403.

[27] HORTON, J. J., RAND, D. G., AND ZECKHAUSER, R. J. The online laboratory: Conducting experiments in a real labor market. *Experimental Economics 14*, 3 (2011), 399–425.

[28] JENSEN, C., POTTS, C., AND JENSEN, C. Privacy practices of internet users: Self-reports versus observed behavior. *International Journal of Human-Computer Studies 63*, 1 (2005), 203–227.

[29] JENSEN, D. Credibility. In *The Sage Encyclopedia of Qualitative Research Methods*, L. M. Given, Ed. SAGE Publications, 2008, pp. 139–140.

[30] JENSEN, D. Transferability. In *The Sage Encyclopedia of Qualitative Research Methods*, L. M. Given, Ed. SAGE Publications, 2008, p. 887.

[31] JENTZSCH, N., PREIBUSCH, S., AND HARASSER, A. Study on monetising privacy: An economic model for pricing personal information. *ENISA, Feb* (2012).

[32] KANG, R., BROWN, S., DABBISH, L., AND KIESLER, S. B. Privacy attitudes of mechanical turk workers and the us public. In *SOUPS* (2014), pp. 37–49.

[33] KIRLAPPOS, I., BEAUTEMENT, A., AND SASSE, M. A. "comply or die" is dead: Long live security-aware principal agents. In *Financial Cryptography and Data Security*. Springer, 2013, pp. 70–82.

[34] KIRLAPPOS, I., PARKIN, S., AND SASSE, M. A. Learning from "shadow security": Why understanding non-compliance provides the basis for effective security. In *USEC 2014: NDSS Workshop on Usable Security* (2014).

[35] KLEIN, G. *Streetlights and shadows: Searching for the keys to adaptive decision making*. MIT Press, 2009.

[36] KROL, K., MOROZ, M., AND SASSE, M. A. Don't work. Can't work? Why it's time to rethink security warnings. In *International Conference on Risk and Security of Internet and Systems (CRiSIS'12)* (2012), IEEE, pp. 1–8.

[37] KROL, K., PARKIN, S., AND SASSE, M. A. Better the devil you know: A user study of two CAPTCHAs and a possible replacement technology. In *USEC 2016: NDSS Workshop on Usable Security* (2016).

[38] KROL, K., RAHMAN, M. S., PARKIN, S., DE CRISTOFARO, E., AND VASSERMAN, E. Y. An Exploratory Study of User Perceptions of Payment Methods in the UK and the US. In *USEC 2016: NDSS Workshop on Usable Security* (2016).

[39] KUHN, T. S. *The structure of scientific revolutions*, fourth ed. University of Chicago press, 2012.

[40] LISA M. GIVEN, K. S. Trustworthiness. In *The Sage Encyclopedia of Qualitative Research Methods*, L. M. Given, Ed. SAGE Publications, 2008, pp. 896–897.

[41] MALHEIROS, M., BROSTOFF, S., JENNETT, C., AND SASSE, M. A. Would you sell your mother's data? Personal data disclosure in a simulated credit card application. In *The Economics of Information Security and Privacy*. Springer, 2013, pp. 237–261.

[42] METCALF, L. B., AND SPRING, J. M. Blacklist ecosystem analysis: Spanning Jan 2012 to Jun 2014. In *The 2nd ACM Workshop on Information Sharing and Collaborative Security* (Denver, Oct 2015), pp. 13–22.

[43] MILLER, P. Validity. In *The Sage Encyclopedia of Qualitative Research Methods*, L. M. Given, Ed. SAGE Publications, 2008, pp. 910–911.

[44] PFLEEGER, S. L., AND CAPUTO, D. D. Leveraging behavioral science to mitigate cyber security risk. *computers & security 31*, 4 (2012), 597–611.

[45] POPPER, K. R. *The logic of scientific discovery*. 1959.

[46] PREIBUSCH, S. Economic aspects of privacy negotiations. Master's thesis, Technische Universität Berlin, 2008.

[47] PREIBUSCH, S. How to explore consumers' privacy choices with behavioral economics. In *Privacy in a Digital, Networked World*. Springer, 2015, pp. 313–341.

[48] RADER, E., AND WASH, R. Identifying patterns in informal sources of security information. *Journal of Cybersecurity 1*, 1 (2015), 121–144.

[49] RENAUD, K., VOLKAMER, M., AND RENKEMA-PADMOS, A. Why doesnt jane protect her privacy? In *Privacy Enhancing Technologies* (2014), Springer, pp. 244–262.

[50] RIVERS, W., AND WEBBER, H. The action of caffeine on the capacity for muscular work. *The Journal of physiology 36*, 1 (1907), 33–47.

[51] ROSSOW, C., DIETRICH, C. J., GRIER, C., KREIBICH, C., PAXSON, V., POHLMANN, N., BOS, H., AND VAN STEEN, M. Prudent practices for designing malware experiments: Status quo and outlook. In *Security and Privacy (S&P), IEEE Symposium on* (2012), pp. 65–79.

[52] SASSE, A. Scaring and bullying people into security won't work. *IEEE Security & Privacy*, 3 (2015), 80–83.

[53] SASSE, M. A., BROSTOFF, S., AND WEIRICH, D. Transforming the 'weakest link' – a human/computer interaction approach to usable and effective security. *BT Technology Journal 19*, 3 (2001), 122–131.

[54] SCHECHTER, S. E., DHAMIJA, R., OZMENT, A., AND FISCHER, I. The emperor's new security indicators. In *Security and Privacy, 2007. SP'07. IEEE Symposium on* (2007), IEEE, pp. 51–65.

[55] SCHNEIER, B. Secrets and lies: Security in a digital world, 2000.

[56] SCHNORF, S., SEDLEY, A., ORTLIEB, M., AND WOODRUFF, A. A comparison of six sample providers regarding online privacy benchmarks. In *SOUPS Workshop on Privacy Personas and Segmentation* (2014).

[57] SHENG, S., BRODERICK, L., KORANDA, C. A., AND HYLAND, J. J. Why Johnny still can't encrypt: Evaluating the usability of email encryption software. In *Symposium On Usable Privacy and Security* (2006).

[58] SHIREY, R. Internet Security Glossary, Version 2. RFC 4949 (Informational), Aug. 2007.

[59] SIMON, H. A. Rational choice and the structure of the environment. *Psychological Review; Psychological Review 63*, 2 (1956), 129.

[60] SPIEKERMANN, S., GROSSKLAGS, J., AND BERENDT, B. E-privacy in 2nd generation e-commerce: privacy preferences versus actual behavior. In *Proceedings of the 3rd ACM conference on Electronic Commerce* (2001), ACM, pp. 38–47.

[61] SPRING, J. M. Toward realistic modeling criteria of games in internet security. *Journal of Cyber Security & Information Systems 2*, 2 (2014), 2–11.

[62] SPRING, J. M., METCALF, L. B., AND STONER, E. Correlating domain registrations and dns first activity in general and for malware. In *Securing and Trusting Internet Names: SATIN* (Teddington, UK, March 2011).

[63] STEVES, M., CHISNELL, D., SASSE, M. A., KROL, K., THEOFANOS, M., AND WALD, H. Report: Authentication Diary Study. National Institute of Standards and Technology (NISTIR) 7983, 2014.

[64] WHITTEN, A., AND TYGER, J. D. Why Johnny Can't Encrypt: A Usability Case Study of PGP 5.0. In *USENIX Security* (1999), pp. 169–184.

[65] YEE, K.-P. Aligning security and usability. *IEEE Security & Privacy*, 5 (2004), 48–55.

[66] ZINS, C. Conceptual approaches for defining data, information, and knowledge. *Journal of the American society for information science and technology 58*, 4 (2007), 479–493.

[67] ZURKO, M. E. User-centered security: Stepping up to the grand challenge. In *Computer Security Applications Conference, 21st Annual* (2005), IEEE, pp. 14–pp.

Results and Lessons Learned from a User Study of Display Effectiveness with Experienced Cyber Security Network Analysts

Christopher J. Garneau, Robert F. Erbacher, Renée E. Etoty, and Steve E. Hutchinson
U.S. Army Research Laboratory

Abstract

Background. Visualization tools have been developed for various network analysis tasks for Computer Network Defense (CND) analysts, yet there are few empirical studies in the domain of cyber security that validate the efficacy of various graphical constructions with respect to enhancing analysts' situation awareness.

Aim. The aim of this study is to empirically evaluate the utility of graphical tools for enhancing analysts' situation awareness of network alert data compared with traditional tabular/textual tools. This paper focuses on results of the study and lessons learned for future similar studies.

Method. A tabular display was presented along with two alternative graphical displays in a web-based environment to 24 experienced network analysts. Participants were asked to use the displays sequentially to identify intrusion attempts as quickly and accurately as possible. Data were fabricated by an experienced analyst and do not rely on alert data from a real network.

Results. Analysts performed well on the tabular (baseline) display and also preferred this display to others. However, they were slightly faster and similarly accurate using one of the graphical alternatives (node-link). Subjective feedback shows that many analysts are receptive to new tools while some are skeptical.

Conclusions. Graphical analysis tools have the capability of enhancing situation awareness by preprocessing and graphically arranging data for analysis. Real-world analysts bring a wealth of experience and insight to this sort of research, and the large number of expert responses included in this study is unique. Tempering analyst expectations for the study by clearly explaining the study environment and tasks to be completed would likely lead to more accurate results.

1. Introduction

Suspicious computer network activity identified by an Intrusion Detection System (IDS) requires Computer Network Defense (CND) analysts to make quick, accurate decisions about activity that warrants further investigation and possible remediation. In this initial triage phase of intrusion detection, details on any potential attacks are less important than overall situation awareness of suspicious activity as identified by the IDS configuration. Constant monitoring of textual log files is a difficult task for humans, even for analysts who are trained to quickly recognize abnormal patterns in the data. There exists an opportunity to develop and implement visualizations that preprocess and graphically arrange data to aid in the cyber security analysts' search activities, however graphical techniques have not seen wide implementation in analysts' operations [7]. This paper discusses a user study that investigated three interfaces for representing network alert data to gain insight on features and visual attributes that would be most effective for enhancing analyst situation awareness. The focus of this paper is on the design of the study and how subjective feedback from the study may inform and improve the design of future studies.

1.1. Background

Visualization tools have been developed for various network analysis tasks for CND analysts, including identifying salient features in datasets, tracking analyses, reusing effective workflows, testing hypotheses, and so on. However, in general, there are few available studies in the domain of cyber security that validate the efficacy of various constructions with respect to enhancing analysts' comprehension of alert data. Some studies have investigated analyst needs and have employed cognitive task analysis (CTA) [4, 6]. Requirements and characteristics of next-generation visualizations have resulted from these efforts. More research to better understand analyst needs and validate visual tools will benefit the state of the art in cyber security network analysis and the available tools used to support such analyses.

1.2. Goals of the Study

The overarching goal of this study is to evaluate the utility of graphical tools for enhancing analysts' situation awareness, compared with traditional tabular/textual tools. Specifically, questions posed by the study—relevant to the discussion in this paper—include:

- Do graphical displays enhance performance?

- What barriers limit the adoption of graphical displays?
- What types of graphical displays would be most effective?

In this paper, only results from experienced network analysts are considered, even though the study also included novices (university students) in the pool of participants.

2. Related Work

Evaluating scientific visualization techniques is a longstanding challenge [1, 2, 10]. Similarly, the field of information visualization has a strong tradition in pioneering research in evaluation techniques [3, 14, 17]. User studies often rely on timing and accuracy information collected during the study coupled with subjective user surveys given after the experiment is completed. This combination of empirical measurement with subjective questionnaire is designed to assess the efficacy of a visualization technique with respect to related methods. However, the analysis of user evaluation studies remains difficult. These challenges are often compounded by the limited empirical data acquired during the study. Beyond the specific details of the many user study experiments, they all share a common goal: to assess the strengths and weaknesses inherent to a visualization technique or system. Incorporating as many objective measures as possible into the experiment not only provides a more robust analysis, but also mitigates subjectivity often introduced by users' preferences, biases, and retrospection.

Due to the nature of today's complex scientific data, simply displaying all available information does not adequately meet the demands of domain scientists. A wide variety of visualizations for cyber security analysts have been proposed [16]. Determining the best use of visualization techniques is one of the goals of scientific visualization evaluations. The types of improvements offered by the method being studied dictate evaluation methods. Some evaluations are concerned primarily with technological improvements such as rendering speed or the management of large data. User studies have been used to evaluate everything from aircraft cockpits [15] and surgical environments [13] to visualization methods [11]. Evaluating visualization methods that focus on human factors often employ user studies or expert evaluations to determine their effects on interpretation and usability. An expert assessment takes advantage of knowledgeable users to enable more poignant analysis of use cases and these experts also bring with them their own preconceptions and preferences that can skew studies. Traditional evaluation methods provide mechanisms to gauge aspects of visualizations or environment. Unfortunately, experiments using surveys to measure user experience introduce subjectivity and bias from the users. Subjectivity in user responses may be partially mitigated using questionnaires developed with the Likert Scale [12]. Subjectivity in evaluation may provide important insights into how users interact with the systems being studied. However, subjective measures do not help answer questions regarding how effective a method is at eliciting insight from a dataset. This is a primary purpose of visualization. Our goal and purpose is to use this project as an empirical study to examine the cognitive aspects of visual displays with the goal of identifying components and representations that most effectively aid the computer network analyst in interpreting the underlying activity in a network sample. Results from the study are helpful to understand the potential and limitations of the suggested visual displays attempting to aid analysts' needs to better achieve their tasks.

3. Study Overview and Method

In the study, participants acted as analysts and their job was to identify as many network threats as possible within a limited set of IDS alert data. Because the goal of the study was to examine how situation awareness may be enhanced in the initial triage phase, no additional investigation of alerts was required or permitted and analysts were expected to discriminate between potentially malicious and benign alerts based strictly upon the data presented in the displays. Objective response variables include: (1) true-positive rate of identification of intrusion attempts for each type of display, (2) false-positive rate of identification of intrusion attempts for each type of display, and (3) time required for identification for each type of display. Subjective feedback was also collected and is the focus of the "lessons learned" presented in this paper.

3.1. Display Design

Three types of displays were chosen for inclusion in the study:

1. **Tabular display** (baseline): Basic functionality is similar to Microsoft Excel. Participants could sort and filter data by any parameter, and individual rows were selected for submission via checkboxes. See Fig. 1.

	ID	Time	Src Entity	Src Port	Dst Entity	Dst Port	Dst Country	Alert
☐	1	7/5/13 5:37	USA.3	52869	USA.12	445	USA	Fragmented IP Packet
☐	2	7/5/13 5:11	USA.113	2787	land of OZ.159	80	land of OZ	WEB-MISC Netscape Enterprise Server directory view
☐	3	7/5/13 5:30	USA.12	3377	Pellucidar.28	443	Pellucidar	Fragmented IP Packet
☐	4	7/5/13 5:15	USA.110	2986	land of OZ.99	80	land of OZ	WEB-MISC Netscape Enterprise Server directory view
☐	5	7/5/13 5:35	USA.12	3498	Neverland.49	443	Neverland	Fragmented IP Packet
☐	6	7/5/13 5:09	USA.3	2660	Dreamlands.106	80	Dreamlands	WEB-CGI php.cgi access
☐	7	7/5/13 5:38	USA.76	53249	Hogwarts.88	8080	Hogwarts	ET TROJAN Qhosts Trojan Check-in
☐	8	7/5/13 5:21	USA.12	3193	Wonderland.8	80	Wonderland	WEB-CGI php.cgi access
☐	9	7/5/13 5:11	USA.113	2800	Gullivers World.198	443	Gullivers World	Fragmented IP Packet
☐	10	7/5/13 5:12	USA.12	2852	USA.4	21	USA	FTP Satan Scan
☐	11	7/5/13 5:30	USA.12	3386	Dreamlands.106	80	Dreamlands	WEB-CGI php.cgi access
☐	12	7/5/13 5:07	USA.3	52593	Blefuscu.112	443	Blefuscu	Fragmented IP Packet
☐	13	7/5/13 5:04	USA.12	2587	Utopia.211	80	Utopia	Javascript Exploit CVE-2012-09-10a
☐	14	7/5/13 5:38	USA.12	52870	Pern.152	21	Pern	FTP STOR overflow attempt
☐	15	7/5/13 5:11	USA.113	2768	Neverland.49	443	Neverland	Fragmented IP Packet
☐	16	7/5/13 5:12	USA.12	2859	USA.1	21	USA	FTP Satan Scan
☐	17	7/5/13 5:11	USA.113	2737	Middle-Earth.64	80	Middle-Earth	INFO Connection Closed MSG from Port 80
☐	18	7/5/13 5:30	USA.12	3383	Pern.110	443	Pern	Fragmented IP Packet
☐	19	7/5/13 5:25	USA.96	65113	Atlantis.10	8000	Atlantis	ET TROJAN Sality Variant Downloader Activity
☐	20	7/5/13 5:11	USA.113	2731	Deltora.36	80	Deltora	INFO Connection Closed MSG from Port 80
☐	21	7/5/13 5:21	USA.12	20	USA.3	52644	USA	FTP - Suspicious MGET Command
☐	22	7/5/13 5:30	USA.12	3394	Tatooine.157	443	Tatooine	Fragmented IP Packet
☐	23	7/5/13 5:11	USA.113	2758	Pandora.116	80	Pandora	WEB-CGI webspeed access
☐	24	7/5/13 5:14	USA.12	2948	USA.11	21	USA	FTP Satan Scan
☐	25	7/5/13 5:12	USA.12	2853	USA.5	21	USA	FTP Satan Scan

Fig. 1 Tabular display, showing alerts with ID 1-24 (no alerts selected).

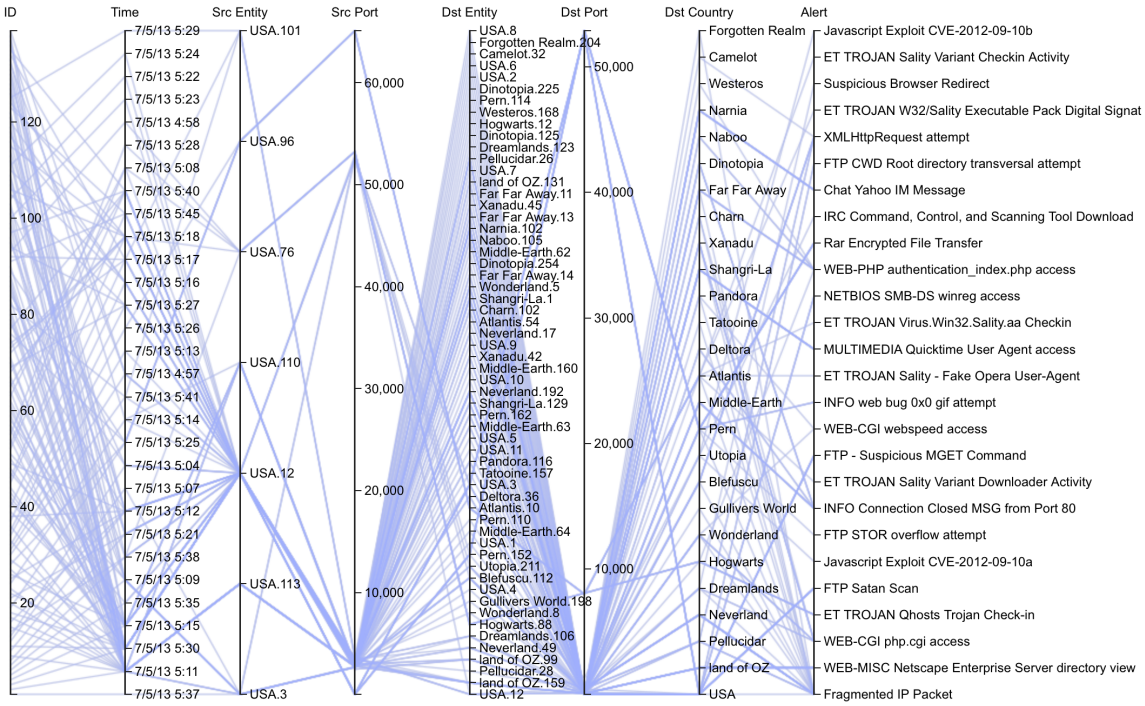


Fig. 2 Parallel coordinates display shown to participants (no alerts selected).

2. **Parallel coordinates display** (graphical alternative 1): This display is one of the most published multivariate visualization techniques [e.g., 8, 9]. Participants could highlight/filter a range of values on any of the parameters; to further refine the selection for submission, ranges on additional axes could be selected. See Fig. 2.

3. **Node-link display** (graphical alternative 2): This display has been tailored to the task of intrusion detection based on related visualization research [5]. As a participant moused over a

marker (red dot), a popup appeared showing details of the alert associated with the source-destination node pair. Participants could click any number of markers for submission. See Fig. 3. This display was selected by an ex-analyst who had switched to the R&D side of the house and had experience reviewing many different display formats.

The source data presented in each display were identical and were synthesized by an expert from an hour's worth of alert messages. For security reasons, it was not possible to use real data captured from an operational environment and so data were fabricated for this study. The data uses an attack scenario where many external nodes are attacking a smaller number of friendly peer nodes. The alert data contain three types of intrusion attempts of varying difficulty: (1) a three-stage intrusion that consisted of a web infection, scanning, and data exfiltration (32 alerts, "easy" difficulty), (2) periodic Trojan scanning (5 alerts, "moderate" difficulty), and (3) Sality Trojan infection (5 alerts, "hard" difficulty). 42 alerts of a total 139 alerts belonged to one of the three intrusion attempts. Eight parameters were associated with each alert message in each of the displays: (1) alert ID, (2) date/time stamp, (3) source entity/IP, (4) source port, (5) destination entity/IP, (6) destination port, (7) destination country, and (8) alert message (see Fig. 1 for a tabular representation of a subset of the data).

3.2. Study Design

The study collected background and demographic data, utilized pre- and post-task questionnaires, and also obtained objective and subjective feedback. The study was administered as a within-subjects design (i.e., every single participant subjected to every single treatment). Each participant completed the task independently while sitting at a computer workstation using the three displays sequentially, and the display presentation order was varied to minimize the effects of practice and order bias. In other words, each participant completed the task using their first assigned display (with a time limit of 20 minutes), then their second assigned display, then

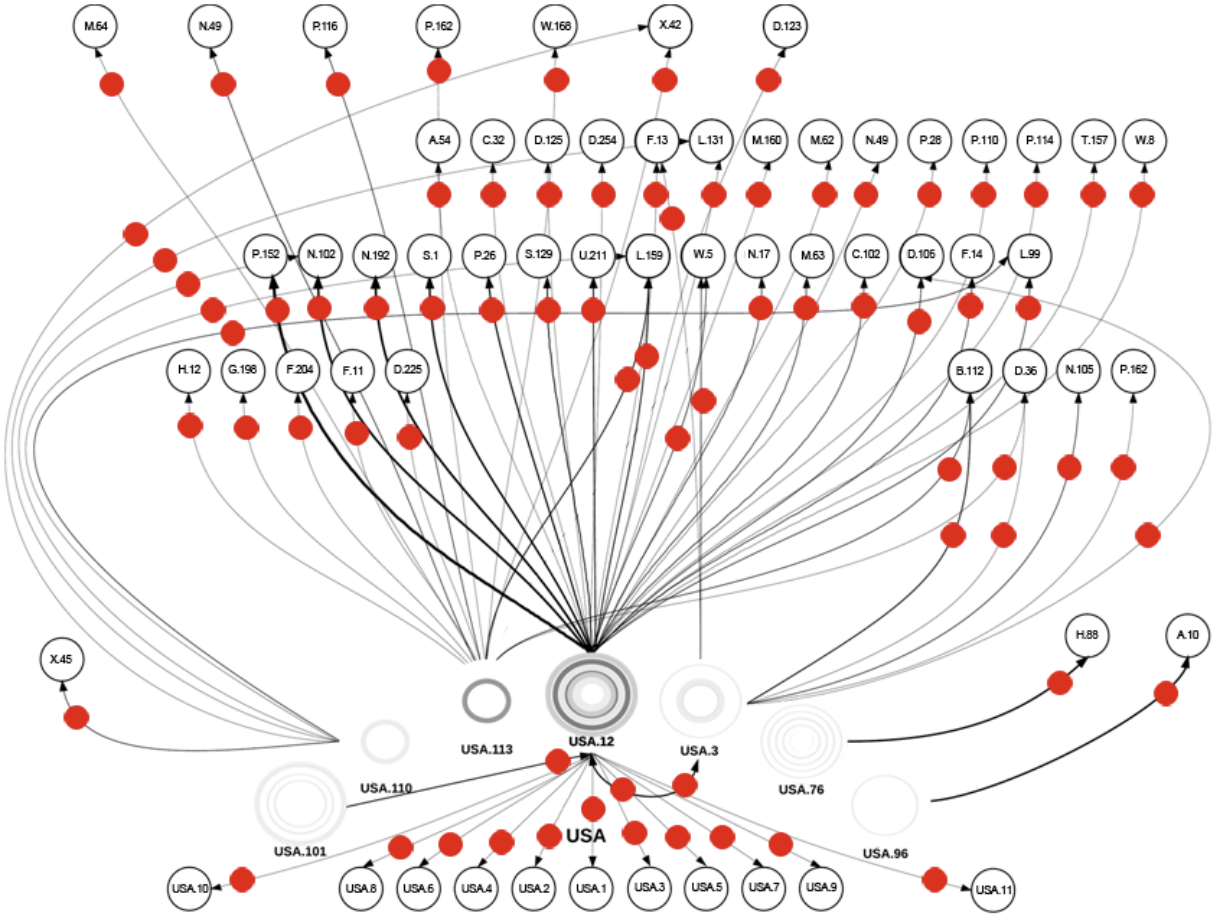


Fig. 3 Node-link display (no alerts selected).

their third assigned display. Since there are three unique displays there were six permutations of the ordering. The order of the displays was assigned such that a similar number of participants were assigned to each display order (i.e., roughly equal number of participants were assigned to the following orderings: TNP, TPN, NPT, NTP, PTN, PNT, where T=Tabular, P=Parallel Coordinates, and N=Node-link). The computer workstation consisted of a typical laptop computer, external monitor, keyboard, and mouse with a typical desk and office chair. The study was conducted in a web-based environment; survey questions were administered with the survey application LimeSurvey[1] and the main study task with the three displays was administered via custom HTML and JavaScript code. The study was approved by an appropriately constituted institutional review board (IRB) at ARL.

3.3. Procedure

The procedure used for the study is described in this section.

Step 1. Study began with a welcome followed by an introduction of the investigators.

Step 2. Investigators then briefed the participants on the study and obtain informed consent. Participants of this study were given a random identification number.

Step 3. The investigators explained each visual display and the techniques for representing a network system's attributes.

Step 4. The investigators conducted a demo of the participants' tasks in the web-based environment. This served as practice for the participants.

Step 5. Investigators provided time to entertain participants' questions concerning their tasks or any other aspects of the study.

Step 6. Participants completed background, demographic, and pre-task questionnaires.

Step 7. Participants completed main task of the study with the three visual representations as described in Section 3.2 (i.e., they were presented with the three displays according to the assigned ordering and were asked to identify as many intrusion attempts as possible in each).

Step 8. Participants completed their post-task questionnaires and provided the investigators with any final remarks or comments.

Step 9. Investigators lead a debrief session and provided the participants with a copy of the signed consent form.

Participants were given a maximum of three hours to complete the tasks described above as well as tasks for a similar related study. Most participants completed the tasks for this study only—including pre- and post-task questionnaires—in about 1.5 hours. Participants completed the tasks independently either alone with the investigator in the room or with the investigator plus one other participant in the room (but working separately at opposite sides of the room).

4. Results

Results of the study are presented next, including demographics of the participants, objective performance of participants on the analysis task, and a subset of comments provided by participants.

4.1. Participant Characteristics and Demographics

The participant population consisted of 24 experienced analysts from ARL who actively or previously had conducted CND analyses and were employed by ARL at the time of the study. These analysts were selected for inclusion in the study due to their unique skillset and availability. A majority of the analysts' full-time job is monitoring sensors for malicious activity—initially through generated alerts and subsequently through raw data logs files. While "expert" is a vague and potentially misleading label, the participants in this study are considered to be experts specifically at analyzing network data for malicious activity. All were assumed to be familiar with tabular tools and may or may not have been familiar with graphical tools. All were eighteen years of age or older, had 20/20 vision (or corrected 20/20 vision), all passed a test for colorblindness, and none reported having any other disabilities. Note that demographic questions did not force a response so some participants did not answer some questions. Table 1 summarizes participant demographics.

Table 1 Demographics for participants. Some participants did not answer one or more of these questions.

	Number of Analysts
Gender	
Female	0
Male	22
Race	
White	13
Black or African American	5

[1] https://www.limesurvey.org

Asian	1
Other	3
Age	
18-25 years	3
26-35 years	11
36-45 years	8
46-55 years	2
Highest level of education completed	
Some college but did not finish	5
Two-year college degree/A.A./A.S.	7
Four-year college degree/B.A./B.S.	7
Some graduate work	3
Completed Masters or professional degree	2
Experience as cyber analyst	
Less than 1 year	2
1-3 years	7
3-5 years	11
5-10 years	2
Greater than 10 years	1

Table 2 Objective performance parameters for each of the three displays. TP and FP give the average number of true positives and false positives identified, respectively. n indicates the number of responses considered for the given display—this metric varies because not all analysts completed the task using all displays.

	Tabular	Parallel Coordinates	Node-Link
n	23	21	23
TP	24.8	20.3	25.9
FP	17.1	30.4	15.7
Completion Time (min)	15.6	11.9	12.2
Accuracy	0.752	0.624	0.771
Precision	0.670	0.501	0.703
Recall	0.590	0.482	0.617

Table 3 Differences between means of various objective performance parameters are indicated for tabular vs. parallel coordinates (PC) and tabular vs. node-link. Positive values indicate higher values for the first of each pair. Significance is also indicated (* Significant at the 0.05 probability level, ** Significant at the 0.01 probability level, * Significant at the 0.001 probability level).**

	Tabular vs. PC	Tabular vs. Node-link	Node-link vs. PC
TP	+4.50	-1.09	+5.58
FP	-13.34	+1.35	-14.7*
Time (min)	+3.76*	+3.44*	+0.32
Accuracy	+0.13*	-0.019	+0.15**
Precision	+0.17	-0.033	+0.20*
Recall	+0.11	-0.028	+0.14

4.2. Objective Performance

Objective performance is measured in terms of: (1) true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) that analysts identified in the dataset; (2) measures derived from total TP, FP, TN, and FN (such as accuracy); and (3) time/duration to complete tasks. The derived measures are defined in terms of total TP, FP, TN, and FN as follows:

$$\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

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

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

Tables 2 and 3 provide summaries of these metrics across the displays. Fig. 4 plots the number of true positives identified against the number of false positives identified for each participant using each of the three

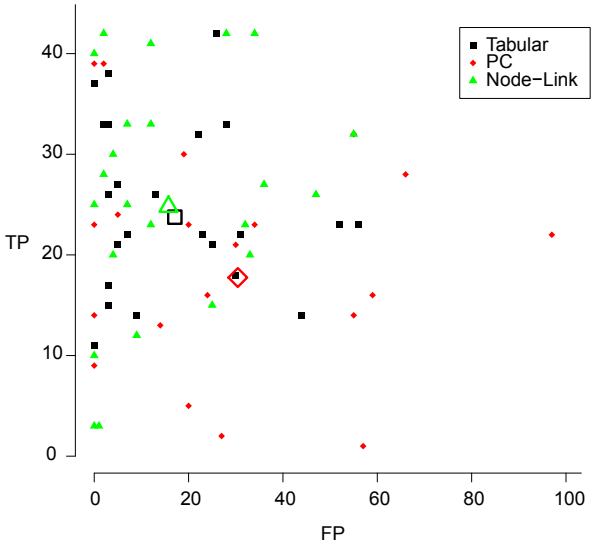


Fig. 4 Number of true positives (TP) vs. false positives (FP) for each participant using each display. Averages for each of the three displays are shown using the larger, open (unfilled) shapes.

displays.

4.3. Subjective Feedback/Comments

Subjective feedback was collected via questionnaires. This section presents a subset of participant comments to the indicated free response prompts that are representative of most responses (duplicate or very short responses are excluded, instead focusing on those providing a unique point of view). All responses could not be included here due to limited space.

"What components of the visual displays (table, parallel coordinates, and node link display) were most effective?"

- "Table is much better at identifying the actual alerts. Parallel is more involved with showing all the alerts that are on the node. Node Link is best at showing all the alerts that are attached to each ID"
- "The table was the most effective. It is easy, you don't need all these fancy visual tools to find issues. Make everything simpler. The Node display and the Parallel display looked like a lot of noise. I wasn't interested in using them."

"What aspects of the visualizations (table, parallel coordinates, and node link display) did you like best?"

- "I thought the node link display was interesting. It was nice to see a view of a network topology and the path the data travels."
- "Visually separating the internal hosts from external hosts to quickly see flow of data between internal only and internal to external."
- "Table is much better for seeing and lumping the alerts together. Parallel showed the flow a lot better. Node link showed the flow and other ID numbers to the alerts faster."
- "I liked being able to quickly identify the most active times of the day and the most common request domains of the parallel coordinates displays."

"What aspects of the visualizations (table, parallel coordinates, and node link display) did you not like?"

- "It takes a while to glance at everything when you can glance at percentages and numbers. Those are quicker to grasp sometimes."
- "The line display and node display were convoluted at best; they detracted from the information presented. In terms of investigative procedure, I believe they would only serve to stifle."
- "The parallel coordinates was a nightmare to visualize and the node link display was far too time consuming to hover through."
- "Connections were very hard to follow, information was displayed in a non-intuitive manner, correlations were very difficult to find without excessive work."
- "The parallel coordinates interface was not useful or intuitive and I could not sort the coordinates. Moreover, once I was finished with an alert it should have been removed from my view. Also I should be able to remove noise from my view with a filter. The node link display was slightly more useful but the node sizes were not intuitive nor were the most important piece of analysis data displayed up front: the alerts!"
- "Parallel coordinates is good for fine analysis but not for raw / bulk analysis"
- "The graphical representations were completely unusable to me. The table was fine but there needs to be drill down options to see more data. I look at an alert then I check some traffic if I see something suspicious I dump the traffic and do a thorough investigation."

"What did you learn from this study?"

- "That there are some great relationship tools for network intrusion, and some not so great ones."
- "Being able to see correlations is very important (time, src ip, dst ip, etc). All of this information is very difficult to fit into a graphical interface. Without being able to sort and filter, this makes the analysts job far more difficult. The graphical displays seem decent for a "birds eye view", but a nightmare for actual everyday analysis."
- "Aggregated data obfuscates signal!"
- "A simple table can be the better option sometimes."
- "Graphics helps make analysis easier"
- "With all due respect regarding this project, it seems like there is a long way to go before a very useful graph or node visualization would be effective or efficient to use."
- "I do not like graphical displays while doing analysis. I find them highly unnecessary, unless of course I was presented with ones that are more intuitive."

5. Discussion of Results and Lessons Learned

The results from the study presented in this paper provide insight into two areas of inquiry: (1) what types of displays and visual attributes of those displays are most effective for enhancing analyst performance?, and (2) what aspects of the study implementation worked well

and what aspects need improvement for future similar studies, based on analyst feedback?

5.1. Discussion of Performance

Analysts performed well on the tabular (baseline) display, and also preferred this display to others. Based on their familiarity with this representation—and its use in their existing workflow—this was to be expected. Compared with performance on the parallel coordinates display, analysts were slower but more accurate using the tabular display. Compared with the node-link display, analysts were slower and about as accurate using the tabular display. It is notable that analysts had a high rate of false positive identification for the parallel coordinates display (see Table 2), despite the display's advantages for representing many-dimensional data. This would translate into more time wasted in a real-world scenario investigating benign alerts.

Subjective feedback confirms the objective performance results. Some analysts could see the value in the parallel coordinates display, but most felt strongly against it ("parallel coordinates is a nightmare to visualize", "the parallel coordinates interface was not useful or intuitive"). In general, the baseline tabular display was preferred to graphical alternatives ("the graphical representations were completely unusable to me"). However, some analysts could see the value in the graphical displays ("it was nice to see a view of a network topology and the path the data travels", "node link showed the flow and other ID numbers to the alerts faster").

5.2. Lessons Learned

It is instructive to consider analyst performance and feedback so future studies can improve upon the current work.

To avoid inaccuracies in study results, it is important to take participants' expectations into account in the study methodology. While specific instructions were provided to the analysts, some wanted more data and could not understand how to use the displays for the task provided. One said, "Table data was fine but I need more than play data to do proper analysis. It isn't just the alert or a darker line of traffic that determines infection or compromise." It seems as though the intent of the study was not well communicated to this participant. The displays were not intended to replicate a complete analysis session, but rather provide a tool for rapidly identifying indicators of compromise for further investigation and enhance situation awareness, i.e., we were only investigating initial triage, initial indicators at this stage. Future studies will need to be performed for detailed analysis of these indicators for full analysis. Moreover, one way to enhance participants' experience and also gain further insight would be to ask participants what their next analysis steps would be, even though these are not considered in the experiment; this might give analysts a sense that they have completed the analysis.

Designing a complete simulated environment would be ideal for the most accurate results. Such a study environment should include features such as: (1) data of real-world appearance and scale, (2) the ability to contact other teams, such as a threat center or a remote target site through a fully scripted conversation, (3) ability to make simulated communications with the forensic analysis team, and so on. However, such a study would require years of research and design and may not provide results that justify such an undertaking, and numerous pilot studies (similar to the current work) would have to precede such a detailed study. Leveraging expectations by noting to analysts that they were participating in a scaled-down study should have been emphasized in the pre-study briefing.

It likely would have been beneficial to emphasize to analysts possible future improvements to their workflow and explicitly ask them to consider components of the display that they found useful or interesting (rather than dismissing a certain display altogether as a "nightmare"). While experienced analysts bring a wealth of knowledge and insight to research of this nature, they have a certain way they approach their work and may be critical of alternatives. Some were receptive to new tools ("It's good that I got a chance to see what type of tools can be deployed in future and felt very good to leave feedback about these tools"). However, others had more cynical viewpoints ("The table was the most effective. It is easy, you don't need all these fancy visual tools to find issues. Make everything simpler. The Node display and the Parallel display looked like a lot of noise. I wasn't interested in using them"). It is undesirable to eliminate contrary perspectives, but approaching the study with a "help me to help you" attitude may enhance results. While analysts possess extensive knowledge in the cyber security domain, they likely do not have much knowledge of the principles for effective display design and may instinctively react negatively to a display that is unfamiliar. Negative feedback should be considered seriously but ought not discourage future innovation in tools and displays for the field.

Other modifications to the implementation of the study would likely improve results and participants' perception of the displays. Including as much interactivity as possible in the displays to be evaluated would benefit the participant experience and thus enhance credibility

of the results. While the web environment in this study used lightly customized JavaScript libraries (e.g., D3.js[2]) and permitted some interactivity, more extensive interactivity likely would have mitigated analyst complaints about limitations of the displays ("The parallel coordinates interface was not useful or intuitive and I could not sort the coordinates. Moreover, once I was finished with an alert it should have been removed from my view.", "Connections were very hard to follow, information was displayed in a non-intuitive manner, correlations were very difficult to find without excessive work").

The study was invasive for participants, requiring them to leave their regular work site and adjust to an unfamiliar laboratory setting. Future studies should attempt to make the study as non-invasive as possible. Ideally, experimentation would occur in the regular work environment, but if this is not possible (e.g., due to security restrictions), future studies should attempt to replicate environmental conditions of the analysts' environment (lighting, temperature, computer hardware, etc.) as closely as possible. These adjustments would make it more likely that a participant would behave as they normally do, which should be a goal of any future studies. However, such modifications should be weighed against the benefit of further instrumentation and data collection capability; for instance, eye tracking would provide insight into the elements of the displays to which participants were paying attention.

There were also other limitations in the study implementation that should be noted. Using the same alert data across the three displays might introduce confounding effects; i.e., participant exposure to the underlying dataset in the first presented display might shape interactions in subsequent displays. While randomizing presentation order for the displays somewhat lessens this effect, generating distinct yet similar data sets for each display might be preferable. In an attempt to enhance participants' incentive to perform well (and lend a game-like quality to the test environment) an accuracy indicator was added to each of the displays. However, there are several drawbacks to its inclusion: it would not exist in a real world scenario, it may have influenced perception of the tools, and it may have altered performance in unexpected ways. Similarly, a 20-minute time limit (enforced by a countdown timer visible to the participant) perhaps added a certain sense of realism and time pressure to the task but the time limit was chosen arbitrarily and benefits and effects on results are unclear. There also likely differences in policies based on site, and interpreting and understanding such differences is an important analyst responsibility. Future studies should address this component.

[2] http://d3js.org

5.3. Future Work

Future studies might consider several changes and enhancements to the study implementation discussed in this paper. First, modifying the experiment design by asking participants to self-rate confidence in their answers for comparison with actual accuracy scores might yield insight about the user experience of the interface. To better align with analyst expectations, future studies might better contextualize the visualizations within other tasks that analysts perform (analyzing alerts is only a part of discriminating malicious network activity from benign activity). Thoroughly understanding analyst workflow and the current tools used by the analyst participants would be essential.

Future studies might also further investigate integrating elements of traditional/tabular and graphical displays. While the displays selected for inclusion in this study were intended to be representative of different types of multivariate displays indicated for the analyst tasks under consideration, they have not been optimized for usability (e.g., placement and size of elements, controls, and so on) and fully implemented with the necessary features for detailed analysis. Future studies might investigate more complete and perhaps alternate types of displays for representing alert information (incorporating interactivity as discussed previously). Finally, future work might investigate the use of different populations of participants. While unreported here, this study also gathered input from "novice" users (university students); future work might investigate including novices that possess domain knowledge but have little or no operational experience (i.e., a new hire) to assess how training varies among the different kinds of displays.

6. Conclusion

This study revealed that analysts are most comfortable using analysis tools with which they are already familiar (i.e., tabular/textual tools), yet are able to achieve similar accuracy in less time for an alert scanning task using some graphical alternatives (node-link). Such graphical displays have the capability of enhancing situation awareness by preprocessing and graphically arranging data for analysis. Real-world analysts bring a wealth of experience and insight to the research, but tempering analyst expectations for the study by clearly explaining the study environment and tasks to be completed will likely lead to more accurate results. Similar future studies validating proposed alternative graphical tools should also try to make the interfaces as interac-

tive as possible and should be constructed with a keen knowledge of existing analyst tools and workflow.

7. References

[1] Acevedo, D. and Laidlaw, D. "Subjective quantification of perceptual interactions among some 2D scientific visualization methods." *IEEE Transactions on Visualization and Computer Graphics* 12.5 (2006): 1133-1140.

[2] Acevedo, D., Jacson, C., Drury, F. and Laidlaw, D. "Using visual design experts in critique-based evaluation of 2D vector visualization methods." *IEEE Transactions on Visualization and Computer Graphics* 14.4 (2008): 877-884.

[3] Carpendale, S. "Evaluating information visualizations." *Information Visualization*. Ed. Kerren, A., Stasko, J. T., Fekete, J., and North, C. Berlin: Springer, 2008. 19-45. Print.

[4] D'Amico, A., Whitley, K., Tesone, D., O'Brien, B., and Roth, E. "Achieving cyber defense situational awareness: A cognitive task analysis of information assurance analysts." *Proceedings of the 49th Human Factors and Ergonomics Society Annual Meeting, Orlando, FL, 26-30 September 2005*. Santa Monica, CA: Human Factors and Ergonomics Society, 2005. 229-233.

[5] Erbacher, R. F., Walker, K. L., and Frincke, D. A. Intrusion and misuse detection in large-scale systems. *IEEE Computer Graphics and Applications* 22.1 (2002): 38-47.

[6] Erbacher, R. F., Frincke, D. A., Moody, S. J., Fink, G. "A Multi-Phase Network Situational Awareness Cognitive Task Analysis." *Information Visualization* 9.3 (2010): 204-219.

[7] Etoty, R. E. and Erbacher, R. F. *A Survey of Visualization Tools Assessed for Anomoly-Based Intrusion Detection Analysis*. Adelphi, MD: U.S. Army Research Laboratory, 2014. Print. ARL-TR-6891.

[8] Giacobe, N. and Xu, S. "Geovisual analytics for cyber security: Adopting the Geoviz toolkit." *Proceedings from the 6th Visual Analytics Science and Technology (VAST) IEEE Conference, Providence, RI, 23-28 October 2011*. Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2011. 315-316.

[9] Goodall, J. R., and Sowul, M. "VIAssist: Visual analytics for cyber defense." *Proceedings of the 2009 IEEE Conference on Technologies for Homeland Security (HST), Waltham, MA, 11-12 May 2009*. Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2009. 143-150.

[10] Kosara, R., Healey, C. G., Interrante, V., Laidlaw, D. H., and Ware, C. "Thoughts on user studies: Why, how and when." *IEEE Computer Graphics and Applications* 23.4 (2003): 20-25.

[11] Laidlaw, D. H., Kirby, R. M., Jackson, C. D., Davidson, J.S., Miller, T. S., Da Silva, M., Warren, W. H., and Tarr, M. J. "Comparing 2D vector field visualization methods: A user study." *IEEE Transactions on Visualization and Computer Graphics* 11.1 (2005): 59-70.

[12] Likert, R. "A technique for the measurement of attitudes." *Archives of Psychology* 22.140 (1932): 1-55.

[13] Reitinger, B., Bornik, A., Beichel, R., and Schmalstieg, D. "Liver surgery planning using virtual reality." *IEEE Computer Graphics and Applications* 26.6 (2006): 36-47.

[14] Riche, N. "Beyond system logging: Human logging for evaluating information visualization." *Proceedings of BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV) 2010, a workshop of the ACM Conference on Human Factors in Computing Systems, Atlanta, GA, 10-11 April 2010*. New York City, NY: Association for Computing Machinery Special Interest Group on Computer-Human Interaction, 2010.

[15] Sarter, N. B. and Woods, D. D. "Pilot interaction with cockpit automation II: An experimental study of pilots' model and awareness of the flight management system." *The International Journal of Aviation Psychology* 4.1 (1994): 1-28.

[16] Shiravi, H., Shiravi, A. and Ghorbani, A. "A survey of visualization systems for network security." *IEEE Transactions on Visualization and Computer Graphics* 18.8 (2012): 1313–1329.

[17] Shneiderman, B. and Plaisant, C. "Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies." *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI) 2006, Venezia, Italy, May 23–26, 2006*. New York City, NY: Association for Computing Machinery Special Interest Group on Computer-Human Interaction, 2006.

Combining Qualitative Coding and Sentiment Analysis: Deconstructing Perceptions of Usable Security in Organisations

Ingolf Becker, Simon Parkin and M. Angela Sasse
University College London
{*i.becker, s.parkin, a.sasse*}*@cs.ucl.ac.uk*

Abstract

Background: A person's security behavior is driven by underlying mental constructs, perceptions and beliefs. Examination of security behavior is often based on dialogue with users of security, which is analysed in textual form by qualitative research methods such as Qualitative Coding (QC). Yet QC has drawbacks: security issues are often time-sensitive but QC is extremely time-consuming. QC is often carried out by a single researcher raising questions about the validity and repeatability of the results. Previous research has identified frequent tensions between security and other tasks, which can evoke emotional responses. Sentiment Analysis (SA) is simpler to execute and has been shown to deliver accurate and repeatable results.

Aim: By combining QC with SA we aim to focus the analysis to areas of strongly represented sentiment. Additionally we can analyse the variations in sentiment across populations for each of the QC codes, allowing us to identify beneficial and harmful security practises.

Method: We code QC-annotated transcripts independently for sentiment. The distribution of sentiment for each QC code is statistically tested against the distribution of sentiment of all other QC codes. Similarly we also test the sentiment of each QC code across population subsets. We compare our findings with the results from the original QC analysis. Here we analyse 21 QC-treated interviews with 9 security specialists, 9 developers and 3 usability experts, at 3 large organisations claiming to develop 'usable security products'. This combines 4983 manually annotated instances of sentiment with 3737 quotations over 76 QC codes.

Results: The methodology identified 83 statistically significant variations (with $p < 0.05$). The original qualitative analysis implied that organisations considered usability only when not doing so impacted revenue; our approach finds that developers appreciate usability tools to aid the development process, but that conflicts arise due to the disconnect of customers and developers. We find organisational cultures which put security first, creating an artificial trade-off for developers between security and usability.

Conclusions: Our methodology confirmed many of the QC findings, but gave more nuanced insights. The analysis across different organisations and employees confirmed the repeatability of our approach, and provided evidence of variations that were lost in the QC findings alone. The methodology adds objectivity to QC in the form of reliable SA, but does not remove the need for interpretation. Instead it shifts it from large QC data to condensed statistical tables which make it more accessible to a wider audience not necessarily versed in QC and SA.

1 Introduction

Information technology has become ubiquitous within organisations. From document management to communications, virtually all aspects of business processes are touched upon by IT. These changes have created systems and data that support a huge increase in productivity which in turn makes them – and the data they contain – a target for attacks. Organisations must invest in an ongoing effort to secure IT assets and electronic data. However, security is a secondary activity for businesses, and security mechanisms that get in the way of users' and employees' business tasks are often circumvented, especially when security responsibilities accumulate over time [6]. The gains that IT affords in productivity are often undone by unusable security solutions that place excessive demands on users. The reasons for ignoring or circumventing security have been uncovered in successive studies since 1997 [1].

In various efforts to understand the elements of security usability, qualitative research methods have been used by a great number of works for the analysis of semi-structured self-reports – by individuals such as home users and company employees – of their per-

ceptions and comprehension around security [2, 5, 27] and privacy [21]. Much of this research is open-ended and investigative, although qualitative methods such as Grounded Theory offer a focused and structured approach to analysing textual data arising from these investigations [13].

Individuals are tasked not only with behaving securely, but with using IT securely and applying security technologies to support their activities. We examine the roles of security and usability in the development of IT security software in three large organisations (between 14,000 and 300,000 employees). All three organisations use a large number of off-the-shelf products, but also develop solutions in-house. In all cases the companies develop products at more than one location. The three organisations have very different customers, both governmental and private. More importantly, they prioritise security and usability very differently. The organizations range from a "security first" corporate culture with a low tolerance for deliberate security violations, to one where security is usually not the primary focus of each business unit. The studies are conducted as part of research by the Institute for Information Infrastructure Protection (I3P), and the QC analysis is published as [8]. Their main research question consider "Why each organization added usability and security elements to its software development process", "how and where the organization added them", and "how the organization determined that the resulting software was usable and secure."

The research presented in this paper builds on the QC conducted by Caputo et al. in [8]. Our contributions are as follows: we lay out our hypothesis of gaining additional insights by combining QC and SA in section 2 and describe the methodology in section 4. We perform a sentiment analysis by additionally coding the data for sentiment (section 5.3), independent of the existing QC annotations. This is followed by our results in section 6, which is finished off with a comparison of our findings with the findings from the QC exercise.

2 Aim

Our new approach combines two existing qualitative methodologies into one statistically validated model. Figure 1 depicts the methodology: Both QC and SA work on the conceptions held by people. These conceptions are concealed within (often large) bodies of text, where both methods have developed to expose specific elements of these conceptions. QC focuses on uncovering a structured theory, attempting to explain the relationships between concepts and artifacts. SA reveals the emotions towards the conceptions, revealing contentious conceptions.

In isolation both methodologies have their limitations.

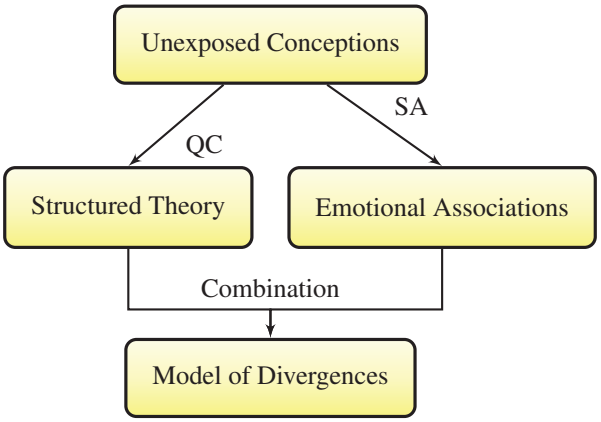


Figure 1: The output structure of our methodology

QC provides a comprehensive overview of themes which are used to construct a grounded theory of the data, yet there is scope to more directly measure the perceived importance of identified themes as according to the individuals under observation. Sentiment Analysis fills this gap: by independently measuring sentiment within the source documents, we get an accurate measure of the expressed sentiment towards each QC concept.

Our methodology combines QC and SA to provide a model of divergences, which reveals the friction points between the themes discovered by QC. The combination approach improves the reliability of traditional QC by providing metrics which further highlight important themes, and can guide application of remedial interventions to critical issues.

3 Background

This paper combines three distinct topics: QC, SA and Usable Security.

3.1 Qualitative Coding

The aim of qualitative coding is the extraction of knowledge from data. Most coding techniques iterate over textual data, where the researcher applies labels ('codes') to sections of the text (also referred to as quotations). These analysis techniques vary in the amount of interpretation of the data that is required, and by describing qualitative coding as a black-box methodology overlook the opportunity to convey the reliability of the exercise. In these cases an additional validation step as described in this research would support transparency.

The first step of the annotation process is *open coding* [25]. While developing QC, the researchers constantly refine their codes in an iterative process, questioning the choice of every code by comparing it to all other in-

stances of the same code throughout the data. The individual codes are further grouped into conceptual categories as part of a process Strauss and Corbin called *axial coding* [25].

It is at this stage that we extract the coding for our analysis. Most QC exercises span around 100 codes, spread across 15-25 categories (e.g. [20]), where these numbers of course vary depending on the size of the study. Each code is applied many times in the source documents, typically leading to many thousands of quotations. The quotations (of different codes) are often overlapping. In some sense the QC represents a very accurate topic model, but compared to statistically-based topic models, the high level of rigour and consistency of the QC represent a reliable basis for further analysis.

3.2 Sentiment Analysis (SA)

The definition of sentiment is remarkably vague: Pang and Lee describe it as "settled opinion reflective of one's feelings" [18]. Yet when identifying conflicts between usability and security, it is essential to consider an individual's sentiment as it reflects the importance of the issue to the individual. It is important to understand the dependencies between issues in order to identify the root cause. In many cases, solving a relatively minor issue that evokes strong emotional responses is a prerequisite for solving more important, but less noticed issues.

Until today, much of the research still restricts the classification space into three scores: positive, negative and objective/neutral [18]. This approach is often sufficient as the sentiment scores are aggregated over the unit of interest, such as all Twitter messages containing a specific hashtag. There are various methods for identifying sentiment. While some approaches attempt to algorithmically determine sentiment on the basis of sentiment dictionaries (where each word has a sentiment score), syntax and semantics, the most successful approaches are based on documents manually annotated for sentiment, in which case Sentiment Analysis becomes a simple annotation task with a fixed code book. These documents then form the training set of supervised learning techniques.

3.3 Complexity of Usable Security

Usable security integrates security and usability considerations with the primary task to form one continuous process [28]. The ideal outcome is an increased level of security with no loss of usability. Preserving usability and security together should enable increases in productivity as barriers are identified and eliminated.

Yet practice is far from this ideal world. Users face a huge number of barriers due to ill-fitting security in their productive processes every day; and their compliance budget [6] — the amount of rules an individual will follow before taking shortcuts — is regularly exceeded. These findings have led researchers to investigate non-compliant behaviour in greater detail, with surprising results: the individual's rationale for non-compliant behaviour can very well be rational. In fact, the non-compliant behaviour is worth studying as it reveals not just organisational failures but alternative approaches to maintaining security and usability developed by individuals themselves [15]. A further dimension to the complexity of security is the cognitive strain placed on the individual when performing security tasks. Security is rarely designed with humans' upper bounds of comfortably performing cognitive tasks in mind [7].

All of this research highlights a divergence from the intention of security to support business processes and the implementation of security. It is this security misalignment that enables those vulnerabilities that have been left unresolved, and introduces new weaknesses. Managing, formulating and implementing effective policies requires understanding of the causes and consequences of this misalignment. Modelling these asymmetric divergences is a challenging task [9], but should be the basis of any new policy considered. Resolution of this misalignment requires insights drawn from divergent fields of research, including behavioural sciences [19].

4 Methodology

This section describes the combination of QC and SA, the final step in figure 1. We present a methodology that combines the structured theory produced by QC and the emotional associations labelled through SA. This produces a more nuanced model of perceptions, crucially associating contributory factors with indicators of the perceived effort and stress associated with interacting with them, as is seen with attempts by employees to complete tasks and navigate security controls.

4.1 The Basic Unit of QC+SA

As input the methodology uses documents that have been annotated using both QC and SA. By analysing intersections between QC quotations and sentiment annotations we compute a distribution of sentiment for each QC code. As every word in the source documents has some sentiment annotated with it, each quotation (as a sequence of words) will be linked to a number of validated sentiment instances as well as to the codes.

For each QC quotation, we aggregate the sentiment annotations that are linked to it. This aggregation is described in the following section. Each QC code has many

quotations linked to it, and hence a distribution of sentiment. This distribution will be different for each QC code. Using statistical tests, we can analyse the differences in sentiment distributions for individual QC concepts, and draw conclusions concerning the emotional state of the QC codes.

Further, we can restrict the sampling space to individual organisations or individual interviews to create a comparison across different aspects. In the case study presented here for example, this allows us to compare the sentiment of developers, managers and usability professionals towards specific factors in the application development process.

Assuming that the QC analysis is conducted with sufficient rigor, a study of this kind can be repeated within and across different organisations over time and allow for direct comparisons with the previous data sets. This will allow researchers to measure the perception within the organisation of a business process before and after changes and compare results - something that has previously been difficult to do reliably.

4.2 Classifying Instances of QC+SA

Given a QC quotation, we retrieve the sentiment associated with the raw text. There may be multiple sentiments attached to a single quotation. Experiments with constructed test data have shown that the most reliable way of averaging the sentiments of quotation text is by weighting on the number of words of overlap between the sentiment text and the quotation text. This is because the boundaries of the sentiment annotations are the most unreliable part of the annotations. Even when sentiment annotators agree on the overall sentiment, the exact location of the boundary of the sentiment remains fuzzy. This gives the following formula for the sentiment of a quotation q, where S is the set of all sentiment annotations:

$$sent(q) = \frac{\sum_{s \in S : |s_t \cap q_t| > 0} s_s * w(s,q)}{\sum_{s \in S : |s_t \cap q_t| > 0} w(s,q)}. \quad (1)$$

Here s_t and q_t denote the words of the quotation or sentiment annotation, and s_s denotes the sentiment score (either -1, 0 or 1). Hence $|s_t \cap q_t|$ denotes the number of words that both the text of the sentiment annotation s and the quotation q have in common, with

$$w(s,q) = \frac{|s_t \cap q_t|^2}{\min\{|s_t|^2, |q_t|^2\}}. \quad (2)$$

The weights in equation (2) are squared to decrease the influence of small overlaps with neighbouring sentiment annotations.

For a given code c, the distribution of sentiments is just the sentiment score of each of its quotations:

$$sent(c) = \{sent(q) : q \in c_q\}, \quad (3)$$

where c_q is the set of quotations associated with c.

4.3 Worked Example

Let us consider a fictitious example which has been annotated for sentiment by two annotators:

Only when you have got a large development team you need usability experts. But they can be useful.

Here, underlines represent negative sentiment and overlines represent positive sentiment. Where there are less than two lines present, a neutral sentiment exists. Independent of the sentiment annotations, this excerpt has been annotated with two QC quotations. The first quotation q_1 spans the first sentence and is linked with the code *Development: Team size*. The second quotation q_2 begins at "usability" and contains the remainder of the excerpt. The second quotation is linked to the QC code *Actor: Usability expert*.

In order to determine the sentiment for each of these quotations, we apply equation (1). Consider the first quotation (q_1, the first sentence) of length 13. There are four sentiments present: The first spans the entire sentence, the other two end and begin to the left and right of the second 'you' respectively. The last sentiment is the neutral and spans only the word 'you' (on line 2). The length of these four sentiments (s_1, s_2, s_3 and s_4, say) are 13, 9, 8 and 1 words respectively. Equation (2) gives us the weights for each sentiment given the quotations. This gives

$$\begin{aligned} w(s_1,q_1) &= 13^2/\min\{13^2,13^2\} = 1, \\ w(s_2,q_1) &= 9^2/\min\{9^2,13^2\} = 1, \\ w(s_3,q_1) &= 3^2/\min\{8^2,13^2\} = 9/64 = 0.14, \\ w(s_4,q_1) &= 1^2/\min\{1^2,13^2\} = 1 \end{aligned}$$

as for $w(s_3,q_1)$ only 3 of the 8 words of s_3 intersect the quotation.

Given equation (1), we can now work out the average sentiment for this quotation as -0.592.

This score is the sentiment for one quotation alone — but each QC code is linked to many quotations, giving us the distributional sentiment to base our further analysis on.

5 The Three Case Studies

We applied our methodology to a study on usable software development. The study details are presented in [8]

and [20]. Three large US based organisations were selected to be studied by a team of 5 researchers with the aim of characterising usable software development activities. The study was driven by three research questions: 1. Why has an organisation added usability and security elements to its software development process? 2. How and where were they added? 3. How did the organisation determine that the resulting software was usable and secure? Supporting hypotheses examined the effect of individual roles on the development process. Research questions were then investigated by way of semi-structured interviews with individual employees of the organisations.

5.1 Data Collection

Interviews were conducted on the organisations' sites. At least three researchers were physically present at each interview. Audio recording devices were not permitted, hence three interviewers orthographically transcribed the conversations verbatim. Inconsistencies across transcripts were then reconciled to produce a merged transcript for each interview. In total, 21 interviews were conducted, with a combined length totalling 87496 words. Seven interviews were conducted at organisation A and nine and five interviews at organisation B and C respectively.

5.2 Qualitative Coding

The team used the Atlas.TI qualitative analysis software [3]. The coding was carried out by five expert coders, one of whom is one of the authors of this paper.

There is little guidance on distributing QC analysis in the literature. While Glaser and Strauss describe the procedure of QC as an iterative refinement of code/category/theme by comparison of all instances to each other, this becomes counter-productive when sections of the source text are distributed across a number of annotators.

As a trade-off between methodological accuracy and efficiency the annotators began by all coding the same interview individually. An entirely open approach was chosen, in line with Strauss and Corbin [25]. The coding choices of all five coders were discussed in a dedicated session to expose the biases of individual coders early on. Subsequently, every interview was independently coded openly by at least three coders, expanding the code book as necessary. After the open coding of each interview, the annotations were discussed in plenum to identify and resolve all differences in the meaning of open codes across coders. This step was intended to mimic the constant comparison [14] of all quotations of one code to each other though, rather than comparing all instances, here coders' code definitions were compared and unified. This process aligned each coder's individual code book, giving a unified QC that is in agreement with every coder.

This process was repeated for the axial coding of the data. Here the discrete codes were grouped into conceptual families that reflect commonalities among codes. 76 codes span 3737 quotations, roughly equally distributed between the three organisation. The codes span topics such as usability, security and topics related to the organisational structure and business processes of each organisation. The identified code families show the focus of interviews on the interactions of security and usability within organisations. There are also a significant number of codes concerning decision drivers and the focus of these decisions (i.e. goals, methods and solutions).

5.3 Sentiment Annotation

The source documents under analysis have a high degree of specialised language, being as they are driven by subject-specific expertise. Without verifying the accuracy by annotating at least a section of the primary documents manually, applying an off-the-shelf machine learning approach does not satisfy our desire to ensure accuracy.

Since sentiment annotations are inherently subjective, multiple independent annotators are required to ensure consistency and provide a score of inter-annotator agreement across annotations. As we already need to annotate a section of the source documents to verify the quality of the annotations, we decided to manually annotate the entire set of raw documents, providing us with a set of gold standard documents to base further research on.

5.4 Methodology of Manual Sentiment Annotations

The methodologies of sentiment annotations in previous work vary significantly. Strapparava and Mihalcea ask their annotators simply to annotate the given title for sentiment [24]. No further guidance or training was carried out. Annotators were free to use any additional resource. The instructions given to annotators by Nakov et al. are similarly short [17], but a list of example sentences with annotations is given.

In light of the issues presented by these two approaches we have chosen to give more detailed instructions but have refrained from giving explicit examples. The instructions given to the annotators can be found in figure 2.

For each of the documents please annotate each sentiment occurrence as either positive or negative. If you think no sentiment is present, just leave the text as it is. We are interested in the underlying, implicit sentiment. This is what the interviewee thinks about the topic at the given expression. Be generous when annotating, annotating sentiment is inherently subjective. As you are annotating transcribed speech, it may be very possible that sentiments change abruptly. Make sure that the content of the annotations should preserve the context, but this may not always be possible.

Figure 2: Annotation instructions given to annotators

5.5 Analysis

As part of our practical contribution 21 transcribed interviews have been manually annotated by 3 annotators, two of which are authors of this article. The annotations have been carried out using Atlas.TI [3] with a code book limited to *positive* and *negative*.

Annotator	total	#pos	#neg	avg #words
1	1450	731	719	17.129
2	1677	873	804	32.291
3	870	419	451	15.380

Table 1: Annotation statistics per coder

The combined length of the 21 transcribed interviews is 87,496 words. Table 1 lists the distribution of annotated phrases for each of the annotators. The difference in the number of phrases that have been annotated is surprising: annotators 1 and 2 have each annotated many more phrases with sentiment than annotator 3. While annotators 1 and 2 have a similar number of sentiment annotations, each annotation of annotator 2 spans nearly twice as many words.

These divergent results highlight the difficulty of clearly annotating sentiment. As we gave no examples of sentiment annotations to the annotators, the annotation lengths varied.

5.6 Cross Annotator Agreement

In the literature there is wide spread disagreement about the choice of metric and its interpretation [17, 26]. The two measures widely used in literature are K [22] (also called Multi-π [12]) and Fleiss' Kappa [10] (also called Multi-κ).

The annotation task described above is a multi-coder boundary annotation problem with multiple overlapping categories. The issue for this class of annotation problems is that the reliability should not be calculated token wise (unitise, as K and Kappa do), but should rather respect the blurry nature of their boundaries — annotators may agree that a specific sentiment is present, but have different begin and end tokens. One measure that does not overly penalise on non-exact boundary matches is Krippendorff's α_U [16], a non-trivial variant of α. Unfortunately we could not find a single use of this measure in the literature.

For K (or Multi-π) and Fleiss' Kappa (or Multi-κ) we can report agreement figures of 0.59 and 0.60 respectively.

	% by phrase	% by words
Perfect agreement	48.60	47.49
Majority agreement	95.55	96.42

Table 2: Agreement statistics

An alternative measure used widely is an agreement table [11]. Table 2 represents an accumulation of an agreement table. Perfect agreement represents the percentage of all-negative, all-neutral and all-positive tokens. Here agreement rates are weak, with 48% of phrases showing full sentiment agreement. To soften the measure slightly, we include a figure for majority agreement, where at least 2 of the annotators agree on the sentiment assignment of a phrase or token.

5.7 Discussion

The reliability values presented here can be classified as reasonably consistent. A recent publication does not report annotator agreement metrics but reports "accuracy bounds" without specifying their meaning or derivation [17]. It seems likely that the reported bounds of between 77% and 89% represent majority agreement, which would fit well with our findings. Interpreting K and Fleiss' Kappa is more difficult. The literature does not agree on strict bounds for these measures, but only values > 0.67 are generally seen as reliable, but other researchers argue that $0.40 < K \leq 0.60$ indicates moderate agreement [11]. Our divergence may be a result of the ambiguity of sentiment annotations. Emotions are perceived differently by individuals, partly due to different life experiences and partly due to personality issues [4] and gender [23]. Secondly, the annotation process itself may be responsible for these variations. By phrasing the annotation instructions vaguely, a large number of weak sentiments have been annotated. Rather than penalising the process for these inaccuracies, we argue that, in this subjective context, this level of uncertainty

Code	Label	Freq	p	t	r	mean
67	Usability problem	48	0.000	−6.333	0.612	−0.222
69	Usability problem: tradeoffs	13	0.000	−4.854	0.510	−0.395
45	Security problems	36	0.000	−4.452	0.478	−0.161
68	Usability problem: difficult to understand	8	0.000	−3.678	0.410	−0.380
14	Development: conflict	12	0.001	−3.424	0.386	−0.259
27	Other conflict	13	0.013	−2.479	0.290	−0.143
49	Security success criteria: Better than before	6	0.024	2.258	0.266	0.480
22	Education, training, skills	32	0.032	2.141	0.253	0.268
51	Team	36	0.043	2.028	0.240	0.252

Table 3: T-test comparing the distribution of sentiment of one code to the distribution of the sentiment scores of all other codes of organisation A. The overall mean sentiment is 0.1226 with 67 degrees of freedom.

is beneficial. Instead of aggregating the sentiment annotations, preserving the uncertainty for the down-stream applications will lead to an enhanced understanding as these tasks will utilise the existing measures of uncertainties in the statistical tests. This will allow us to be fully confident of the results of the analysis given the limitations presented here.

6 Results

In this section we discuss the results of the methodology. Results are presented in three categories: first, each organisation is analysed in isolation; second, the three organisations are compared and third, the different interviewee groups are compared across the organisations.

Each individual organisation has internal security experts, developers and usability experts for the creation of internal and commercial security management products. The development processes of these products are the focus of the interviews, specifically how the products are designed to be usable and secure, and what – if any – criteria have been used to measure usability and security.

6.1 Per-Organisation Analysis

For each QC code 'c' a t-test was conducted comparing c's distribution of sentiment (equation (3)) against the distribution of sentiment of all other codes (i.e. the union of equation (3) for all other codes) applied to that organisation. These distributions of sentiment are distributed approximately normally in $[-1,1]$. If the distribution of c differs to an extent that is statistically significant, the opinions expressed in the quotations linked to c are significantly different than the opinions expressed on average. It is these codes that tell us what the issues and concepts are that the interviewees feel strongly about.

Tables 3 and 4 shows the output of such an experiment for organisation A and B (similar results are produced for organisation C, but not presented here). Only those codes that exhibit a statistically significant variation are listed. Columns p and t give the significance of the correlation. This has been converted into Pearson's r value, indicating the strength of the correlation. The last column lists the mean of sentiment distribution of that code.

From Table 3 it can be seen that conflicts and problems are prevalent in the organisation. Usability in particular is causing a significant amount of negative emotion, as the trade-offs made between usability and security leave a negative impact on the development process. This highlights the problem of adding usability as an add-on to an existing product (as is the case in organisation A). The three codes with statistically significant positive scores contrast this: organisation A prides itself in providing better security. The provided education and training are well regarded and employees like working in their existing team.

The issues found in organisations B (table 4) and C share similarities with organisation A: Usability is an add-on to existing products. This creates conflict in the development process, as developers struggle to understand the usability problem (as the significant negative emotions for the code *Usability problem: difficult to understand* highlight). But we can identify some positive messages from these organisations too: interviewees of organisations B and C agree that *usability as a goal* is desirable and some *usability success criteria* are seen to be statistically significantly more positive. While the development process struggles to integrate usability, there are positive instances (such as *user satisfaction* in B and *better functionality* in C) where the benefit of making the product more usable is bearing fruits. Yet the *funding of resources* and the *organisational structure* (in B) as well as the *corporate culture* (in C) wear heavily on the development process.

Organisation B stands alone in the positive view of their ability to *measure security*, although their metric is

Code	p	t
Usability problem	0.000	−6.1
Security problems	0.000	−5.2
Development:conflict	0.000	−4.1
Usability problem: difficult to understand	0.000	−3.9
Usability success criteria: user satisfaction	0.001	3.5
Usability problem: tradeoffs	0.003	−3.0
Resources: Funding	0.004	−2.9
Usability goal	0.009	2.6
Other conflict	0.012	−2.5
Security methods: Measurement	0.013	2.5
organisation: structure	0.021	−2.3
Security methods: active monitoring	0.030	2.2
Security problems: access control	0.043	−2.0

Table 4: T-test of organisation B. Mean sentiment is 0.08489.

defined unscientifically as *'it is secure if we can't break it ourselves. And we continuously try'*. By pushing security as far as possible, it supersedes all other stakeholders in the product – including usability. This issue is amplified by the positive view towards this approach: the organisation is proud of their security. In contrast, employees suffer the shortcomings of the organisation's approach to security every day, as the negative view on *access control* highlights.

6.2 Cross-Organisation Comparison

With our methodology a t-test can be conducted to compare the distribution of sentiment scores between the organisations for each of the codes. In table 5, an arrow pointing upwards represents a statistically significantly more positive sentiment score compared to the sentiment scores of the other organisations; similarly an arrow pointing downwards represents a statistically significant more negative score (with $p < 0.05$). A horizontal arrow represents a non-significant change towards positive or negative. A field that is left empty represents insufficient data for this organisation and code.

The data in table 5 shows some strong trends. For the majority of statistically significant variations the quotations belonging to organisation A have more positive emotions attached. Similarly organisation C does not exhibit a single code which is more positive than in the other organisations. This pattern may point to the overall morale of the organisations in question: the sentiment portrayed by the interviewees at organisation A was a lot more positive than at organisation C. This is reflected

Code	A	B	C
Actor: developer	⟷	⟷	↓
Actor: salesperson	↑		↓
Actor: usability specialist	⟷	↑	↓
Development	↑	⟷	↓
Development: process	↑	⟷	↓
Development: requirements	⟷	↑	↓
Education, training, skills	⟷	⟷	↓
Organisation: corporate culture	↑	⟷	↓
Other conflict	↑	⟷	
Resources	↑	⟷	⟷
Resources:funding	↑	⟷	⟷
Security	↑	⟷	⟷
Security methods: measurement	↓	↑	
Team	↑	⟷	↓
Tools: development	⟷	↑	⟷
Usability	⟷	⟷	↓
Usability problem	↑	⟷	⟷

Table 5: T-test comparing the three organisations. The up, horizontal and down arrows indicate positive, neutral and negative variation respectively at $p < 0.05$.

in codes such as *Team*, *Organisation: corporate culture* and *Development*, where both A is uniquely more positive and C uniquely more negative than the other organisations.

The negative morale in C may seem unsurprising. However, the existing conflict between usability experts and the rest of the organisation is further highlighted by the relatively negative views towards the three actors types *developer*, *salesperson* and *usability specialist*. The actor *salesperson* did not show up in the analysis of each organisation in isolation, but here it suggests another source of conflict.

For organisation B only four codes display significant variations and all of these are positive. The fact that codes such as *Usability problem* are not significantly more positive for organisation B than for A and C conflicts with the significantly more positive code *Actor: usability specialist* in B. This reinforces the assessment that different aspects of usability have been accepted to different degrees. The same conclusion can be drawn for organisation A. While *Usability* is seen more positively than in the other organisations, *Actor: usability specialist* is not. The understanding of what usability means in practice is a point of contention.

6.3 Cross-Role Comparison

Here we explore the potential sources of conflict from the perspective of those who live them, by assessing the interviews according to the three interviewee role categories illustrated in table 6.

	A	B	C
Number of interviews	7	9	5
Number of developer	5	2	2
Number of security specialist	2	8	0
Number of senior usability expert	0	0	3

Table 6: Distribution of interviewee types over the organisations

All of the interviewees aligned with one of the three categories apart from in organisation B where one interviewee was classified as both a developer and security specialist. Note that the distribution of roles varies, despite an original intention for there to be an equal split [20].

Code	D	S	U
Actor: developer	⟷	⟷	↓
Actor: salesperson	⟷	⟷	↓
Actor: usability specialist	⟷	⟷	↓
Decision driver	⟷	⟷	↓
Development	⟷	↑	↓
Development: process	⟷	↑	↓
Development: requirements	⟷	⟷	↓
Education, training, skills	⟷	⟷	↓
Organisation: corporate culture	⟷	↑	↓
Security goals	⟷	↑	⟷
Security goals: preserve reputation/funding	↓	⟷	
Security success criteria: better than before		↑	
Team	↑	⟷	↓
Usability method	⟷	⟷	↓
Usability method: testing	↓	⟷	⟷
Usability success criteria	⟷	⟷	↓

Table 7: T-test comparing the three interviewee types. The up, horizontal and down arrows indicate positive, neutral and negative variation respectively at $p < 0.05$.

Table 7 follows the format of the previous section but with *D, S & U* standing for *Developer*, *Security expert* and *Usability expert* respectively. The table summarises the perspectives across all interviewees who share each role classification. In the case of usability experts all statistically significant variations are more negative, more so than for developers and security specialists. This may be linked to the results of the analysis between organisations: the only three usability experts in our data set were at organisation C.

It is clear that security experts have the most positive view. This reinforces our assertion that the focus of product development remains on security and that the development process is tailored towards security over integrating usability. The positive feeling towards *corporate culture* supports this. The negative emotions of the developers towards *usability method: testing* highlight an additional shortcoming, in that developers fail to see any benefit in usability testing and instead regard it as adding additional strain.

While powerful comparison tools, tables 5 and 7 do suffer from a potential bias due to the lower number of interviews that make up the separate organisations and employee types. Further, for each of these tables up to 228 t-tests were performed which raises the chance of false positives. Yet even with a conservative Bonferroni correction, some interesting artifacts remain statistically significant, shrinking the number of significant variations in tables 5 and 7 by approximately one third. Further, as described in the following section, the results are mostly in line with the pure qualitative analysis, validating the approach chosen.

6.4 Comparison to QC Findings

Here we summarise the findings from the complementary quantitative coding work [8] and discuss the additional benefits of our approach. Caputo et al. hypothesised three distinct explanations of why changes in the software development process might lead to more usable security (from [8]): 1. The "key individual" theory: Improved outcomes resulted not from the process changes but instead from the efforts of a single individual who cares about usable security; 2. The "experienced team" theory: Improved outcomes resulted not from the process changes but instead from the team's prior experience in building usable security, and; 3. The "incentives" theory: Improved outcomes resulted not from the process changes but from incentives placed on team performance with respect to usable security.

Of note is that none of these theories were confirmed in their analysis. Our results agree: organisation C is the only organisation with usability experts, and for this organisation the positive codes are *usability expert to develop software* as well as *use cases* (see section 6.1). As we stated previously, negative codes such as *Usability-security trade-offs* and *development conflicts* highlight that their impact is small. In general when comparing the three organisations in table 5 or the three different

employee types in table 7 we do not find indications to support any of the three theories. Hypothesis 1) can be analysed particularly well by our methodology as it investigates emotions towards security — exactly what our methodology focuses on through its use of SA. The original quantitative analysis did not consider the use of sentiment.

Rather, the authors present a list of five findings: 1. Usability is a grudge sale: only when losses in sales could be linked to a lack of usability, did the organisation respond; 2. The negative effects of a lack of usability occur at the organisational level and are not passed on to the developers. Hence there are no incentives to deliver usable software. This is the exact opposite of the third hypothesis above; 3. Wildly varying definitions of usability; 4. Lack of knowledge by developers of capabilities and limitations of human perception and cognition, primary task, and context of use, and; 5. Developers think they know users because they use the software themselves.

Our methodology provides a more detailed picture, detailing the extent of the "divergences" illustrated in figure 1. We are able to assert that the interviewees in fact acknowledge the importance of building a usable application (see the analyses of table 3 above in section 6) - but when it comes to security, they lack knowledge on how to reconcile what they conceive as competing demands. Our analysis shows this stems from a number of factors: there is no definition of the usability problem, and there is an existing belief that security 'comes first' in the organisations' priorities, and hence in the development processes. There are some positive notes however: in organisation C, *personas* were perceived positively as a usability tool to aid the development process.

The methodology also facilitated analysis of the differences between the three organisations. While the original study [8] attempted to identify exemplary development processes that integrate security and usability, the authors did not find practices that could be recommended. Yet our analysis detects positive differences — in terms of partial improvements that can serve as building blocks for an integrated development process. One could argue that these may have been found with a more rigorous qualitative analysis, but a quantitative approach simplifies the task of comparing across three organisations.

Caputo et al. finish with some open questions which can be answered by our methodology. They speculate that the integration of usability into the software development process is less important than having motivated developers and usability specialists. We can support this hypothesis: Our data has shown that the conflict between usability and security centers around the individual employees and the organisational culture, rather than the software development process. The addition of usability experts to organisation C has shown positive effects on usability tools, as well as codes such as *personas*. Resolving the misconception of a security-usability tradeoff will go a long way to improving usability of security.

Caputo et al.'s second open question concerns cultural barriers to usable security. This manifests in different perceptions of usability throughout the organisation. Our analysis between the different types of employees certainly answers this open question: there are clear differences between the developers, security and usability experts that we described in section 6.3 and table 7.

7 Conclusion

We have introduced a new methodology: by performing an additional level of analysis on Qualitative Coding (QC) with Sentiment Analysis (SA), we can gain additional insight into the emotional colouring of statements.

As a proof-of-concept, we performed this analysis on QC text from 21 interviews with developers, security experts and usability experts in 3 organisations. Whilst the QC analysis uncovered that all 3 organisations were able to 'talk the talk', 'walking the walk' of usable security was a different matter. There were no usability criteria, and few usability methods were employed during the development of the products we discussed.

Our analysis agrees with many of the original QC findings, but from the QC exercise condenses the data requiring interpretation into a number of tables of statistically significant rows. This mechanism serves as a filter for pointing out specific findings that were missed in the original QC analysis. We are able to approach the original dataset from different angles, and compare aspects across organisations and employee types allowing us to draw additional conclusions through cross-comparison.

Through our methodology, security and policy managers can pinpoint friction points and conflicts in organisation processes not only through interview studies, but also other shared communications platforms such as corporate forums and dedicated support channels. For security researchers, this repeatable method offers a powerful tool that generates verifiable quantitative results to harden the results of qualitative analysis. We also explore the potential to transfer findings across organisations to different teams, where the methodology can identify aspects of professional cultures shared across separate organisations.

For future work, appropriate reliability metrics for QC are needed, to ensure that future studies can be compared by the quality of the annotations. There is also room to explore the analysis further - cross-linking different codes and the sentiment annotations could potentially create a powerful deductive tool for researchers, although visualising multi-dimensional relationships is non-trivial. As analysis becomes more elaborate there is

the challenge not just of gathering more source data, but also annotating it. Future research may then explore how machine learning can be used to automate annotation.

8 Acknowledgements

We would like to thank the LASER program comittee, and in particular our shepherd. The authors are supported in part by UK EPSRC grants, no. EP/G037264/1 and no. EP/K006517/1. We would like to acknowledge the contribution of Adam Beautement, toward the manual sentiment annotations, as well as Deanna Caputo, Shari Lawrence Pfleeger, Paul Ammann and Jeff Offutt who performed the interviews and coded the transcripts as part of the I3P project funded by NIST and DHS.

References

1. Adams, A., and Sasse, M. A. Users are not the enemy. *Commun ACM*, 42(12), 1999: 40–46.
2. Ashenden, D., and Sasse, M. A. CISOs and organisational culture: their own worst enemy? *Computers & Security*, 39, Part B, 2013: 396–405.
3. ATLAS.ti GmbH. ATLAS.ti. Berlin, 2013.
4. Barrett, L. F. Valence is a basic building block of emotional life. *J Res Pers*, 40(1), 2006: 35–55.
5. Bartsch, S., and Sasse, M. A. How users bypass access control and why: the impact of authorization problems on individuals and the organization. *Proceedings of the 21st European Conference on Information Systems*. 2013, Paper 402.
6. Beautement, A., Sasse, M. A., and Wonham, M. The compliance budget: managing security behaviour in organisations *Proc. NSPW '08*, 47–58.
7. Benenson, Z., Lenzini, G., Oliveira, D., Parkin, S., and Uebelacker, S. Maybe Poor Johnny Really Cannot Encrypt: The Case for a Complexity Theory for Usable Security *Proc. NSPW '15*, 85–99.
8. Caputo, D., Pfleeger, S., Sasse, A., Ammann, P., and Offutt, J. Barriers to usable security: three organizational case studies. under Review. 2016.
9. Caulfield, T., Pym, D., and Williams, J. Compositional security modelling. In: *Human Aspects of Information Security, Privacy, and Trust*. Ed. by T. Tryfonas and I. Askoxylakis. Vol. 8533. LNCS. Springer, 2014, 233–245.
10. Davies, M., and Fleiss, J. L. Measuring agreement for multinomial data. *Biometrics*, 38(4), 1982: 1047–1051.
11. Di Eugenio, B., and Glass, M. The kappa statistic: a second look. *Comput. Ling.* 30(1), 2004: 95–101.
12. Fleiss, J. L. Measuring nominal scale agreement among many raters. *Psychol Bull*, 76(5), 1971: 378–382.
13. Glaser, B. G., and Strauss, A. L. The discovery of grounded theory: strategies for qualitative research. Chicago: Aldine Transaction, 1967.
14. Harry, B., Sturges, K. M., and Klingner, J. K. Mapping the process: an exemplar of process and challenge in grounded theory analysis. *Educational Researcher*, 34(2), 2005: 3–13.
15. Kirlappos, I., Parkin, S., and Sasse, M. A. Learning from "shadow security": why understanding non-compliance provides the basis for effective security. *Proc. USEC*. 2014.
16. Krippendorff, K. On the reliability of unitizing continuous data. *Sociol Methodol*, 25, 1995: 47–76.
17. Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., and Wilson, T. SemEval-2013 task 2: sentiment analysis in twitter. *Proc. SemEval*. 2013, 312–320.
18. Pang, B., and Lee, L. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 1-2. 2008.
19. Pfleeger, S. L., and Caputo, D. D. Leveraging behavioral science to mitigate cyber security risk. *Computers & Security*, 31(4), 2012: 597–611.
20. Pfleeger, S. L., and Sasse, M. A. Studying usable security: how to design and conduct case study. under Review. 2016.
21. Renaud, K., Volkamer, M., and Renkema-Padmos, A. Why doesn't jane protect her privacy? *Privacy Enhancing Technologies*. 2014, 244–262.
22. Siegel, S. Nonparametric statistics for the behavioral sciences. 2nd ed. In collab. with N. Castellan. New York ; London: McGraw-Hill, 1988.
23. Stoppard, J. M., and Gruchy, C. D. G. Gender, context, and expression of positive emotion. *Pers Soc Psychol Bull*, 19(2), 1993: 143–150.
24. Strapparava, C., and Mihalcea, R. Learning to identify emotions in text *Proc. SAC '08*, 1556–60.
25. Strauss, A., and Corbin, J. M. Basics of qualitative research: grounded theory procedures and techniques. Thousand Oaks, CA, US: Sage Publications, 1990.
26. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A. Sentiment strength detection in short informal text. *J. Am. Soc. Inf. Sci.* 61(12), 2010: 2544–2558.
27. Wash, R. Folk models of home computer security *Proc. SOUPS '10*, 11:1–11:16.
28. Yee, K.-P. Aligning security and usability. *IEEE S&P*, 2(5), 2004: 48–55.

Effect of Cognitive Depletion on Password Choice

Thomas Groß
Newcastle University

Kovila Coopamootoo
Newcastle University

Amina Al-Jabri
Newcastle University

Abstract

Background. The Limited Strength model [3] of cognitive psychology predicts that human capacity to exert cognitive effort is limited and that decision making is impeded once high depletion is reached.
Aim. We investigate how password choice differs between depleted and undepleted users.
Method. Two groups of 50 subjects each were asked to generate a password. One group was cognitively depleted, the other was not. Password strength was measured and compared across groups.
Results. Using a stepwise linear regression we found that password strength is predicted by depletion level, personality traits and mood, with an overall adjusted $R^2 = .206$. The depletion level was the strongest predictor of password strength (predictor importance 0.371 and $p = .001$). Participants with slight effortful exertion created significantly better passwords than the undepleted control group. Participants with high depletion created worse passwords than the control group.
Conclusions. That strong depletion diminishes the capacity to choose strong passwords indicates that cognitive effort is necessary for the creation of strong passwords. It is surprising that slight exertion of cognitive effort prior to the password creation leads to stronger passwords. Our findings open up new avenues for usable security research through deliberately eliciting cognitive effort and replenishing after depletion and indicate the potential of investigating personality traits and current mood.

1 Introduction

Users often set easy-to-remember passwords constructed for example from their wife's name, or recycle and re-use passwords across services [1]. These are predictable and easily guessed. This is because the panoply of separate services mean that users have a list of accounts to manage each with their own login credentials. However, managing and remembering a large number of complex passwords remain a challenge. So far, the question has not been addressed how users create passwords when they are cognitively tired or depleted. In fact, it is an open question whether cognitive effort is *necessary* for the creation of strong passwords.

The limited strength model of cognitive psychology states that human capacity to exert cognitive effort is limited and that decisions as well as effortful tasks are impeded under cognitive depletion [2]. We report on a study with $N = 100$ participants designed to measure the strength of passwords set by cognitively depleted versus non-depleted users. We hypothesise $\mathbf{H}_1$: *Cognitively depleted users create weaker passwords than non-depleted users*.

We replicate existing methods from cognitive psychology [3, 2, 15, 26] to induce cognitive depletion, in particular with thought suppression, impulse control and cognitively effortful tasks. We check depletion manipulation via a Brief Mood Inventory [26]. We measure password strength across groups using a password meter. We evaluate the im-

pact of cognitive depletion on password strength.

Contribution. Our findings indicate that slight exertion of cognitive effort leads to significantly better passwords than an undepleted control group. High depletion leads to worse passwords than an undepleted control group. This is the first study to show the impact of cognitive effort and depletion on password creation. It highlights an important factor for password and usable security research that has not been addressed to date.

2 Background

This section looks at literature on password strengths in relation to users' ability to set and remember them. We then introduce cognitive effort and explain the state of ego depletion. Lastly we review how affect and personality traits influence depletion states and decisions.

2.1 Strength & Memorability

The use of text usernames and passwords is the cheapest and most commonly used method of computer authentication. The average user has 6.5 passwords, each shared across 3.9 different sites, each user has 25 accounts requiring passwords and type 8 passwords per day [10]. Users have to not only remember the passwords but also the system and userid associated, which password restriction apply to which system and whether they have changed a password and what they have changed it to [1].

Recalling strong passwords is a humanly impossible task since non-meaningful items are inherently difficult to remember [22]. When forced to comply to security policies such as monthly password reset, a large number of users are frustrated [13]. They use strategies such as writing passwords down, incrementing the number in the password at each reset [1], storing passwords in electronic files and reusing or recycling old passwords [13]. While it is possible to create strong and meaningful passwords using pseudo-random combinations of letters, numbers and characters that are meaningful only to the owner [29], four to five passwords are the most a typical user can be expected to use effectively [1].

Thus memory issues impede the strength of password chosen by the user. To help users in setting strong passwords and aid memorability, graphical passwords have been proposed. Methods such as *draw-a-secret* are an improvement on usability of password authentication. Password strength is improved too as even a small subset of graphical passwords constitutes a much larger password space than dictionaries of textual passwords [14]. They work on the principle that humans can remember pictures better than text [24]. However no research has investigated how effortful setting password is for the user nor how depletion states impact password choice and subsequent memorability.

2.2 Cognitive Effort and Depletion

Human beings have a limited store of cognitive energy or capacity [2]. Self-control tasks, choice and decision-making draw from this inner resource. Tasks requiring self-control tasks span across spheres such as controlling attention, emotions, impulses, thoughts and cognitive processing, choice and volition and social processing [3]. In general, all tasks that are cognitively effortful—and thereby *System 2* in the terminology of the dual-process model—draw from the limited cognitive energy. As a muscle that gets tired with exertion, self-control tasks cause short-term impairments in subsequent self-control tasks. This is termed a state of *ego depletion* or *cognitive depletion*. There are levels of depletion beyond which individuals may be unable to control themselves effectively, regardless of what is at stake [3] and in unrelated sphere of activity [2]. This phenomenon has been observed in areas of over-eating, -drinking, -spending, underachievement, and sexuality [3].

An underlying question of this research is whether the creation of strong passwords is a cognitively effortful task. If that is the case, then we expect to observe that the creation of passwords is impaired under cognitive depletion. On such an observation, we can further conclude that cognitive effort is necessary for password creation. Given that cognitive depletion permeates different activities and is yielded by a variety of self-control tasks, we can

therefore expect that password creation will be impaired by the user's other effortful activities.

2.2.1 Beliefs

In this context, it is an important question whether all people are equally cognitively depleted. Interestingly, a person's beliefs have an influence on the level of that person's cognitive depletion. There is a line of research in (motivational) psychology investigating the impact of beliefs on the nature of human attributes. A classical example is the belief whether intelligence is fixed or malleable [5, 9]. It turns out that implicit beliefs about willpower as a limited resource [15] impact the extent of cognitive depletion. Consequently, individuals who believe in unlimited willpower are less affected by cognitive depletion. This effect impacts our experiment because participants are not equally affected by the manipulation inducing cognitive depletion, and we expect to see differing depletion levels in the experiment group.

2.2.2 Personality Traits

While beliefs or mindsets of persons constitute personality traits already, we expect other personality traits to influence the capacity to bear cognitive effort as well as the strength of chosen passwords. Capacity theories of self-control conceptualise it as a dispositional trait like construct that differ across individuals. Thus people high in dispositional self-control will have more resources at their disposal than individuals lower in trait self-control. In addition, certain people are dispositionally motivated to act in a certain way such as over-eating, -drinking. Personality traits has already been linked with security research, for example impulsive individuals are more likely to fall for phishing e-mails while trait-based susceptibility to social engineering attacks is recognised [27].

2.2.3 Affect

Security tasks including password security often leads to user frustration [13]. While affect states impact decisions, they also influence cognition [23]. Affect states enable recall of mood congruent information that might influence judgments, or the heuristic adopted to make decisions [6].

In addition the active regulation of emotion or mood deplete self-control resources and invoke ego depletion [2]. Regulating affect often requires the individual to overcome the innate tendency to display emotions while negative affect, induced by demanding and frustrating tasks, is implicated in development of ego depletion.

3 Method

3.1 Participants

The sample consisted of university students, $N = 100$, of which 50 were women. The mean age was 28.18 years ($SD = 5.241$) for the 83 participants who revealed their age. The participants were balanced by gender and assigned randomly to either the depletion ($n = 50$) or control ($n = 50$) condition. They were mostly non-computer science students from Newcastle and Northumbria University, of mainly international background (common countries included Oman, China and Iraq). Tiredness and cognitive depletion over the course of a day are affected by the participants' circadian rhythm. Hence to control the confounds of the circadian rhythm, the experiment runs were balanced in time-of-day for depletion ($M = 4.167$, $SD = 1.403$) and control ($M = 4.167$, $SD = 1.642$) conditions. We ran a Wilcoxon signed-rank test on the two conditions matched by time of day. We find that the distribution of participants across the two groups was not statistically different, with $Z = 0.00$ and $p = 1.00$.

3.2 Procedure

The experiment was designed to enable a comparison of the influence of cognitive depletion on password strength. The experiment group was artificially cognitively depleted with tasks that required impulse control while the control group was not depleted, completing non-depleting tasks with similar length and flavour.

The procedure consisted of (a) pre-task questionnaires for demographics and personality traits, (b) a manipulation to induce cognitive depletion, (c) a manipulation check on the level of depletion, (d) a password entry for a mock-up GMail registration, and (e) a debriefing and memorability check one week after the task with a GMail login mockup. Figure 1 depicts the experiment design.

3.2.1 GMail Registration Task

Participants were asked to generate a new password for a Google Mail (GMail) account, on a mock-up page which was visually identical to a GMail registration. The participants were told (a) to create the account carefully and fill in all the fields; (b) to give correct and valid information; (c) that the account is highly important; and (d) that they should ensure they can remember the password. Participants were also asked to return to the lab one week after the registration task. Registered e-mail address and password were recorded. The strength of the password was measured.

3.2.2 Inducing Cognitive Depletion

We induce cognitive depletion for the experiment condition, reproducing manipulation components of Baumeister et al. [26]. In the experiment condition, the participants are asked to suppress thoughts, control impulses to follow a learned routine and to execute a cognitively effortful Stroop task. In the control condition, the participants fulfil tasks with a similar structure, flavor and length, however without the depleting conditions.

As discussed in Section 2.2.1, we expect participants in the experiment condition to be affected by the induction of cognitive effort to differing degrees. Especially the implicit theories about willpower have been shown to have a significant effect on cognitive depletion [15]. Consequently, we will control the strength of the manipulation with a manipulation check based on a brief mood inventory (Section 3.3.2) evaluated in the Results Section 4.1.

1. Thought supression task. In the experiment condition, the participants are shown a lot white bear and asked *not* to think of the white bear, a procedure following Wegner et al. [28]. They are to raise their hand should they have thought of the white bear and failed to suppress the thought. In the control condition, the participants are asked to record whenever they think about a white bear, but not instructed suppress it. The control condition, is not cognitively depleting as the participants do not need to suppress their thoughts.

2. Impulse control task. This task is adapted from Muraven et al. [21]. Participants are asked to cross out all letters 'e' in a complex statistical text for five minutes. This establishes a learned routine. Then the participants are given another statistical text. In the experiment condition, the participants are asked to follow a new rule, to cross out all letters 'e' unless they are adjacent to a vowel. This rule interferes with the learned routine and asks the participants to exercise impulse control on it, which is depleting. In the control condition, the participants are asked to follow the same routine to cross out all letters 'e'. This rule does not require impulse control and is thereby non-depleting.

3. Cognitively effortful task. We used the Stroop task [25] as cognitively effortful task. Participants are asked to voice the printed color of a color word. The Stroop condition is that the name of a color (e.g., 'red') is printed in a color not denoted by the name (incongruent color and name). This task is a cognitively effortful when the Stroop condition is fulfilled. The experiment condition involved answering 10 Stroop items with the Stroop condition. The control condition involved answering 10 items without Stroop condition.

3.3 Measures

3.3.1 Password Strength

We tested multiple password meters, such as the Microsoft password meter and finally settled for the password meter Web site[1] because it uses an interval scale and makes the components of the password score transparent. Each component, such as 'number of characters' or presence of 'numbers', gives a bonus or malus for the overall score. All

[1] `http://www.passwordmeter.com`

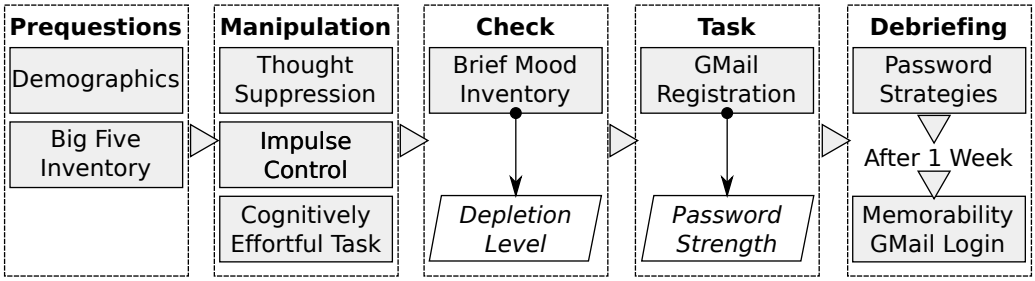


Figure 1: Overview of the experiment procedure. The control group did manipulation tasks with similar structure and flavor, yet without the depleting condition.

component scores were recorded individually and their sum computed as password score. Whereas the password meter itself caps the scores at 0 and 100, our final score could be negative or greater than 100.

The password meter does not account for weaknesses such as the use of dictionary words or personal identifiable information (e.g., name, username) as part of the password. By the NIST password guidance [7], those conditions, especially failing the dictionary test, make weak passwords. Hence, we adjusted the obtained password scores with penalties if the password contained:

- an unmodified dictionary word (-25),
- part of the user's real name (-50),
- the username (-50), or
- the user's student id (-50).

The dictionary words we checked were the large list of the Openwall wordlist collection[2], intended primarily for use with password crackers such as John the Ripper and with password recovery utilities. For the username, we argue that this information is often at the disposal of an adversary in offline attacks. For instance, for the Linux `/etc/passwd` file, the username is the first field of each entry, the real name in often encoded in the comment field. We obtain a final password strength score on an interval scale, with values between -100 and 150. The password strength obtained from this procedure was nearly normally distributed across the participants.

In addition to the password meter score, we evaluated the NIST password entropy according to the heuristic given in the NIST Special Publication 800-63 [7] and submitted the passwords to the CMU Password Guessability Service (PGS) [19], however both methods offered limited differentiation across the range of password strengths.

[2] `http://www.openwall.com/wordlists/`

3.3.2 Brief Mood Inventory

Earlier research found that cognitive depletion can be checked with a brief mood inventory, either the Brief Mood Introspection Scale (BMIS) [20, 2] or a short form. We use a short form of a brief mood inventory used as manipulation check in Baumeister's research [26], including the dimensions (a) excited, (b) thoughtful, (c) tired, (d) happy, (e) worn out, (f) sad, (g) angry, (h) calm, rated on 5-point Likert-type items between 1 Disagree strongly and 5 Agree strongly, with 3 Neither agree nor disagree as central point. Baumeister el al. [26] found that tiredness and feeling worn out are significantly affected by cognitive depletion and can therefore be used as self-report manipulation check.

3.3.3 Big Five Inventory

The personality traits of the users were queried with a 60-item Berkeley Big Five Inventory (BFI) [11, 16, 17]. The inventory measures the traits (a) Openness to experience, (b) Conscientiousness, (c) Ex-

traversion, (d) Agreeableness, and (e) Neuroticism, with a 5-point Likert-type items between 1 Disagree strongly and 5 Agree strongly computing the scores as means of items for each domain.

4 Results

All inferential statistics are computed with two-tailed tests and at an alpha level of .05.

4.1 Manipulation Check

We used the brief mood inventory introduced in Section 3.3.2 as manipulation check on the cognitive depletion, following a methodology of Baumeister et al. [26].

A comparison across groups on tired and worn out suggested that the manipulation was successful (Mann-Whitney U, two-tailed, tired: $U = 368, Z = -6.299$, significance $p = .000 < .05$; worn out: $U = 669, Z = -4.145$, significance $p = .000 < .05$). As expected following Baumeister et al. [26] in the use of the brief mood inventory: the moods of feeling tired and feeling worn out were found to be significantly higher in the depleted group than in the control group. The effect size of the manipulation for reporting feeling tired is $r = 0.63$ and for feeling being worn out is $r = 0.42$. That constitutes a large effect on feeling tired and a medium to large effect on feeling worn out. Consequently, this suggests that the cognitive depletion of the participants has been induced by the manipulation.

4.2 Password Strength Score

The distribution of the Passwordmeter password strength score is measured on interval level and is not significantly different from a normal distribution, Saphiro-Wilk, $D(100) = .99$, $p = .652 > .05$. The distribution of the Password Guessability Service (PGS) results are measured on interval level and significantly different from a normal distribution, Saphiro-Wilk, $D(100) = .59$, $p = .000 < .05$. We continue the analysis with the Passwordmeter password strength score. We computed Levene's test for the homogeneity of variances. For the password meter scores, the variances were not significantly unequal (a) for experiment and control condition, $F(1,98) = 3.369$, $p = .069 > .05$, and for (b) for gender, $F(1,98) = 1.378$, $p = .243 > .05$.

Univariate Analysis of Variance (GLM). We conducted a Univariate Analysis of Variance (GLM) with Type III Sums of Squares—robust against unequal sample sizes—with password strength as dependent variable. We used condition, gender and the Brief Mood Inventory (BMI) items as fixed variables, the Big Five Inventory (BFI) as covariates.

(a) There was a significant effect of gender, $F(1,60) = 6.824$, $p = .011$, partial $\eta^2 = .102$. (b) We observed a significant effect of BMI_Tired, $F(4,60) = 3.687$, $p = .009$, partial $\eta^2 = .197$. (c) BMI_Calm had a significant effect, $F(4,60) = 4.264$, $p = .004$, partial $\eta^2 = .221$. Other factors did not show significant effects. The corrected model offered a variance explained of $R^2 = .533$ (adjusted $R^2 = .229$).

4.3 Automated Linear Regression.

The impact on the password strength score was analyzed with a multi-predictor forward stepwise linear regression. The linear regression has an adjusted $R^2 = .206$. The studentized residual was close to a normal distribution. The outcomes of the gender, Big Five (5) and brief mood inventory (8) were predictors on the password strength score as target variable. The gender did not have a significant effect in the linear regression. We provide coefficients of the linear regression in Table 3. The extended version of this paper [12] contains detailed tables of the regression results. We give details of the automated data preparation of the regression first and will subsequently describe the effects in decreasing order of predictor importance.

Depletion Level from Brief Mood Inventory The SPSS automated data preparation of the linear regression merged categories of BMI_Tired to maximise the association with the target. We accept this grouping and name the cases introduced by SPSS and call it the *depletion level* of (a) non-depleted, (b) effortful, and (c) depleted. Strongly

disagree, disagree slightly and neither agree nor disagree were grouped as BMI_Tired_T = 0. We label this case as depletion level non-depleted. The agree slightly of BMI_Tired was transformed into BMI_Tired_T = 1, which we label as depletion level effortful. BMI_Tired of Agree strongly was transformed into BMI_Tired_T = 2, which we label as depletion level depleted.

The extended version of this paper [12] contains further analysis of justification of this grouping.

Accepting the grouping of the SPSS automated data preparation, we have: Of the control group, 49 participants were consequently rated non-depleted; 0 participants were rated as effortful; 1 participant was rated as depleted. Of the experiment group, 23 participants were rated non-depleted; 17 participants were rated as effortful; 10 participants were rated as depleted. Table 1 contains an overview of descriptive statistics over these groups.

Effects of Cognitive Depletion. The depletion level was indeed the most important predictor in the regression (significance $p = .001 < .05$, predictor importance$= 0.371$). The effortful level, that is only slightly depleted, had a coefficient of 50.65 (significance $p = .000 < .05$). The non-depleted level had a coefficient of 31.62 (significance $p = .006 < .05$).

We summarize the descriptive statistics of password strength score by depletion level in Table 1 and depict them in Figure 2. The descriptive statistics on password guessability determined by PGS show the same overall outcome in terms of means of password guesses as well as percentage of passwords determined as unguessable (cf. Table 2).

Effects of Mood. BMI Thoughtfulness and Calmness had significant effects. Strong disagreement to thoughtfulness implied stronger passwords (significance $p = .018 < .05$, predictor importance $= 0.251$, coefficient 40.072). Strong disagreement to calmness implied stronger passwords (significance $p = .012 < .05$, predictor importance $= 0.172$, coefficient 38.799).

Effects of Personality Traits. Of the Big Five personality traits, the BFI Agreeableness score was the most important predictor on the password strength (significance $p = .025 < .05$, predictor importance $= 0.137$, coefficient 14.649), where higher agreeableness significantly implied stronger passwords. The BFI Extraversion was a notable yet non-significant negative predictor on password strength (significance $p = .108 > .05$, predictor importance $= 0.069$, coefficient -11.538).

5 Discussion

This study applied the methodology of previous cognitive depletion studies [3, 2, 15, 26] to a ubiquitous security context. We observe that cognitive effort is a major predictor of password strength. Moderate cognitive exertion leads to stronger passwords than in an non-depleted and in depleted states. Strong depletion leads to weaker passwords than in moderate exertion and non-depleted states.

This is in accord with Kahneman's observation that initial effortful activity introduces a bias towards exerting further cognitive effort [18]. This outcome can also be explained with Selye's arousal curve [8], an inverse U-shaped relation between the activity of the stress system and the quality of a human's performance, yielding an optimum performance under moderate stress. This result vouches for further investigation, in particular to what extent this observation can be operationalized to improve the quality of password choice.

Our analysis also showed the impact of mood and personality, hence indicates the importance of studying other human dimensions. The results on the brief mood inventory are surprising in themselves, in particular because participants who reported themselves as not thoughtful or not calm chose better passwords. This result can substantiate the explanation of Selye's arousal curve as a possible explanation. In any case, these results ask for the investigation of the influence of current stress and mood on password choice, in particular whether negative emotions such as fear are involved.

For personality traits, it is plausible that BFI Agreeableness has an impact on the password strength, because the NEO PI classifies Compliance one of its facets, and hence postulates a tendency to avoid conflict and cooperate. Therefore, we assume users with high agreeableness to comply to password policies, as well. However, the results on the

Table 1: Descriptive statistics of password strength via password meter by condition and depletion level.

Condition	Depletion Level	N	Mean	Std. Dev.	Std. Error	Min	Max
Control	Non-depleted	49	40.65	30.97	4.43	-45	121
	Depleted	1	16.00	-	-	16	16
	Total	50	40.16	30.87	4.37	-45	121
Experiment	Non-depleted	23	33.30	33.81	7.05	-19	102
	Effortful	17	57.12	45.07	10.93	-24	138
	Depleted	10	11.10	45.97	14.54	-64	70
	Total	50	36.96	42.99	6.08	-64	138
Total		100	38.56	37.27	3.73	-64	138

Table 2: Descriptive statistics of password guessability determined by the CMU Password Guessability Service (PGS) by condition and depletion level. PGS declares passwords "unguessable" at 2.E+13 guessing attempts.

Condition	Depletion Level	N	Mean	Std. Dev.	Std. Error	Unguessable Pwds % (#)
Control	Non-depleted	49	4.85E+12	7.19E+12	1.03E+12	22.4% (11)
	Depleted	1	40945471	-	-	0% (0)
	Total	50	4.75E+12	7.15E+12	1.01E+12	22% (11)
Experiment	Non-depleted	23	2.37E+12	5.69E+12	1.19E+12	13%(3)
	Effortful	17	7.83E+12	8.49E+12	2.06E+12	47.1% (8)
	Depleted	10	1.71E+12	5.22E+12	1.65E+12	9.1% (1)
	Total	50	4.09E+12	7.11E+12	1.01E+12	24% (12)
Total		100	4.42E+12	7.10E+11	7.10+11	23% (23)

Table 3: Coefficients of the Automated Forward Stepwise Linear Regression.

Model Term	Cases	Coef.	Std. Err.	t	Sig.	95% Conf. Interval		Imp.
						Lower	Upper	
Intercept		-28.064	36.131	-0.777	.439	-99.833	43.705	
BMI_Tired	non-depleted	31.623	11.331	2.791	.006	9.115	54.132	0.371
BMI_Tired	effortful	50.650	13.477	3.758	.000	23.880	77.420	0.371
BMI_Thoughtful	disagree strongly	40.072	16.704	2.399	.018	6.892	73.252	0.251
BMI_Thoughtful	neither/nor, slightly agree	19.405	8.514	2.279	.025	2.493	36.318	0.251
BMI_Calm	disagree strongly	38.799	15.187	2.555	.012	8.632	68.967	0.172
BFI_Agreeableness		14.649	6.415	2.283	.025	1.906	27.392	0.137
BFI_Extraversion		-11.538	7.117	-1.621	.108	-25.675	2.600	0.069

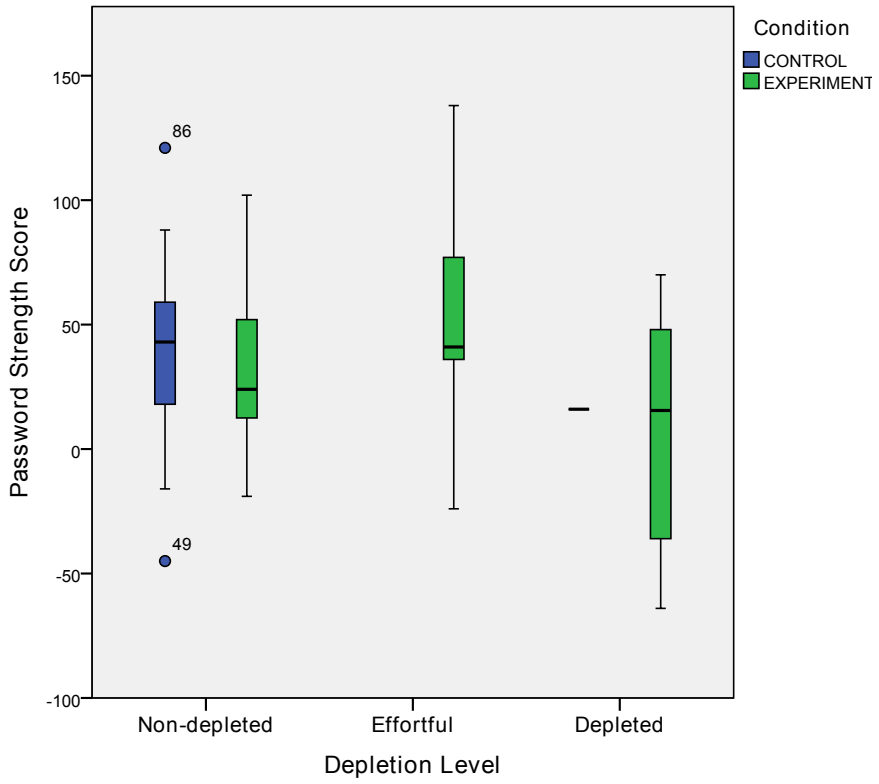


Figure 2: Boxplots of the password strength score by depletion level and experiment condition. 49 participants of the control condition were non-depleted, one participant was depleted. 23 participants of the experiment condition were non-depleted, 17 classified as effortful, 10 as depleted.

BFI ask for further investigation, as the experiment cannot distinguish whether the participants sought to please the experimenter, constituting a confounding variable, or whether the effect of compliance persists in real-world scenarios. It is notable that BFI Conscientiousness, the tendency to show self-discipline and be dutiful, did not have a significant effect on the password strength.

5.1 Ethics

The experiment followed the ethical guidelines of the university and has received ethical approval. The participants were informed that personal identifiable data will be stored in hard and soft copy and have consented to the experiment procedures. The participants were informed of the rough experiment effort and the requirement to come back to the lab in a week, before choosing to participate. The participants were paid a time compensation of $15 for partial completion and $23 for completing all components of the experiment. The participants data in hard and soft copy was stored securely in an office under lock and key, on stationary machines or laptops with full hard disk encryption. The participants passwords were stripped from username and other PII before being uploaded to CMU's Password Guessability Service (PGS). The data was deleted from CMU's servers after 14 days.

5.2 Ecological Validity

We developed a mockup of GMail, which was visually identical to GMail's account registration page. In this sense the experiment is generalisable to real-life settings. Even though the experimenter did not disclose that the GMail registration was a mockup, we cannot exclude that participants might have noticed that it was not the real GMail registration page. The experiment included a memorability check for which the participants were asked to return to the lab one week after the registration task. They were to enter the set password in a GMail login mockup. The participants were made aware of this requirement in the initial pre-experiment briefing.

5.3 Limitations

We account for limitations of the given experiment and offer mitigation options for future experiments where appropriate.

Experiment Design. Whereas the experiment was balanced in that the participants of experiment and control group did manipulation tasks of similar structure that only differed in the depletion condition, the experiment was not designed to be double-blind. The experimenter knew which condition the participants were in. Future experiments can use a second experimenter for the tasks after the manipulation unaware of the depletion state.

Strength of Manipulation. The number of participants that reported strong depletion was low ($n = 11$), limiting the importance of this coefficient. Future experiments will need to achieve stronger depletion throughout and manipulate a slight cognitive effort stimulus deliberately. The cognitive depletion manipulation was only partially successful with 28 participants out of 50 reporting slight or strong agreement with tiredness. Earlier studies, such as [15] used 48 Stroop task items, while our study only contained 10 items. Future experiments can increase the cognitive effort by increasing the number Stroop task items.

Unequal Sample Sizes. Due to the differing impact of the manipulation on the participants, the grouping by depletion level yielded unequal sample sizes. This could have confounded analysis with One-way ANOVA. Consequently, we have employed an Univariate Analysis of Variance with Type III Sums of Squares robust in this situation. Alternative approaches with random re-sampling to groups with equal sample sizes agreed to the analysis outcomes.

Low granularity on depletion levels. Depletion levels have only been differentiated on a 5-point Likert-type scale. Further categorisation of these levels will be beneficial to investigate when cognitive effort promotes or inhibits security behaviour. Future experiments can mitigate this limitation with either a 9-point Likert-type scale or a Visual Analogue Scale (VAS).

Password measures. The password strength mea-

surements considered in this study all come with weaknesses: The password meter has limitations in being purely heuristic. The other two measures lack in differentiation across the range of password strengths. The NIST entropy estimate only offers a low differentiation between password strengths. The results of the CMU Password Guessability Service (PGS) are not normally distributed and come with a static cut-off at which the service considers a password "unguessable", conflating the strengths of secure passwords to a single value.

Low Adjusted R^2 in Linear Regression. The adjusted $R^2 = .206$, hence the automated linear regression accounts for 20.6% of the variability, adjusted for the number of predictors in the model. We observe that the variability in the experiment group as well as in the participants with a depletion level of effortful or depleted was higher than in the control group.

6 Conclusion

We offer the first comprehensive study of cognitive effort and depletion in a security context. We conclude that cognitive effort is a *necessary* condition for the creation of strong passwords, which in turn implies an involvement of *System 2* in terms of the dual-process model. It is an intriguing observation that slight cognitive effort improves password strength and that cognitive depletion diminishes password strength. It has far-reaching consequences for the design of password user interfaces and password policies. First, we observe that the user's cognitive effort and depletion may be more important than solely concentrating on password complexity requirements. Second, the user's cognitive depletion may yield an alternative explanation for and substantiate earlier research on the security compliance budget [4]. Third, our investigations indicate practical amendments to password policies ("Only set a new password when you feel fresh and awake.") and possible HCI interventions to strengthen password behavior (e.g., by inducing cognitive effort before the password generation).

Acknowledgments

We are grateful to Roy Maxion and Pam Briggs for the insightful discussions on this research. We are grateful for the insightful comments of the LASER reviewers, the LASER'16 workshop participants and of Chris Kanich. We thank Lorrie Cranor, Sean Segreti and Blase Ur for the support in the use of the CMU Password Guessability Service (PGS). This work was supported by the EU FP7 Project FutureID (GA nº318424) and the EPSRC Research Institute in Science of Cyber Security (RISCS; EP/K006568/1 ChAISe).

References

[1] ADAMS, A., AND SASSE, M. A. Users are not the enemy. *Communications of the ACM 42*, 12 (1999), 40–46.

[2] BAUMEISTER, R., BRATSLAVSKY, E., MURAVEN, E., AND TICE, D. Ego depletion: is the active self a limited resource? *Personality and social psychology 74* (1998), 1252–1265.

[3] BAUMEISTER, R. F., VOHS, K. D., AND TICE, D. M. The strength model of self-control. *Current directions in psychological science 16*, 6 (2007), 351–355.

[4] BEAUTEMENT, A., SASSE, M. A., AND WONHAM, M. The compliance budget: managing security behaviour in organisations. In *Proceedings of the 2008 workshop on New security paradigms* (2009), ACM, pp. 47–58.

[5] BLACKWELL, L. S., TRZESNIEWSKI, K. H., AND DWECK, C. S. Implicit theories of intelligence predict achievement across an adolescent transition: A longitudinal study and an intervention. *Child development 78*, 1 (2007), 246–263.

[6] BOWER, G. H. Mood congruity of social judgments. *Emotion and social judgments* (1991), 31–53.

[7] BURR, W. E., DODSON, D. F., AND POLK, W. T. Electronic authentication guideline. NIST Special Publication 800-63, NIST, jun 2004.

[8] CHROUSOS, G. P. Stressors, stress, and neuroendocrine integration of the adaptive response: the 1997 hans selye memorial lecture. *Annals of the New York Academy of Sciences 851*, 1 (1998), 311–335.

[9] DWECK, C. S. *Self-theories: Their role in motivation, personality, and development*. Psychology Press, 2000.

[10] FLORENCIO, D., AND HERLEY, C. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 657–666.

[11] GOLDBERG, L. R. An alternative" description of personality": the big-five factor structure. *Journal of personality and social psychology 59*, 6 (1990), 1216.

[12] GROSS, T., COOPAMOOTOO, K., AND AL-JABRI, A. Effect of cognitive depletion on password choice. Tech. Rep. TR-1496, Newcastle University, July 2016.

[13] HOONAKKER, P., BORNOE, N., AND CARAYON, P. Password authentication from a human factors perspective. In *Proc. Human Factors and Ergonomics Society Annual Meeting* (2009), vol. 53, SAGE Publications, pp. 459–463.

[14] JERMYN, I., MAYER, A. J., MONROSE, F., REITER, M. K., RUBIN, A. D., ET AL. The design and analysis of graphical passwords. In *Usenix Security* (1999).

[15] JOB, V., DWECK, C. S., AND WALTON, G. M. Ego depletion is it all in your head? implicit theories about willpower affect self-regulation. *Psychological science* (2010).

[16] JOHN, O. P., DONAHUE, E. M., AND KENTLE, R. L. The big five inventory – versions 4a and 54. Tech. rep., Berkeley, CA: University of California, Berkeley, Institute of Personality and Social Research, 1991.

[17] JOHN, O. P., AND SRIVASTAVA, S. The big five trait taxonomy: History, measurement, and theoretical perspectives. *Handbook of personality: Theory and research 2*, 1999 (1999), 102–138.

[18] KAHNEMAN, D. *Thinking fast and slow*. Farrar, Strauss, 2011.

[19] KELLEY, P. G., KOMANDURI, S., MAZUREK, M. L., SHAY, R., VIDAS, T., BAUER, L., CHRISTIN, N., CRANOR, L. F., AND LOPEZ, J. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *Security and Privacy (SP), 2012 IEEE Symposium on* (2012), IEEE, pp. 523–537.

[20] MAYER, J. D., AND GASCHKE, Y. N. The experience and meta-experience of mood. *Journal of personality and social psychology 55*, 1 (1988), 102.

[21] MURAVEN, M., TICE, D. M., AND BAUMEISTER, R. F. Self-control as a limited resource: Regulatory depletion patterns. *Journal of personality and social psychology 74*, 3 (1998), 774.

[22] SASSE, M. A., BROSTOFF, S., AND WEIRICH, D. Transforming the weakest link: a human/computer interaction approach to usable and effective security. *BT technology journal 19*, 3 (2001), 122–131.

[23] SCHWARZ, N., AND CLORE, G. L. How do i feel about it? the informative function of affective states. *Affect, cognition, and social behavior* (1988), 44–62.

[24] SHEPARD, R. N. Recognition memory for words, sentences, and pictures. *Journal of verbal Learning and verbal Behavior 6*, 1 (1967), 156–163.

[25] STROOP, J. R. Studies of interference in serial verbal reactions. *Journal of experimental psychology 18*, 6 (1935), 643.

[26] TICE, D. M., BAUMEISTER, R. F., SHMUELI, D., AND MURAVEN, M. Restoring the self: Positive affect helps improve self-regulation following ego depletion. *Journal of Experimental Social Psychology 43*, 3 (2007), 379–384.

[27] UEBELACKER, S., AND QUIEL, S. The social engineering personality framework. In *Socio-Technical Aspects in Security and Trust (STAST), 2014 Workshop on* (2014), IEEE, pp. 24–30.

[28] WEGNER, D. M., SCHNEIDER, D. J., CARTER, S. R., AND WHITE, T. L. Paradoxical effects of thought suppression. *Journal of personality and social psychology 53*, 1 (1987), 5.

[29] ZVIRAN, M., AND HAGA, W. J. A comparison of password techniques for multilevel authentication mechanisms. *The Computer Journal 36*, 3 (1993), 227–237.