An eye for deception: A case study in utilising the Human-As-A-Security-Sensor paradigm to detect zero-day semantic social engineering attacks

Ryan Heartfield*, George Loukas, Diane Gan
Department of Computing and
Information Systems
University of Greenwich, UK
*Email: r.j.heartfield@gre.ac.uk

***Abstract*—In a number of information security scenarios, human beings can be better than technical security measures at detecting threats. This is particularly the case when a threat is based on deception of the user rather than exploitation of a specific technical flaw, as is the case of spear-phishing, application spoofing, multimedia masquerading and other semantic social engineering attacks. Here, we put the concept of the human-as-a-security-sensor to the test with a first case study on a small number of participants subjected to different attacks in a controlled laboratory environment and provided with a mechanism to report these attacks if they spot them. A key challenge is to estimate the reliability of each report, which we address with a machine learning approach. For comparison, we evaluate the ability of known technical security countermeasures in detecting the same threats. This initial proof of concept study shows that the concept is viable.**

Keywords*—*Human-as-a-Sensor, social engineering, semantic attacks, cyber security.

I. Introduction

There is a growing realisation in the security industry that the users need to be at the core of any systems security design [1]–[6]. Our aim is to progress a step further and empower users to directly contribute to the security of themselves, their organisation or the wider community actively. For the detection of semantic social engineering attacks, users require an interface that provides them with functionality to report suspicious or anomalous activity that uses deceptive attack vectors rather than technical exploitations; for which the human user often is a more accurate sensor than an organisations technical security systems.

The aim of Human-as-a-Security-Sensor (HaaSS), and indeed most user-driven defences, is not to replace technical security systems, especially those that have been shown to work well in detecting and mitigating certain semantic attacks (e.g., phishing websites [7]), but to enhance or complement them by leveraging human sensing capacity and experience. More specifically, HaaSS can be used to actively augment existing technical defence mechanisms by combining telemetry generated by user threat detection with threats flagged by technical defence platforms; helping confirm the existence and highlight the extent of the threat, or crucially, for detecting semantic attacks that have been largely undetectable by technical systems. In this capacity, HaaSS allows for proactive and preemptive detection of semantic attacks by positioning (and empowering) the user as a platform security sensor in order to identify and report suspected attacks in real-time. To add clarity to the function of HaaSS in the context of semantic attacks and the wider computer security threat space, we propose the following definition:

Human-as-a-Security-Sensor. The paradigm of leveraging the ability of human users to act as sensors that can detect and report information security threats.

In this work we take the first steps towards exploring the applicability of the HaaSS concept for semantic attack detection within the context of active threat scanning (where sensors are purposely searching for the presence of threats), by testing the reliability of human users as security sensors in a laboratory-based experiment with *Cogni-Sense*; a proof of concept HaaSS platform.

II. Related Work

The *human as a sensor* paradigm has been deployed in a number of scenarios and contexts related the detection of physical threats, such as unfolding emergencies [8], or adverse physical conditions related to noise pollution [9]. The extension of this concept, which positions human computer users as physical sensors of cyber attacks, and in this specific case semantic attacks, is relatively new. Stembert et al. [10] have recently proposed combining a reporting function with blocking and warning of suspicious emails and the provision of educative tips, so as to harness the intelligence of expert and novice users in detecting email phishing attacks in a corporate environment. Initial experimental results of their mock-up have been encouraging for the applicability of the human as a security sensor concept in this context. Another recent example of utilising the concept was demonstrated by Malisa et al. [11] where the researchers developed an accurate and automated mobile application spoofing detection system by leveraging user visual similarity perception; integrating the human sensing data collected as an integral component of the technical systems detection decision making. To detect semantic attacks with some accuracy, both systems utilise explicitly user expertise and knowledge, but there is no exploration or measurement of what determines the users' performance as security sensors. By establishing such insight, a technical system could highlight the key attributes associated to user

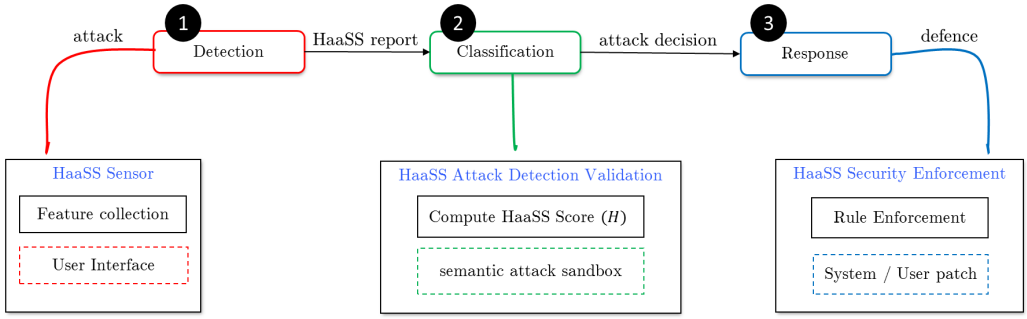


Fig. 1: Human-as-as-Security-Sensor defence model

threat detection and as a result improve system performance by recognising which threat reports are more credible.

An experimental study conducted by Heartfield et al. demonstrated that the reliability of a users attack reporting depends on their activity profile, as defined by characteristics, such as the amount and type of security training, familiarity with each system, frequency and duration of system access etc. [12], [13]. The profile also serves to define one's predicted susceptibility to semantic attacks. However, before building a system that depends extensively on a particular type of sensor (and the human sensor is no exception), one needs to be able to measure or estimate its overall reliability. In the case of HaaSS, this requires expanding upon theoretical observations made under survey or questionnaire conditions [14].

Here, we take the first steps in evaluating HaaSS performance by first measuring participants' detection efficacy profiles and then testing their ability to detect semantic attacks in an interactive laboratory environment using a specifically developed reporting mechanism. Furthermore, using the same attacks, we evaluate HaaSS comparatively with existing technical defence systems, which claim to provide technology to protect users against phishing and social engineering.

III. *COGNI-SENSE*: A PROTOTYPE HAASS SYSTEM

To leverage human user capacity for deception-based threat detection, it is first important to establish a systematic process for modelling HaaSS attack detection, classification and security enforcement. In Figure 1, we follow a conventional and well established technical defence approach [15] to describe a HaaSS defence model as a series of discrete system processes. In process 1, users form a HaaSS defence systems edge sensor detection mechanism, where threat exposure on different platforms user interfaces trigger attack detection and reporting. In this sensor component, HaaSS detection efficacy features are collected according to the user activity profile. Process 2 classifies received reports (as correctly detected attacks or not) and uses the HaaSS features to compute a HaaSS score which indicates the sensors expected reliability for the specific report. The HaaSS score is discussed in section III-A. In the classification process, if this score exceeds a defined threshold, the report is automatically assigned an action or sent to a sandbox for manual review. In both cases, after report classification, process 3 enforces a defence response which results in execution of security enforcing rules (e.g., blocking a website URL, or adding an email domain to spam lists) to deal with confirmed threats.

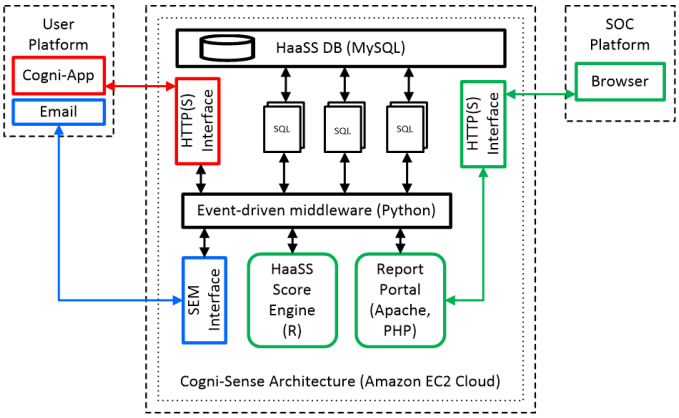


Fig. 2: Overview of *Cogni-Sense* technical architecture

ID	Plat.	Component	Description
1	Users	Cogni-app	Python app monitoring user platform activity and attack report interface with HTTP connectivity to the *Cogni-Sense* cloud platform
2	Cloud	*H* engine	R engine with Random Forest HaaSS Score prediction model producing a detection probability output for report classification
2	Cloud	Report portal	Apache PHP web server with MYSQL back-end for storing HaaSS reports and features, also provides a screenshot based sandbox for manual report review and classification
3	Both	SEM interface	Python interface to execute security rule enforcement on external defence platforms (currently integrated with SMTP server for sending attack awareness reports to HaaSS sensors)

TABLE I: Summary of major *Cogni-Sense* system components - ID refers to the associated process element as defined in Figure 1

Using the above process-driven model, we have developed a prototype HaaSS platform called *Cogni-Sense*. The prototype's technical architecture is shown in Figure 2, where

each of the coloured boxes within the architecture refers to a component's functional role within the overall HaaSS defence process, as shown in Figure 1. By combining each of the system processes into a technical system, *Cogni-Sense* provides a user with a practical facility to report suspected attacks, where credible reports are then utilised to implement tangible defences against identified threats. Table I summaries the technical components within *Cogni-Sense* with high-level description of their functionality. Within the system itself, detection, classification and security response are coordinated between Python middleware which provides communication between different components of the system, including a MySQL database which stores all HaaSS profile and report data, an R engine with the Random Forest HaaSS score model, an Apache PHP web-server which hosts the HaaSS report portal and sandbox and external security platform connectivity.

Feature	Variable Format
Familiarity with platform	Not very, somewhat, very
Frequency of platform use	Never, <1x a month, 1x month, weekly, daily
Duration of platform use	None, <30 min, 30 min-1h, 1-2h, 2-4h, >4h
Time since ST (platform)	Never, >1y, ≤1y, ≤6m, ≤3m, ≤1m, ≤2w
Familiarity with plat. type	Not very, somewhat, very
Freq. of plat. type use	Never, <1x a month, 1x month, weekly, daily
Dur. of platform type use	None, <30min, 30min-1h, 1-2h, 2-4h, >4h
Time since ST (plat. type)	Never, >1y, ≤1y, ≤6m, ≤3m, ≤1m, ≤2w
Time since ST (formal edu.)	Never, >1y, ≤1y, ≤6m, ≤3m, ≤1m, ≤2w
ST formal edu. (coursework)	No, Yes
Time since ST (at work)	Never, >1y, ≤1y, ≤6m, ≤3m, ≤1m, ≤2w
ST at work (videos)	No, Yes
ST Work-based (games)	No, Yes
Time since ST (self-study)	Never, >1y, ≤1y, ≤6m, ≤3m, ≤1m, ≤2w
ST self-study (websites)	No, Yes
ST self-study (videos)	No, Yes

ST = Infosec training, edu= education, y = year, m = month, w = week

TABLE II: Random Forest Susceptibility model HaaSS features used to compute HaaSS Score H

A. Predicting Sensor Reliability: The HaaSS Score

To determine the credibility of semantic attack reports, *Cogni-Sense* utilises a semantic attack susceptibility model, based on a Random Forest machine learning algorithm, which has been developed and trained from experiments by Heartfield et al. in [12], which identified key features associated to susceptibility to semantic attacks. This susceptibility model has been directly integrated into the technical implementation of the HaaSS report classification process in *Cogni-Sense*. The susceptibility model facilitates classification of HaaSS report detection accuracy (to confirm a report as an attack or not) by analysing users' computer usage and activity profiles in order to predict their expected detection efficacy when reporting suspected attacks on specific platforms. Here, the model's specific features, which were identified as indicators of semantic attack susceptibility (and therefore attack detection efficacy), are summarised in Table II. Random Forest is a machine learning algorithm that functions as an ensemble of decision trees which evaluates the class association (here attack report *Vs.* not an attack report) of a set of random variables in a feature-set. The outcome of a trained Random Forest is a set of n decisions trees with fixed decision points that have been selected as part of the training process, where if a particular pattern of feature value inputs are observed will produce a

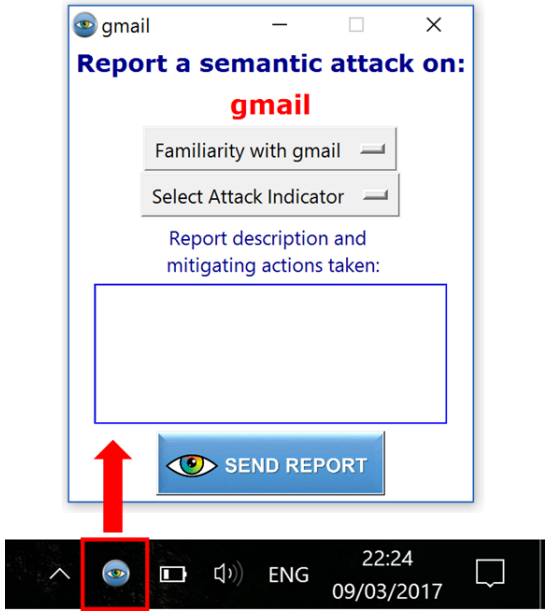


Fig. 3: The *Cogni-Sense* HaaSS reporting app icon running in system tray. When clicked, a reporting window is opened with the detected platform and report information

prediction outcome that follows these fixed decision points. The model can operate in two modes, strict classification or class probability. As strict classification is sensitive to false positive and false negative output, which in practice would result in the discarding of accurate reports or nugatory time spent reviewing non-attack reports, *Cogni-Sense* utilises a class probability mode output for prediction user detection efficacy; producing a probability metric of instead of direct classification (e.g., 0 or 1). Using this approach, the probability metric provides a key facility to prioritise user reports which are most likely to be credible semantic social engineering attacks. We refer to this probability metric as the HaaSS (H) score. This score serves to inform the *Cogni-Sense* system of a HaaSS report's reliability based on the reporting user. Moreover, this metric is also intended to aid the optimisation of resources for investigating and mitigating credible threats (either automatically based on a trusted H score threshold or via a security operations engineer), with the aim to reduce the effective time period of semantic attack exposure and potential exploitation for system users. Here, prioritisation assumes that the higher the score (i.e., probability) the more likely that a report contains a correctly identified attack.

In the following experiment, we evaluate two major components of *Cogni-Sense*: (a) the HaaSS sensor semantic attack detection reporting mechanism *Cogni-app*, installed on the participant experiment environment, as shown in Figure 3, and (b) the viability of the HaaSS score prediction as a utility to determine accurately HaaSS attack detection efficacy compared with a range of technical security platforms which claim to enforce anti-social engineering defences against different semantic attacks. Due to the role-play constraints of the remote, laboratory-based experiment environment, participants' HaaSS features were imported manually in the *Cogni-Sense* system. In this experiment, we did not use or evaluate the

ID	Security Platform	Platform Type	Phishing	Web Rating	URL blocking	Heuristics	On-access malware
E1	Yahoo Mail	Email	✓	✗	✓	✗	✓
E2	Gmail	Email	✓	✗	✓	✗	✓
E3	Outlook	Email	✓	✗	✓	✗	✓
E4	ProtonMail	Email	✓	✗	✓	✗	✓
E5	Yandex	Email	✓	✗	✓	✗	✓
E6	GMX	Email	✓	✗	✓	✗	✓
E7	mail.com	Email	✓	✗	✓	✗	✓
B1	Firefox	Browser	✓	✓	✓	✗	✓
B2	Chrome	Browser	✓	✓	✓	✗	✓
B3	Opera	Browser	✓	✓	✓	✗	✓
B4	Commodo Dragon	Browser	✓	✓	✓	✗	✓
B5	Avast Safezone	Browser	✓	✓	✓	✗	✓
B6	Microsoft Edge	Browser	✓	✓	✓	✗	✓
B7	Safari	Browser	✓	✓	✓	✗	✓
A1	Commodo Cloud	AV	✓	✗	✓	✓	✓
A2	AVG AntiVirus	AV	✓	✗	✓	✓	✓
A3	Avast AntiVirus	AV	✓	✗	✓	✗	✓
A4	Windows Defender	AV	✓	✗	✓	✓	✓
A5	Norton Security	AV	✓	✓	✓	✓	✓
A6	Kaspersky Internet Security	AV	✓	✓	✓	✓	✓
A7	Sophos Intercept X	AV	✓	✓	✓	✓	✓
P1	Facebook	Platform	✓	✗	✓	✗	✗
P2	GoogleDrive	Platform	✓	✗	✗	✗	✓
P3	Windows10	OS	✗	✗	✗	✗	✗

TABLE III: Technical security platforms with built-in anti-phishing, anti-malware functionality tested against laboratory semantic social engineering attacks

automatic HaaSS feature collection functionality developed in *Cogni-Sense*. This is because participants spent an average of thirty minutes attempting to detect attacks, where insufficient learning time was available to collect HaaSS feature data automatically. We do, however, evaluate the functionality of security enforcement module (SEM) integration attack classification e-mail alerting (shown in Figure 6), which was triggered by conducting manual classification on a semantic attack report received by HaaSS sensor 5 (H5 - experiment ID 15) during the experiment. Participants did not have access to the cloud-based *Cogni-Sense* portal during or after the experiment and were unaware of reports made by other participants during the experiment.

IV. HaaSS Vs. Technical defences for detection of semantic attacks: the case of Selina Carlysle

In the following experiment, we directly compare the detection capabilities of a group of participants acting as HaaSS sensors against a range of technical defences which claim to enforce anti-social engineering technologies against a range of different semantic attacks.

A. Experimental environment

The experimental environment was presented in the form of a Windows 10 virtual machine which each participant could remotely access via *TeamViewer*. The task was a role-play exercise in the form of "a day in the life of Selina Carlyle" (an imaginary freelance artist), where all participants conducted a number of computer-based activities that Selina would typically carry out as part of her computer usage. This involved checking her email on Gmail, accessing Facebook and reading messages, notifications, as well accessing other platforms such as Twitter, Pinterest and general web browsing for artwork. In the case of Twitter, Pinterest and web browsing, these were designed as noise activities to prevent participants from presuming that all attacks would reside within Gmail or Facebook.

In total, seven participants were recruited for the experiment as HaaSS sensors by inviting a number of computer science students, lecturers and the general public to complete a questionnaire related to the experiment's purpose and their computer activity profile. The questionnaire described the role-play scenario, the goal of reporting detected attacks using the *Cogni-Sense* app, and collected offline the required HaaSS features for computing the H score of each participant with the Random Forest susceptibility model. Each of the participants were assigned a HaaSS sensor number and reporting user ID to match reports to corresponding HaaSS sensors in the experiment. Participants were given also a user guide on how to use the *Cogni-Sense* reporting tool (Figure 3) when detecting a suspected semantic social engineering attack. The same Windows 10 virtual machine environment was also used to install and test individually each of the technical defence platforms against each of the semantic attacks. In table III, each of the technical platforms and defence systems listed is evaluated according to their individual functional capabilities for detecting semantic attacks. Whilst email and browser platforms tend to offer anti-phishing, URL filtering and anti-malware defence, they do not directly employ heuristic scanning as part of this functionality, which as shown, is exclusively provided by the anti-virus software that we have evaluated. This means that in practice, most email providers rely on signature-based attack recognition for email by query through registered attack databases.

B. Zero-day semantic social engineering attacks

The vast majority of semantic social engineering attacks are largely undetectable by technical defence systems, because they primarily rely on cosmetic or behavioural deception vectors and as a result often leave very small technical footprint that can be analysed, especially if the deception has been designed to utilise intended user functionality [16]. Consequently, technical heuristic detection capabilities have a limited view of potential attack vectors through user actions, instead of system interfacing malware. In most cases, technical

Attack	Emulated Attack	Depend.	Description
1.1	Spear Phishing Email	-	Targeted participant email advertising job role specific to their profile from fake recruitment company with URL to purported job description PDF document on Google drive
1.2	Cloud Storage File Masquerading	1.1	Malware HTA file masquerading as PDF in online Google Drive folder
2.1	IM Phishing	-	Unsolicited Facebook message containing Facebook page link
2.2	Multimedia masquerading	2.1	Malicious image link masquerading as Facebook video post
3.1	Phishing Email	-	Order confirmation email from Amazon with order details and tracking URLs leading to phishing Amazon login web page
3.2	Phishing website	3.1	Amazon login phishing website which captures user login details

TABLE IV: Experiment emulated semantic attacks sent to participants with indicated date and time at which the attacks were launched for all participants (this does not guarantee that participants were exposed to the attacks at the time of launch)

defence systems rely on attack reports before they can develop signatures that can be matched against similar patterns when analysing potential threats, or attempting to pre-empt them.

For example, it is difficult to characterise a website as phishing if the URL is not registered with a spam database, and does not use obvious tricks such as similar domains names used as sub-domains, obfuscated by domain suffixes which are not related to the masqueraded website (e.g., amazon.net-shopping.tk). In cases where a phishing website name originates from a legitimate and credible service provider (and does not attempt to obfuscate its appearance), until the website has been reported as malicious (or contains easily identifiable malicious code or web re-directions in the web page), most technical defence platforms will not recognise the website as phishing. The same example can be seen in spam emails where spam protection mechanisms analyse components such as sender from and to address, subject title, domain, hyperlinks, attachments, salutation and common phrases (e.g., urgency) as to match known common patterns phishing attacks. However, if the email body consisted purely of a deceptive image from a domain name not in a black list, then the classifiers effectiveness is significantly reduced, as a spam protection is unlikely to interpret the visual information in the image.

For technical defence systems to stand a chance in detecting unknown semantic attacks, defence mechanisms require the ability to interpret visual and behavioural attributes in real-time to predict the likeliness that a deception attempt is occurring - which without knowing the user or integration with the platform would likely result in many false-positives or false-negatives. On the other-hand, users, by definition, are implicitly interfaced with such attributes and are therefore best placed to decide whether system activity on the user interface is anomalous or not, based on their experience and knowledge. As each of the attacks exposed to users and technical defences were developed specifically for this experiment, and therefore have not been seen by technical defence systems or users before, they are assumed to be zero-day semantic social engineering attacks in this case.

In this study, we aim to evaluate the detection efficacy of HaaSS sensors for identifying zero-day semantic social engineering attacks over technical defence systems, in order to show the potential usefulness of the HaaSS concept for detecting deception-based threats. Each of the attacks in the experiment are described in Table IV.

C. Experimental Results

In Table V, we compare the experimental results for the HaaSS participants and each of the technical platforms for detecting the semantic attacks in Table IV. The spear-phishing email in attack 1.1 proved the most challenging one to detect for HaaSS participants, with only three out of seven participants correctly reporting the email as an attack. However, three out of four of the HaaSS sensors who were exploited by the email by clicking on the GoogleDrive link, then detected the malicious HTML application file in Google drive afterwards (attack 2.2). The H score prediction at a probability threshold of 50% threshold was only 43% accurate, but at a 65% threshold, it was 86% accurate, with a 100% true positive rate and false positive rate of 25%. By comparison only two out of the seven email providers, Yandex and Yahoo mail, sent the email to spam, with all others placing the email in the inbox folder with no warnings of a suspected attack. With the exception of Firefox (B1) and Windows 10 (P3), which prompted the user that they were downloading/running an executable file; Commodo Cloud AV (A1) also ran the downloaded file in a sandbox as a default action of the software because it was an unknown file. All other browsers and antivirus packages failed to detect the email or subsequent file in GoogleDrive (in the online platform or when downloaded and run) as malicious; with no user warnings at all.

For the IM phishing message (attack 2.1) on Facebook only three out of seven HaaSS sensors were actually exposed to the semantic attack, as the message originated from an account which was not a Facebook friend of the Selina persona. This meant that the message was placed under Facebook "message requests" which is more hidden than friend based messages or profile notifications. For the three HaaSS sensors that were exposed, all three clicked on the link in the message, which was included as an image hyperlink leading to another Facebook page. At this point during the technical testing, no technical defences (including the Facebook platform itself) had flagged the message as malicious or anomalous. Even though all three HaaSS sensors were exposed to the video masquerading images on the fake Facebook charity page (attack 2.2), none of the users actually clicked on the image, but they also failed to report it as a semantic attack. Again, none of the technical defences and the Facebook platform itself detected

	HaaSS							Email Prov.							Browsers							AntiVirus							Platform		
Attack	**H1**	**H2**	**H3**	**H4**	**H5**	**H6**	**H7**	**E1**	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**	**B7**	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**P1**	**P2**	**P3**
1.1	.68	**.92**	.61	.77	.64	.61	.78	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	-	-	✗
1.2	.66	**.87**	.49	.19	**.77**	.52	.76	-	-	-	-	-	-	-	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	-	✗	✓
2.1	.78	**.85**	.64	.90	.74	.23	.84	-	-	-	-	-	-	-	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	-	✗
2.2	.78	**.85**	.64	.90	.74	.23	.84	-	-	-	-	-	-	-	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	-	✗
3.1	.68	**.92**	.61	**.77**	.64	.61	.78	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	-	-	✗
3.2	.68	**.79**	.53	.46	.54	.57	.75	-	-	-	-	-	-	-	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	-	-	✗

TABLE V: HaaSS and Technical experiment attack detection results. The number refers to the HaaSS score H, (e.g., probability of detection). The colour refers to detection result: red - not detected, green - detected, orange - precautionary measure taken, but no threat reported, grey - attack not seen.

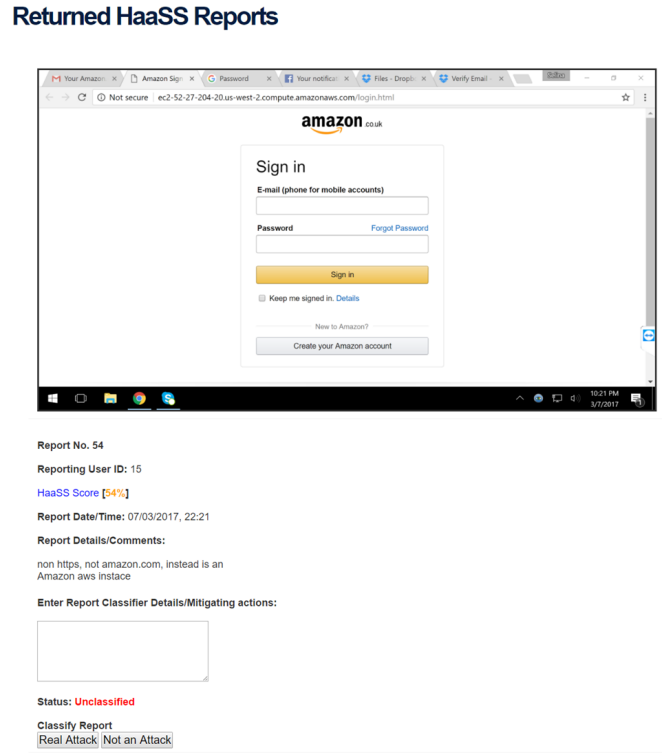


Fig. 4: Attack 3.2 (Amazon login phishing website) *Cogni-Sense* portal report screenshot for HaaSS report by H5

the image as malicious (which was expected given the fact that the image was simply using the in-built Facebook image hyperlink functionality and posing as a video using cosmetic features in the image). Here, the H score predicted that the three users exposed to the threat would detect it, however given the lack of exposure to the threat in a realistic context (e.g., participants' own personal accounts), it is unclear whether the actual response is robust enough to discount the prediction of the H score.

For the Amazon phishing email (attack 3.1), six out of seven HaaSS sensors detected the attack, whereas only Yandex email provider sent the email to spam; Yahoo email blocked the email's images but did not flag the email as malicious. All other technical defences failed to identify the email as a threat or provide any warnings. The HaaSS score correctly predicted six out of seven HaaSS detections at the 50% threshold, but would only have detected four out of seven correctly at the 65% threshold. The one HaaSS sensor that was exploited by clicking on the link in the Amazon phishing email, then correctly detected the following Amazon phishing login webpage - an action correctly predicted by the participant's HaaSS score. Again, all technical defences and platforms failed to detect the phishing website as malicious.

Overall, the HaaSS sensors were more efficient at detecting all threats than the technical defences exposed to the semantic attacks - without prior knowledge of the attacks themselves or any training provided prior to this experiment. By comparison, the technical defences in almost all cases failed to detect the existence of a threat. Surprisingly, Yahoo detected the spear phishing email as spam, but not the Amazon email which used a header alias to look as if the email originated from Amazon. An example a HaaSS detection result for attack 3.1 is shown in Figure 4 and 5, which is the screen capture of the users screen made by the *Cogni-Sense* reporting app (which is sent to the cloud portal) and the accompanying report details showing the computed HaaSS score for the report, time and date when the report was made and the HaaSS sensor report observation details. From a classification perspective, HaaSS was 68% accurate at a 50% probability threshold with a true positive rate of 93% and true negative rate of 43%, precision of 62%, false positive rate of 57% and false negative rate of 7%. However, at the 65% probability threshold, it achieved an accuracy of 64% with a true positive rate of 67%, true negative rate of 6%, precision of 67%, false positive rate of 4% and false negative rate of 33%. Where the HaaSS score was shown to be the same value for consecutive attacks on the same platform (e.g., email on Gmail, social media on Facebook), this is due to the report and classification processes occurring in essentially the same time period, with the same feature set.

In the case of organisational HaaSS defence, if these participants were HaaSS sensors in a security platform, prioritising these reports based on the HaaSS score probability results in the experiment would ensure almost all of the attacks would be identified before reviewing a non-attack report. Furthermore, by utilising such HaaSS sensors, an organisations security platform would indeed detect the semantic attacks in the first place, which, as we have demonstrated, is unlikely to be the case if they were to rely purely on the types of technical defences evaluated in this experiment.

D. Limitations

In our laboratory-based experiment, there are a few limitations that must be considered. Overall, the experiments were taken in a very controlled environment which could have an effect on the detection efficacy of HaaSS sensors compared to their performance in a more realistic scenario. For example, participants were primed to the purpose of the role-play experiment, and such may have been more vigilant and

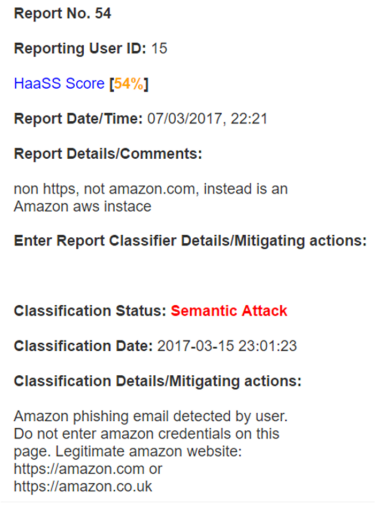


Fig. 5: Attack 3.2 (Amazon login phishing website) *Cogni-Sense* portal report description for HaaSS report by H5, classified as a semantic attack

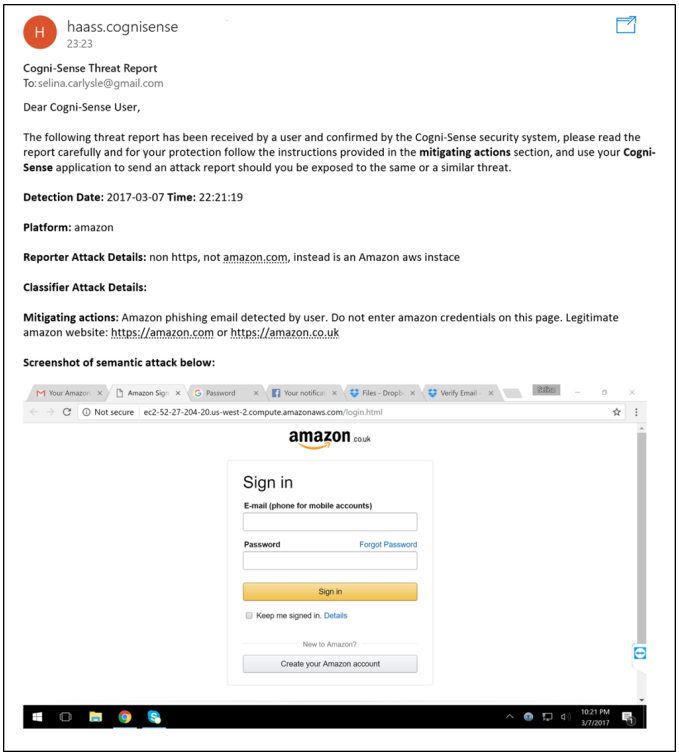
sensitive to each of the deception vectors than they would have normally been; and this may have weakened an attack's effect. Conversely, as the experiment involved role-play, this may have also reduced participants' ability to determine whether certain attack interactions were contextually anomalous (e.g., Amazon order confirmation). Nevertheless, in direct comparison with the operation of technical defence systems, a HaaSS sensor in practice can be purposely employed solely to search for the existence of semantic social engineering attacks in a continuous "online" fashion as part of a security platform within an organisation (which is analogous to a security operations centre continuously monitoring a network traffic security feed for anomalies) or remote paid service (e.g., cloud-based HaaSS reporting). From this perspective, it would be assumed that a HaaSS sensor is constantly vigilant and therefore actively searching for semantic attacks on platforms that are accessed.

In respect to the experiment itself, limitations were also shown in the evaluation of HaaSS against semantic attacks 2.1 and 2.2 (Facebook IM phishing and video masquerading), due to participants not being exposed to the attack or simply missing it. In practice, it is still unclear how effective such attacks would be, and therefore it is important to investigate this more robustly in future work.

V. Conclusion

In this work, we have put the HaaSS paradigm to the test with a first case study in semantic social engineering attacks, and compared against technical platforms that claim to provide defence against such attacks as well as technical defence systems designed to protect specifically from them. In this respect, this first evaluation was successful, as the users performed considerably better than all technical defence systems, and the *Cogni-Sense* application developed for leveraging this ability of users proved fit for the purpose. One of its most important contributions is that it allows not only to capture HaaSS reports but also to score them according to the

Fig. 6: *Cogni-Sense* HaaSS report semantic attack classification for report by H5, triggering SEM module rule: attack awareness email security enforcing function rule



estimated reliability of the users that generated them. In future work, we will evaluate *Cogni-Sense* in a real-world and more extensive experiment, featuring automated collection of user activity features, as well as automated security enforcement based on the reports' HaaSS scores. Furthermore, in future development *Cogni-Sense* will be expanded to mobile devices and the Internet of Things in order to utilise HaaSS holistically across a wide range of future user-computer interfaces.

References

[1] University of Oxford, "Information security - report an incident," 2016. [Online]. Available: https://www.infosec.ox.ac.uk/report-incident

[2] BBC, "Fake news: Facebook rolls out new tools to tackle false stories," 2016. [Online]. Available: http://www.bbc.co.uk/news/world-us-canada-38336212

[3] PhishMe, "Phishme reporter," 2016. [Online]. Available: https://phishme.com/product-services/reporter

[4] Wombat Security, "Wombat security announces new feature to reinforce secure employee behavior against phishing," 2016. [Online]. Available: https://www.wombatsecurity.com/press-releases/phishalarm-email-add-in

[5] Sophos, "Sophos phish threat," 2017. [Online]. Available: https://www.sophos.com/products/phish-threat.aspx

[6] M. A. Sasse, C. C. Palmer, M. Jakobsson, S. Consolvo, R. Wash, and L. J. Camp, "Helping you protect you," *IEEE Security and Privacy*, vol. 12, no. 1, pp. 39–42, 2014.

[7] Webroot, "Webroot real-time anti-phishing service," 2013. [Online]. Available: http://www.webroot.com/shared/pdf/WAP-Anti-Phishing-102013.pdf

[8] M. Avvenuti, M. G. Cimino, S. Cresci, A. Marchetti, and M. Tesconi, "A framework for detecting unfolding emergencies using humans as sensors," *SpringerPlus*, vol. 5, no. 1, pp. 1–23, 2016.

[9] Y. Zheng, T. Liu, Y. Wang, Y. Zhu, Y. Liu, and E. Chang, "Diagnosing new york city's noises with ubiquitous data," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Sep. 2014, pp. 715–725.

[10] N. Stembert, A. Padmos, S. M. Bargh, S. Choenni, and F. Jansen, "A study of preventing email (spear) phishing by enabling human intelligence," in *Intelligence and Security Informatics Conference (EISIC)*. IEEE, 2015, pp. 113–120.

[11] L. Malisa, K. Kostiainen, and S. Capkun, "Detecting mobile application spoofing attacks by leveraging user visual similarity perception," *IACR Cryptology ePrint Archive*, 2015.

[12] R. Heartfield, G. Loukas, and D. Gan, "You are probably not the weakest link: Towards practical prediction of susceptibility to semantic social engineering attacks," *IEEE Access*, vol. 4, pp. 6910–6928, 2016.

[13] R. Heartfield and G. Loukas, "Evaluating the reliability of users as human sensors of social media security threats," in *International Conference On Cyber Situational Awareness, Data Analytics And Assessment (CyberSA)*. IEEE, 2016, pp. 1–7.

[14] M. A. Sasse, M. Smith, C. Herley, H. Lipford, and K. Vaniea, "Debunking security-usability tradeoff myths," *IEEE Security and Privacy*, vol. 14, no. 5, pp. 33–39, 2016.

[15] Bluecoat, "Malware analysis and sandboxing," 2017. [Online]. Available: https://www.bluecoat.com/en-gb/products-and-solutions/malware-analysis

[16] R. Heartfield and G. Loukas, "A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks," *ACM Computing Surveys*, vol. 48, no. 3, 2016.