SNIPPET: Genuine Knowledge-Based Authentication

Karen Renaud[1], Demetris Kennes
School of Computing Science
University of Glasgow
[1]Email: karen.renaud@glasgow.ac.uk

Johan van Niekerk
School of ICT
Nelson Mandela Metropolitan University

Joe Maguire
School of Computing Science
University of Glasgow

***Abstract*—Authentication is traditionally performed based on what you *know*, what you *hold* or what you *are*. The first is the most popular, in the form of the password. This is often referred to as "knowledge-based" authentication. Yet, given the guidelines for password restrictions commonly given to end-users we will argue that this is a misnomer. A strong password is actually a lengthy string of gibberish or nonsense. Common password strength guidelines advise users against choosing meaningful passwords.**

Humans are not very good at remembering nonsense strings so they very reasonably choose meaningful passwords which are easily guessed. This appears to constitute a stand off between the mnemonic needs of end users and the security needs of the system. If we could find a way of reducing the mnemonic load on users they might well be more likely to choose stronger authentication secrets. We could, for example, rely on pre-existing knowledge rather than requiring users to memorise a random alphanumeric string. If we were able to do this it should be easier for them to respond, and also harder for a random intruder to replicate the knowledge.

Testing knowledge directly is probably infeasible in an authentication setting. We will show that experts can identify what they themselves produce as they go about carrying out their own skilled activities.

We trialled a prototype mechanism which tested the memorability, observability and guessability of an authentication mechanism that relied on an expert programmer identifying his/her own code snippets. We conducted a pilot study and report on our findings. These findings are not conclusive given the small number of participants but they do show promise and suggest that this is an area worth pursuing.

***Index Terms*—Authentication, Experts, Knowledge**

I. Introduction

Users increasingly consume services and purchase goods online. Online vendors rely almost ubiquitously on alphanumeric passwords to allow such users to verify their identity, and thence to authorise purchases. Consequently, passwords are being pushed into the hands of a much wider variety of users and being used in environments never envisioned by the people who first decided that the ancient password[1] would be a useful concept in the digital world to confirm identity.

Users generally *identify* themselves online by means of an alphanumeric string, usually an email address. They then need to *authenticate*, in order to prove that they have a right to claim that identity. Authentication is traditionally performed in one of three ways, as explained by the National Institute of Standards and Technology [2] by using: something you *know*, something you *hold* or something you *are (or do)*. The most popular of each of the three categories is the password, the smart card and the fingerprint, respectively.

Pankati [3] predicted at the turn of the century that biometric-based authentication was the future. He argued that since tokens were easily misplaced and it was easy to forget passwords, the only future direction for authentication was the dependable and indisputable biometric [3].

However, despite the intervening thirteen years, biometric-based authentication remains relatively novel and passwords not only persist, they reign supreme, as the *de facto* authentication approach across the globe. In effect, passwords have become the default authentication solution for every context and user. One could argue that this is not necessarily a problem.

Fernado J. Corbató, the project leader behind one of the first systems to use passwords, Compatible Time Sharing System (CTSS) [4], explained that although passwords seemed theoretically strong, in practice many problems emerged. People routinely compromise security by choosing weak passwords [5], and by writing them down and sharing them [6]. A lot of this behaviour is driven by the fact that they have too many passwords [7], [8], and because they have previously forgotten passwords and have no desire to repeat the experience. Weidenbeck argues:

> **"A better way to overcome the password problem is to develop password systems that reduce fundamental memory problems"** [9, p. 105]

The rest of the paper is structured as follows. Section II explores the concept of "Something you Know" authentication. Section III explores the idea of a skill-based authentication, leading to the concept of "what the expert did" authentication. Section IV reports on a survey of programmers to determine whether they thought they would be able to identify their own and others' programming code. The survey results suggested that this method might indeed work, Section V reports on a pilot study we carried out to test a "what the expert produced" authentication mechanism. We report on our results in Section VI. Section VII concludes.

[1]The Tale of Ali Baba and the Forty Thieves in The 1001 Arabian Nights and The Bible both mention password use

II. "Something you Know" Authentication

"Something you know" authentication is the process of confirming a claimed identity through knowledge of a secret, one known only to you and the other party. Since it is a secret, individuals are advised to memorise it and not to record or share it. The secret itself could be a public event or record, but the use thereof must not be revealed.

The alphanumeric password is the best known implementation of "what you know" authentication. There are two reasons for this:

1) the concept of passwords is one which is centuries old and is easily understood by both users and developers; and
2) the interaction mechanism, i.e. keyboard, is over a century old and one can easily enter passwords without additional training or expense.

This made passwords the authentication mechanism of choice for early systems, such as CTSS [4], and operating system designers such as Ken Thompson and Dennis Ritchie.

The problems with passwords emerged soon after their initial deployment. They immediately proved difficult to use and remember [10]. The situation has barely improved as technology has advanced. If anything, as the world becomes increasingly connected, the ubiquitous use of passwords becomes even more problematical. News stories detailing the problems caused by the improper use of passwords are not a rare occurrence. The Federal Trade Commission, for example, has recently taken legal action against Wyndham Hotels after the organisation failed to properly protect the financial information of 500,000 customers, resulting in damages of $10.6 million [11]. The organisation generated weak and simple passwords that were compromised by attackers and allowed them to install software to capture information.

The use of simple passwords is not particularly surprising as users will create simple passwords to avoid the inconvenience of not be able to complete a task, since they have probably forgotten a password previously and do not want to repeat the experience [12]. The following excerpt, extracted from a complaint submitted by the Federal Trade Commission, offers evidence of the use of simple passwords in the aforementioned case, as follows:

> "For example, to allow remote access to a hotel's property management system, which was developed by software developer Micros Systems, Inc,. Defendants used the phrase "micros" as both the user ID and password"
>
> **Federal Trade Commission Compliant [11, p. 11]**

The use of such simple strings for the convenience of a few individuals led to dramatic inconvenience for 500,000 paying guests. A great deal of expense, in terms of time and money, was spent rectifying the problems caused by this irresponsible authorisation mechanism.

However, passwords that are difficult to remember also incur costs for organisations. The estimated cost of password bureaucracy, such as replacement and recovery, is an estimated $17 per call [13]. Moreover, an estimated 30% of call volumes are associated with passwords [13]. Consequently, not only is there a cost associated with each call, there are also a considerable number of calls to cope with.

Despite these problems, the vast majority of authentication in 2013 falls into the "something you know" category. This is often termed *knowledge-based authentication*, which seems intuitively correct. Here we argue that this is, in fact, misguided. To make this argument we need to examine the distinctions between data, information and knowledge.

- *Data:* Data is simply data: no use to anyone until someone provides the context. So, for example, consider the number: 2.5, a simple piece of data. There is no way of knowing what that number refers to.
- *Information:* If we add context and explain that this is the number used to convert a measurement from inches to centimetres, the data has become information, because it now has meaning. It is not yet knowledge, however.
- *Knowledge:* Knowledge is defined by the Oxford dictionary as: "the theoretical or practical understanding of a subject". In other words, knowledge implies an understanding of how to use the information to solve some problem. If one is given the dimensions of a room in inches and asked to calculate the area of the room in cm^2, the information just provided would be applied in order to solve the problem. The person would also have to know how to work out area using the width and breadth and know how to multiply the dimensions by the conversion value to arrive at the correct result. This implies an understanding of how to use the information, and success suggests that you do indeed possess that knowledge.

Knowledge and skills take time to develop, and this process cannot be short-circuited [14]. The benefit is that knowledge and skills are not easily disrupted. The nature of the knowledge and skill acquisition process seems to make a durable footprint on the user's mind that does not easily decay, even with age, especially when learnt before retirement [15]. Moreover, retrieving the knowledge requires less effort than recalling a nonsense data string effortfully memorised and possibly forgotten. Nonsense is forgotten because the brain is economical and performs neural pruning on networks that are not deemed essential [1]. The more interesting and stimulating something is, the more easily it will be remembered. Nonsense is neither stimulating nor interesting, and is deliberately pruned.

It is also of interest to note that the above mentioned "levels" as one progresses from data to knowledge also, to a certain extent, map to the first three levels of Bloom's well-known taxonomy of the cognitive domain [16]. The following lists the first three levels as presented by [16], and briefly shows how these levels relate to the distinction between data, information, and knowledge.

- *Remember:* This is the lowest level of cognition. Remember is the ability to *retrieve* relevant facts from memory but does not include the ability to relate the retrieved facts

to a specific context.

- *Understand:* If we add context to remembered data a person has the ability to understand the data, "construct the meaning of instructional messages" [16, pp. 30], but does not necessarily have the ability to apply it correctly.
- *Apply:* The third level of the cognitive domain is being able use the information correctly in a given situation or context. This level of cognition thus clearly requires the person to have *knowledge*, as defined above.

Now consider authentication. Here is some advice given by CERT[2] for choosing a password:

- Don't use passwords that are based on personal information that can be easily accessed or guessed.
- Don't use words that can be found in any dictionary of any language.
- Develop a mnemonic for remembering complex passwords.
- Use both lowercase and capital letters.
- Use a combination of letters, numbers, and special characters.
- Use passphrases when you can.
- Use different passwords on different systems.

A password chosen according to these guidelines is more akin to data than it is to knowledge. If a password has meaning, it has become information. If it is information then attacks become easier to carry out. Users use information instead of data as passwords so that the password will not be forgotten, Such an information-based password has meaning, usually something related to the user him or herself. This action potentially weakens the password since an attacker who knows the user will be more likely to be able to guess it. Figure 1 shows how the drive for strong passwords conflicts with users' motivation to choose memorable and meaningful passwords.

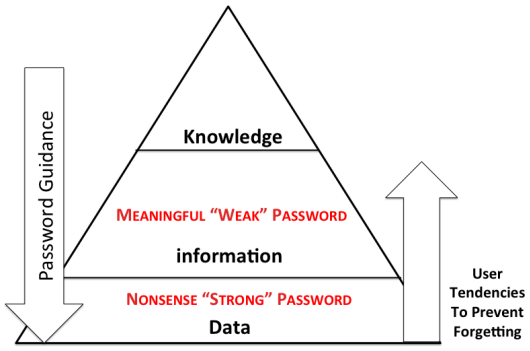

Fig. 1. Passwords Positioned within the DIKW Pyramid

Thus a more realistic moniker for "something you know" authentication would be "some nonsense" authentication. This begs the question: what would knowledge-based authentication actually look like?

There are clear challenges inherent in testing genuine knowledge in this context. Skills are usually tested by asking someone to apply their knowledge in context. We cannot do this at each authentication attempt since it would be too time-consuming. The following section explores the issue of testing knowledge and skills in an authentication setting.

III. Moving up the Pyramid

Generally, one can test "what you know" in one of three ways in authentication: *recall, cued-recall* and *recognition* Cued-recall mechanisms provide some cues which help the user to recall their authentication secret. An example is the Cueblot mechanism [17] which displayed an inkblot-like image to trigger the user's memory when they had to authenticate. Since the cueblot was sufficiently abstract it did not act as a cue for other users, but only for the legitimate user.

Another example of a cued-recall mechanism is Zviran's associative passwords which probe a user's personal experience [18]. This quiz-based approach extracts several pieces of knowledge from the user. The individual is presented a series of *fact-based* and *opinion-based* questions. A fact-based question would be 'What was the first school you attended?', while an opinion-based question would be 'What is your favourite film?'.

Recognition-based mechanisms most often display grids of images and require the user to click on their own image from the challenge set. A number of these have been proposed [19], [20], [21], but Déjà Vu [22] was one of the first. Recognising is easiest for users, since all they have to do is click on their own secret image in order to authenticate: it is cognitively the least demanding mechanism.

A. Acknowledging Individual Differences

Authentication mechanisms tend to treat users as if they were a homogeneous group, making no effort to personalise authentication for individual tastes. They might allow users to choose their own secrets, but they do not offer different dictionaries to different users, being concerned that this will make it easier for attackers to guess the secrets, Yet that seems to ignore, and fails to exploit, the diversity of these same users. Humans are unique and respond and react to the world around them in a way that reflects this.

Most recognition-based authentication mechanisms do not personalise the images used by the mechanism, using the same images for the entire user population. Unfortunately, when they are allowed to choose from a common dictionary their choices are predictable [19], [23]. Perhaps they are still trying to find meaning in their secrets so as to prevent the secret from being forgotten.

Another way of considering this drive for meaning is that users are *personalising* their secrets. They do this when they choose weak passwords and they do this when they make predictable image choices for their recognition-based authentication mechanisms. Is it possible to personalise the secrets, increasing memorability, while maintaining a desirable level of non-predictability?

[2]http://www.us-cert.gov/ncas/tips/ST04-002

B. Personalising Authentication Secrets

In terms of how to go about testing a personalised mechanism, recall will probably not work well. Cued-recall seems, at first glance, a good candidate. Yet, as we discovered from our experiences with cueblots, it is very challenging to provide a cue that makes sense only to the legitimate user and not to any subject expert.

Furthermore, in personalising, and driving towards increased memorability, we have to ensure that the cost for the user does not become excessive. We should not require too-consuming an engagement with the mechanism at each authentication attempt. Zviran and Haga's associated passwords are undoubtedly innovative, but one cannot imagine users welcoming this authentication if a system is used frequently.

A true knowledge-based mechanism might match a recognition-based paradigm very well, since it will test the user's knowledge requiring very little effort from them. One would have to use images which are somehow representative of the user's own skills. It would have to be a personalised facsimile, not a generic output.

For example, if we make use of handwritten mathematical proofs to authenticate mathematicians, we would have to display the user's proof, and then as distractors a number of proofs written by other users. We would expect the expert to identify their own proof, in their own handwriting. Hence authentication based on "something the expert produced". The question is, can humans reliably recognise what they themselves have produced? Moreover, can potential intruders do this reliably as well?

Humans can recognise a lot of things about themselves. For example their own voices [24], their own handwriting [25], [26], their own performance (pianists) [27] odour [28] and gait [29]. Hence images that are related to the user should make them easy to recognise but it might well also make them easier to guess. There are other ways of maximising recognition success. For example, a graphical mechanism using facial images could be tailored to maximise recognition by tailoring the entire challenge set to the age [30], race [31] and gender [32] of the user. This would help the user but not necessarily an attacker. All these variations would personalise the images to maximise the legitimate user's chances of being able to remember and identify their images.

Some authentication schemes have attempted to make use of personalised images. Dynahand [33] relies on the user being able to recognise his or her own handwriting. It collects 10 examples of participants' handwritten numerals at enrolment. It then generates new PINs using the user's own numerals, and generates distractors from other users' numerals. The full challenge set is displayed four times, and each time the user picks out the displayed PIN written in his or her own handwriting. A casual observer has less chance of gaining access to the user's account later because what is being tested, i.e. the handwriting, is relatively obscure and less easily cracked than a straightforward set of pictures. Moreover, it is completely effortless for the user.

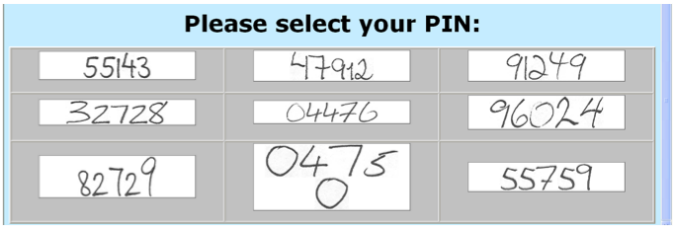

Fig. 2. A Dynahand Challenge Set

Renaud [21] deployed this technique as one stage of the Handwing authentication mechanism to control access to a website used by a community group where the community members very successfully identify their own handwriting to authenticate. The mechanism also exploits the user's ability to recognise their own hand-drawn doodle and has been very successful — and is still being used 10 years later. Renaud [34] also tested the same concepts with a graphical authentication mechanism that used Mikon (my icons) images. Users drew these using a browser-based engine. The majority of the participants in the study were able to remember all their mikons successfully after a 3 month period of non-use.

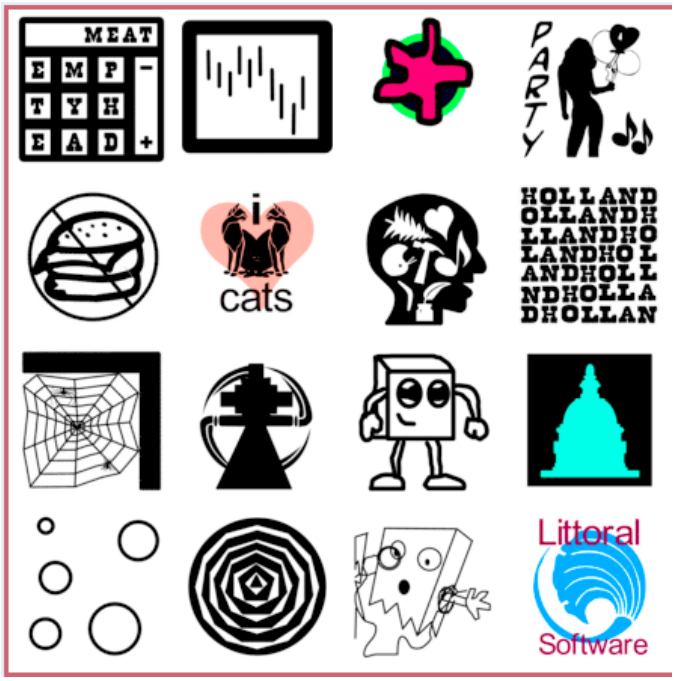

Fig. 3. A Mikon Challenge Set

C. Personalising Secrets for Experts

The schemes mentioned thus far did not exploit a particularly stringent or rare skill: almost everyone can write and draw. They do, however, demonstrate that people have the potential to remember, and to be able to identify, what they have produced. In the case of the drawn images, the images are more memorable than passwords because they rely on visual,

lexical and kinaesthetic memory [35] rather than mere textual memory.

Here we extend this concept to test whether experts can recognise the outputs from their own skilled actions, in this case programming language code. It takes thousands of hours to become a competent programmer [36]. Although there are millions of programmers in the world, the number is significantly smaller than those who can write and draw.

There are some potential advantages of such an expert-product based authentication.

1) It should reduce the potential number of attackers to those who hold the same knowledge as the genuine account owner. Presently any literate attacker can attempt to guess a nonsense-based password.
2) Craik and Tulving [37] argue that the development of memory traces should be considered in terms of *depth of processing*. Programming is a cognitively demanding task and so the production of an artifact should lay down strong memory traces.
3) It would not require the user to memorise anything. This addresses two common nonsense-based authentication problems.
    a) Firstly, users won't need to take precautions such as writing down their passwords.
    b) Secondly, it becomes much harder for them to record or share their passwords since it based on well-worn cognitive pathways and ability to reason, which is not easily conveyed to a non-expert.

The approach is not without its challenges. For example, we need to ensure that the authentication is not overly time-consuming. We also have to ensure that it is not too easy for another person to guess the user's secret code. To explore the viability of the scheme, we consulted a number of programmers.

IV. Fact Finding

In order to determine whether this idea had any chance of succeeding, we started off by posting an online survey. We advertised it to Masters students in our own School and also posted it to developer forums. We surveyed 129 programmers. Out of the 129, 121 had been programming for more than 3 years with the largest group in the 5-10 year category.

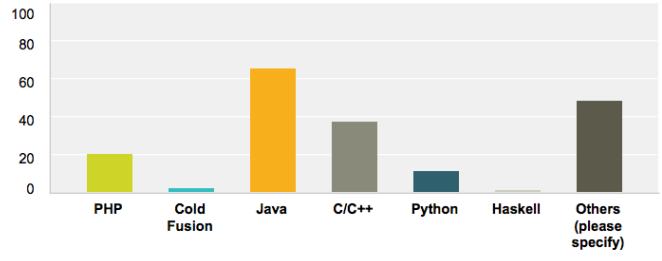

Fig. 4. Which Programming Language did They Use

Figure 4 shows the distribution of programming languages. Some people mentioned C#, ASP, Javascript, PL/1, Perl and Assembler. The most common used language was Java. We provided a box for comments.

77% of the respondents agreed with the statement: "Every Programmer has his/her own programming style". Figure 5 presents the responses. This appears to confirm the findings that people develop personal styles [38]. Some comments from the respondents:

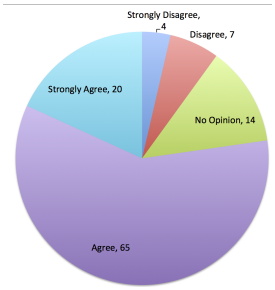

Fig. 5. Every Programmer has his/her own programming style

"programmers I knew all looked to add their own personalisation - it is their baby"
"It's a mistake if a programmer doesn't have his/her own programming style as it is important to recognizing your own programs"
"Programming is an expression of thoughts much like poetry. So a programmers individual style will be reflected in the piece of code that he/she develops. Bottom line is there can be several alternative solutions for a single problem, and different programmer may adopt different style. "

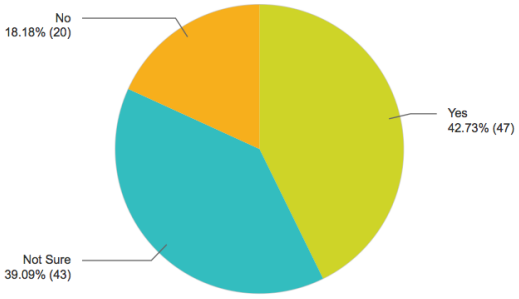

Fig. 6. Could you Identify your own code from a group of code snippets 10 lines long?

42% felt sure they could identify their own code, with another 39% being unsure (Figure 6). Only 21% felt they would be able to identify another programmer's code even if they knew the person well (Figure 7).

The survey responses convinced us that it would be good to trial a scheme which tested whether (1) people could recognise their own code and (2) how well people could recognise each others' code.

V. Experiment

We carried out a proof of concept experiment into the use of "what the expert produced" authentication. The area

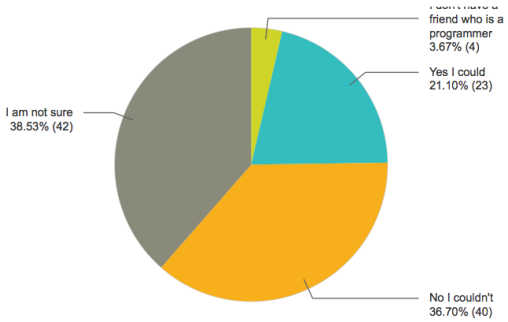

Fig. 7. Could you identify a friend's code from a group of code snippets 10 lines long?

of expertise we focused on was programming, since we had this knowledge ourselves and we worked in an environment that gave us access to a number of expert programmers. The aim was to design an authentication mechanism which would authenticate programmers based on their own programming skills, a genuine knowledge-based test. A recognition-based graphical authentication system which used snippets of code, instead of images, was implemented. We hoped to show that programmers would be able to recognise their own code, but that others, even those who are experts in the same language, would not easily be able to recognise the person's code snippet.

Our participants were 20 programmers. We asked them to provide 5 snippets of code in Java, since this was the most widely used language mentioned in our questionnaire. They were asked to avoid snippets containing comments. This constituted enrolment. We then asked them to return a week later to see whether they could identify their own code from four challenge sets where one piece of code was theirs and the other 15 came from other participants. Figure 8 shows what was required during authentication. Participants were required to identify their own code snippet from a challenge set composed of 16 code snippets. Distractors varied each time the user tried to authenticate, and different targets were chosen too. We had more code snippets than we needed for one authentication attempt. An example challenge set is shown in Figure 10.

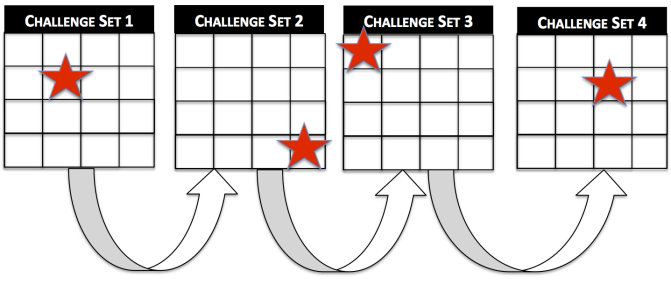

Fig. 8. Authentication - Four Challenge Sets

De Angeli *et al* [39] proposed three aspects to be considered when we examine the strength of an authentication mechanism: *guessability, observability* and *recordability*. In terms of usability we also have to test memorablity of the mechanism. Figure 9 shows how we tested each of these during our pilot study. We did not test recordability in this study, since this aspect deserves a study on its own.

Our participants were Masters students who had been together in the class for some 9 months and were assumed to be familiar. We tested memorability, guessability and observability as follows: participants worked in parallel. For example, Participant A would authenticate while participant B watched. Then Participant B tried to replicate the attempt. Participant C, on the other hand, attempted to guess Participant A's code based on their knowledge of participant A. Hence every participant observed another authenticating and tried to replicate the attempt. They also tried to guess one other person's codes without observing them.

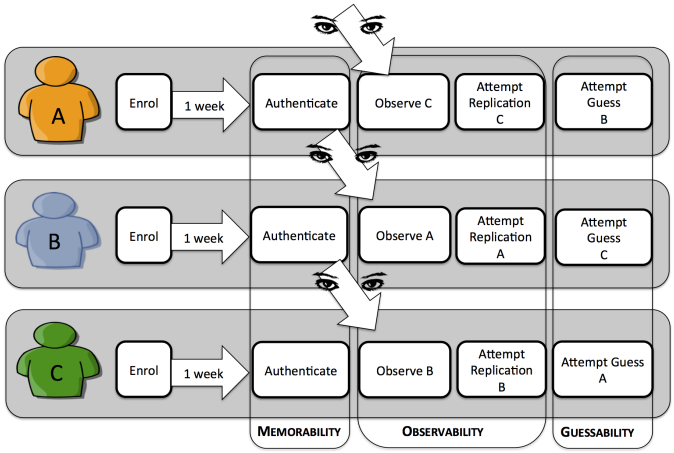

Fig. 9. Participants Working in Pairs

A. Results

Memorability: Identifying Their Own Code: All participants were able to identify their code, some almost immediately, but some needing some time to examine the screen. We asked them what particular aspect of the code made it so memorable. Some of them stated that they identified their variables, others functionality or Java class names. One participant identified his secret sequence of images in less than a minute as the variables were expressed in his national language.

Observability & Guessability: Identifying Another's Code: None of the "attackers" managed to identify another's code images, both when they observed the authentication and when they just tried to guess it. This is probably due to the fact that the images and the distracters are varied so the attacker would need to identify the programmer's style and not one specific piece of code.

Participant Comments: We asked participants to express their opinions about the mechanism when the experiment concluded. All reported finding it easy to locate their own code snippets. 17 of the 20 believed it would be impossible for anyone else to identify their images. Some specific comments:

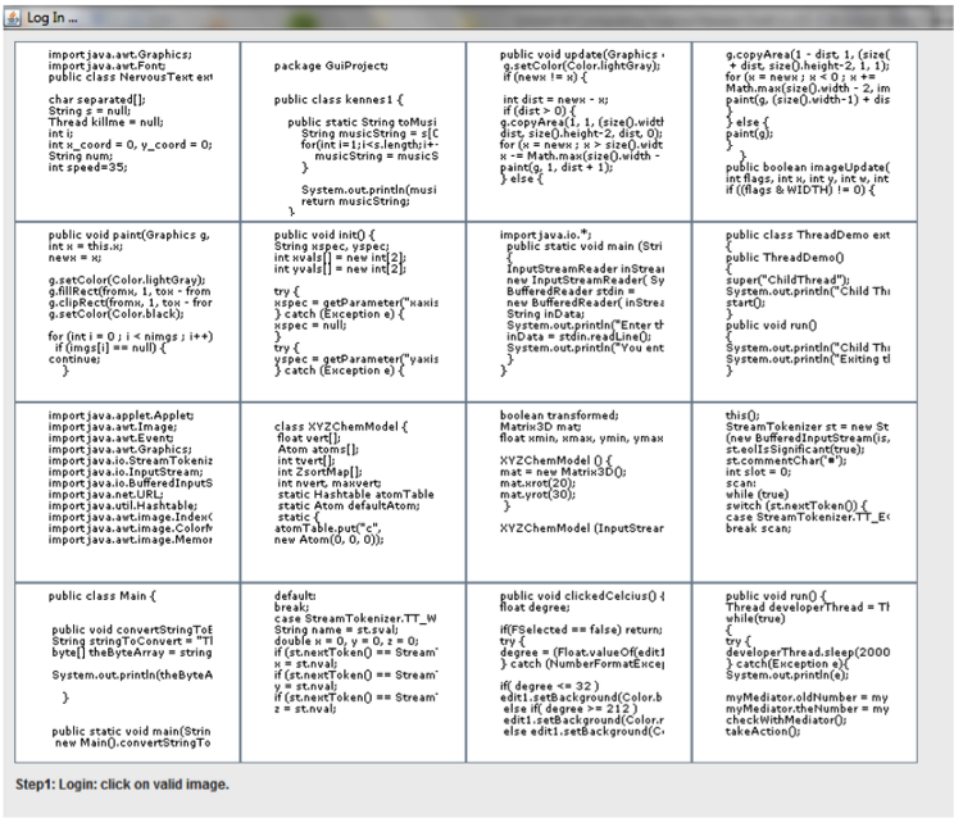

Fig. 10. An Example Challenge Set

"The idea of having code images as passwords is unique and I believe holds a good future"

"First time I used this mechanism was a bit difficult but gradually it became easy for me. Moreover I believe it is easier to remember images than text-based passwords."

VI. Discussion

Kurzban [40] outlines a minimal set of goals that all authentication mechanisms should achieve:

- *Avoid false positives (Allow an Intruder In)*
  An authentication mechanism must strive to ensure that attackers or undesirable individuals are not authenticated. If the primary uses of authentication are *control over access* and *accountability for actions*, then there must be confidence in an authentication mechanism. An authentication mechanism that permits too many false positives is not functionally fit for purpose.
- *Avoid false negatives (Keep the Legitimate User Out)*
  An authentication mechanism must minimise the scenario where an actual user is not authenticated. If legitimate users are unable to access services or complete transactions it does not only have a serious impact on an individual but on an organisation as well, potentially impacting profits.
- *Minimal burden on the User*
  The authentication mechanism must impose minimal burden on the user. The authentication mechanism can not be overly demanding.
- *Cost-Benefit Balance*
  The authentication mechanism should represent a cost-benefit balance. If the purpose of the authentication process is to *regulate access* or *confirm payment* then the expected user-effort should, proportionally, reflect the risk.

Whereas the knowledge-based mechanism performed well in terms of the first three, it fails to meet the last criterion for the developers deploying the mechanism. For the users the mechanism delivered a good cost benefit balance since no effort was involved in recognising their own code snippets. They provided these snippets themselves, which gave the advantage of memorability but since they were produced by skilled actions they were also less predictable than other schemes where users provided their own images [20]. Yet the manual selection of distractors, in order to ensure maximum strength, means that the system is not scalable. These images must be chosen carefully and should be purposely similar to the user's sequence of code snippets, in terms of programming language and perhaps the language used in the comments. In this way we could maximise the possibility that the distractors do not weaken the mechanism by making the target more easily guessed.

VII. Conclusion

In this paper we make an important point: the common and garden password cannot be referred to as and instance of "knowledge-based authentication". Passwords are ideally meaningless and therefore more akin to data than knowledge. We have tested a genuine knowledge-based mechanism, structured as a recognition-based graphical authentication mechanism, and found that it was possible successfully to test the products of skilled activities in an authentication setting. Moreover, such authentication appears to be both memorable and resistant to shoulder-surfing and guessing attacks. Certainly these preliminary findings suggest that further research is worthwhile.

References

[1] R. Smilkstein, "We're born to learn: Using the brain's natural learning process to create today's curriculum," 2003.

[2] NIST, "Guidelines on Evaluation of Techniques for Automated Personal Identification," National Institute of Standards and Technology, Tech. Rep. FIPS-PUB-48, April 1977.

[3] S. Pankanti, R. Bolle, and A. Jain, "Biometrics: The future of identification [guest eeditors' introduction]," *Computer*, vol. 33, no. 2, pp. 46–49, 2000.

[4] Corbató, F.J. and Merwin-Daggett, M. and Daley, R.C., "An Experimental Time-sharing System," in *Proceedings of the Spring Joint Computer Conference*. ACM, May 01-03 1962, pp. 335–344.

[5] B. Riddle, M. Miron, and J. Semo, "Passwords in use in a university timesharing environment," *Computers and Security*, vol. 8, no. 7, pp. 569–578, 1989.

[6] B. Schneier, "Two-factor authentication: Too little, too late." *Communications of the ACM*, vol. 48, no. 4, 2005.

[7] A. Conklin, G. Dietrich, and D. Walz, "Password-based authentication: a system perspective," in *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*. IEEE, 2004, pp. 10–pp.

[8] D. Florêncio, C. Herley, and B. Coskun, "Do strong web passwords accomplish anything?" in *Proceedings of the 2nd USENIX workshop on Hot topics in security*. USENIX Association, 2007, p. 10.

[9] S. Wiedenbeck, J. Waters, J. Birget, A. Brodskiy, and N. Memon, "PassPoints: Design and Longitudinal Evaluation of a Graphical Password System," *International Journal of Human-Computer Studies*, vol. 63, no. 1, pp. 102–127, 2005.

[10] Corbató, F.J., "On building systems that will fail," in *ACM Turing Award Lectures*. ACM, 1990.

[11] Federal Trade Commission, "Federal Trade Commission, Plaintiff, v. Wyndham Worldwide Corporation; Wyndham Hotel Group, LLC; Wyndham Hotels & Resorts, LLC; and Wyndham Hotel Management, Inc., Defendants (United States District Court for the District of Arizona)," August 2012. [Online]. Available: http://www.ftc.gov/os/caselist/1023142/120809wyndhamcmpt.pdf

[12] A. Adams and M. A. Sasse, "Users are not the enemy," *Comm. of the ACM*, pp. 40–46, 1999.

[13] G. Kreizman and A. Allan. (2006, November) Toolkit: Evaluating Enterprise Options for Managing Passwords. [Online]. Available: http://www.gartner.com/id=498322

[14] F. Cunha, J. J. Heckman, L. Lochner, and D. V. Masterov, "Interpreting the evidence on life cycle skill formation," *Handbook of the Economics of Education*, vol. 1, pp. 697–812, 2006.

[15] W. A. Rogers, "Assessing age-related differences in the long-term retention of skills," *Aging and skilled performance: Advances in theory and applications*, pp. 185–200, 1996.

[16] L. Anderson, D. Krathwohl, P. Airasian, K. Cruikshank, R. Mayer, P. Pintrich, J. Raths, and M. Wittrock, *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Complete Edition*, L. Anderson and D. Krathwohl, Eds. Longman, 2001.

[17] K. Renaud, T. McBryan, and P. Siebert, "Password cueing with cue(ink)blots," in *IADIS Computer Graphics and Visualization 2008 (CGV 2008) Conference*, Amsterdam. The Netherlands, 24 - 26 July 2008, pp. 74–81.

[18] M. Zviran and W. J. Haga, "Cognitive passwords: the key to easy access control," *Computers & Security*, vol. 9, no. 8, pp. 723–736, 1990.

[19] D. Davis, F. Monrose, and M. Reiter, "On User Choice in Graphical Password Schemes," in *Proceedings of the 13th USENIX Security Symposium*, 2004, pp. 151–164.

[20] T. Pering, M. Sundar, J. Light, and R. Want, "Photographic authentication through untrusted terminals," *Pervasive Computing, IEEE*, vol. 2, no. 1, pp. 30–36, 2003.

[21] K. Renaud, "A visuo-biometric authentication mechanism for older users," in *People and Computers XIXThe Bigger Picture*. Springer, 2006, pp. 167–182.

[22] R. Dhamija and A. Perrig, "Déjà Vu: A User Study Using Images for Authentication," in *Proceedings of the 9th Conference on USENIX Security Symposium*, 2000.

[23] R. English and R. Poet, "Measuring the revised guessability of graphical passwords," in *Network and System Security (NSS), 2011 5th International Conference on*. IEEE, 2011, pp. 364–368.

[24] C. Fernyhough and J. Russell, "Distinguishing ones own voice from those of others: A function for private speech?" *International Journal of Behavioral Development*, vol. 20, no. 4, pp. 651–665, 1997.

[25] M. Longcamp, J.-L. Anton, M. Roth, J.-L. Velay *et al.*, "Visual presentation of single letters activates a premotor area involved in writing," *Neuroimage*, vol. 19, no. 4, pp. 1492–1500, 2003.

[26] M. Longcamp, T. Tanskanen, and R. Hari, "The imprint of action: motor cortex involvement in visual perception of handwritten letters," *Neuroimage*, vol. 33, no. 2, pp. 681–688, 2006.

[27] B. H. Repp and G. Knoblich, "Perceiving action identity how pianists recognize their own performances," *Psychological Science*, vol. 15, no. 9, pp. 604–609, 2004.

[28] M. Schleidt, "Personal odor and nonverbal communication," *Ethology and Sociobiology*, vol. 1, no. 3, pp. 225–231, 1980.

[29] D. Jokisch, I. Daum, and N. F. Troje, "Self recognition versus recognition of others by biological motion: Viewpoint-dependent effects," *Perception*, vol. 35, pp. 911–920, 2006.

[30] J. S. Anastasi and M. G. Rhodes, "Evidence for an own-age bias in face recognition." *North American Journal of Psychology*, 2006.

[31] J. Stahl, H. Wiese, and S. R. Schweinberger, "Expertise and own-race bias in face processing: an event-related potential study," *Neuroreport*, vol. 19, no. 5, pp. 583–587, 2008.

[32] D. B. Wright and B. Sladden, "An own gender bias and the importance of hair in face recognition," *Acta psychologica*, vol. 114, no. 1, pp. 101–114, 2003.

[33] K. Renaud and E. Olsen, "Dynahand: Observation-resistant recognition-based web authentication," *Technology and Society Magazine, IEEE*, vol. 26, no. 2, pp. 22–31, 2007.

[34] K. Renaud, "Web authentication using mikon images," in *Privacy, Security, Trust and the Management of e-Business, 2009. CONGRESS'09. World Congress on*. IEEE, 2009, pp. 79–88.

[35] K. Renaud and A. De Angeli, "Visual passwords: cure-all or snake-oil?" *Communications of the ACM*, vol. 52, no. 12, pp. 135–140, 2009.

[36] M. Gladwell, *Outliers: The Story of Success*. Penguin, 2009.

[37] F. Craik and E. Tulving, "Depth of Processing and the Retention of Words in Episodic Memory." *Journal of Experimental Psychology: General*, vol. 104, no. 3, pp. 268–294, September 1975.

[38] J. J. Heckman, "The economics, technology, and neuroscience of human capability formation," *Proceedings of the National Academy of Sciences*, vol. 104, no. 33, pp. 13 250–13 255, 2007.

[39] A. De Angeli, L. Coventry, G. Johnson, and K. Renaud, "Is a picture really worth a thousand words? exploring the feasibility of graphical authentication systems," *International Journal of Human-Computer Studies*, vol. 63, no. 1, pp. 128–152, 2005.

[40] S. Kurzban, "Easily remembered passphrases: a better approach," *ACM SIGSAC Review*, vol. 3, no. 2-4, pp. 10–21, 1985.