Towards Narrative Authentication

Or, Against Boring Authentication

Anil Somayaji
School of Computer Science
Carleton University
Ottawa, Canada
soma@scs.carleton.ca

David Mould
School of Computer Science
Carleton University
Ottawa, Canada
mould@scs.carleton.ca

Carson Brown
Shopify, Inc.
Ottawa, Canada
carson@carsonbrown.ca

ABSTRACT

In this paper we propose that what-you-know authentication schemes be built using narrative elements. Specifically, we propose that stories be used as the basis of memory-based user authentication, rather than use a fixed string as the secret for authentication (as is the case with text passwords and PINs). The insight here is that secure text passwords are "boring" and, hence, are hard to remember. Narrative is, in contrast, extremely memorable, forming the basis of much of human communication. We present a simple, implementable scheme for narrative authentication using text adventures. We then also examine other strategies for generating and testing knowledge of narrative.

Categories and Subject Descriptors

Security and Privacy [**Security services**]: Authentication; Security and Privacy [**Human and societal aspects of security and privacy**]: Usability in security and privacy

General Terms

Security

Keywords

authentication; narrative; text adventures

1. INTRODUCTION

With the most commonly used forms of user authentication—text passwords and PINs—maximum security is achieved through using many long random passwords. Time and again, though, it has been shown that people do the opposite, choosing to use a small number of short non-random passwords.

This tendency to minimize the memorization effort involved with passwords would, on its own, imply that human memory was a scarce resource. Yet we have ample evidence to the contrary. Even setting aside the fact that our brains have billions of neurons, our personal experience points to the vast capacities of our memories. Just consider the ability of an adult to remember childhood events, gossip about our peers, events from the news, or stories they have heard. While the details can be fuzzy, and while we clearly forget many things, we can recall many, many more things than we ever have time to even think about, let alone communicate with others about.

As our experiences with passwords shows, though, our vast resources of memory cannot be so easily used to address many memory tasks, at least for those of us who do not possess eidetic (photographic) memory—our memory fails us at key junctures. We forget where we placed our keys. Important details of newly encountered people, such as their names or even their faces, can often be forgotten in a matter of moments. Even our memories of extreme events is suspect: witnesses routinely make profound errors in recalling the details surrounding crimes [17].

To summarize these observations, some things are much, much easier for us to remember than others, and the fidelity of our memory is directly related to the nature of those memories. Some things that we wish to remember—but that we find "boring"—are not retained, while memories we would prefer to forget but that are somehow "interesting" stay with us. As most of us have experienced, long, random passwords generally fall into the easily forgettable, boring category. So, if we are to authenticate people based upon human memory, perhaps it might be advisable to choose a form of information that is more interesting and therefore, easier to remember?

This line of thinking certainly underlies many proposed alternative authentication schemes such as those based on images [16], faces [3], and gestures [12]. User testing, though, has shown that users also have clear limitations in their ability remember distinct authentication patterns in these domains as well [2].

But, what if we had a form of memory that was optimized for memorability? This form would also need to be easy to store, communicate, and, most importantly, verify. While it is probably unrealistic to achieve a zero-knowledge property with a human-machine protocol, eavesdropping on the verification process should not permit trivial replay attacks. Further, it should be possible for a person to remember many distinct patterns to minimize credential reuse.

We posit here that we have such a form of memory: stories. Good stories are almost impossible to forget, and even bad stories can be remembered. We teach each others through stories, both fictional and non-fictional. We con-

NSPW'13, September 9–12, 2013, Banff, AB, Canada.

ACM 978-1-4503-2582-0/13/09 ...$15.00.
http://dx.doi.org/10.1145/2535813.2535820.

vert our experiences into stories so that they can be shared and remembered. In fact, people often tell stories to verify each other's identities by verifying that they both share some common set of stories, often using exchanges that are unintelligible to others who do not know those same stories. Further, those exchanges can be remarkably quick and concise. Thus, stories cover virtually all of the characteristics we might want out of a memory-based authentication scheme except for one: implementation difficulty.

Today we do not know how to have computers understand stories, and indeed this task could be considered to be AI-complete, in that solving it completely might lead to an artificial intelligence of human-or-better capability. However, as research on CAPTCHAs has demonstrated, human cognitive strengths can be exploited for security purposes even when they cannot be duplicated. In the rest of this paper we argue that narrative elements—aspects of stories—could be used as the basis of a variety of authentication schemes that exploit our ability to remember stories.

The rest of this paper proceeds as follows. We first discuss narrative in computational systems in Section 2. We then examine what it would mean to authenticate using narrative in Section 3. In Section 4 we describe and critique a simple authentication scheme incorporating narrative elements. Section 5 discusses possible narrative-based authentication schemes based upon the source of their narratives. We examine the issue of memorability and the attacks the memorability of narratives could engender in Section 6. Section 7 concludes.

2. COMPUTATIONAL NARRATIVE

While narrative has not been previously applied to the problem of authentication, narrative has long been part of computer science, primarily through computer games. Computer games typically present narrative through an iteration of a two-part sequence, consisting of a scene followed by some kind of interaction. The scenes are usually presented in a formal way, using blocks of text, short movie clips ("cutscenes"), or dialogue; the scenes are fully preplanned and are replayed at runtime. The interactions between scenes may be very short, involving something as simple as making a selection from a menu of options, or they may consist of extended gameplay, such as playing through a mission of a real-time strategy game or first-person shooter. Choices made and outcomes in the gameplay may or may not affect the next scene. This structure has its roots in older, non-computational forms, such as Cortazar's *Hopscotch* or the *Choose Your Own Adventure* gamebooks.

Scene sequences in games can be linear, strictly branching, or have some other graph structure [9]. In principle game narratives could be extremely rich and varied, with players receiving different narratives on every playthrough. A rich narrative structure could potentially form the basis of an authentication system: a user's "key" could simply be a specific narrative that is chosen by how a person chooses to play the game.

The narratives in most games, however, do not have anywhere near the number of variants that would be needed for a space of authentication tokens. It is difficult to build large numbers of satisfying playthrough paths, and cutscenes and dialogue can be costly, so narrative scenes are few and sparse relative to the gameplay, and a given playthrough will reach a large proportion of the scenes. Ip [10] reports that in sample playthroughs of selected games, the proportions of game time devoted to narrative elements range from as little as 1% to a high of only 28%. To minimize the visibility of a linear, deterministic game structure, many games are structured as a "string of pearls" [13]. In this structure, "pearls" of more or less unconstrained gameplay are linked by "strings" where only a single path is possible; story advances chiefly through narrative events at the gateways between pearls.

While game narratives are often linear, hence paralleling the structure of other narrative art forms such as movies and novels, that does not mean that the computational narrative of games is equivalent to that of other mediums. For example, narrative events can be divided into *kernels* and *satellites* [7]. Kernels correspond to major plot points: the sequence of kernels is the skeleton of the plot. Satellites are ancillary events that amplify, explain, and reinforce the kernels. The rich narratives of novels and films, and even the simple narratives of newspaper comic strips [7], have few kernels with many satellites per kernel. However, game narratives contain mostly kernels [10]. Since satellites provide context and aid the audience in understanding character motivations and cause and effect, a story lacking satellites will have a barren, inscrutable plot. In such a story, the connections between events are poorly enunciated and difficult for the audience to follow.

While the computational narratives as embodied by modern games are limited, the ability of our computers to understand narrative is limited even more. This inability to process narrative has profound consequences on the structure of games. In games we can move about worlds, interact with objects, and kill enemies, yet we cannot have simple conversations. Computers can tell us stories, but we cannot tell computer stories. As narrative forms the heart of social interactions, computer games have, to date, had limited success in building games based on simulated social interactions. Where social-like interaction has been required, games instead become multiplayer, thus allowing the sharing of narratives between people.

The linear nature of much of computational narrative and the inability of computers to understand narrative might together seem to doom the idea of using narrative for authentication purposes. However, as we will show, by breaking down the concepts of authentication and narrative, we may be able to make some progress.

3. AUTHENTICATION

To authenticate means to verify the identity of another party. In most computer systems authentication is performed by verifying the possession of a secret, which is either information that only the authenticating party possesses or information that is shared between the authenticating party and the authenticator. For example, when a user authenticates to a remote website using a password, the website is verifying that the user knows a secret—the password—that the website has associated with that user's account. To make such an authentication system secure, the secret should be arbitrarily chosen from a sufficiently large space of possibilities.

As previously discussed, entire game playthrough narratives are a poor choice for encoding a secret simply because there are relatively few of them in any game. Rather than use entire game stories, however, we may be able to use components of stories—narrative elements. Narrative el-

ements can be places (settings), characters, objects, and events (plot). To authenticate using narrative elements, we must convert them into a form that a human can remember and a computer can verify knowledge of. The simplest strategy of using a raw textual description of narrative elements, e.g., a textual description of a place, is not feasible for multiple reasons. First, most such descriptions will be relatively long—hundreds or thousands of characters—making them time consuming to enter. They will also be difficult to enter precisely, both because of their length and because people are likely to slightly change the text when it is recalled. And third, such a literal representation will be just as susceptible to replay attacks as standard text passwords. Clearly for this to be feasible we need something better than an extra long passphrase.

Instead, what we want is a form of challenge-response. The remote server should store a complex narrative structure—a story or a set of stories—that is then used to drive a dialogue with the user. The system sends challenges to the user that requires knowledge of the stories to be successfully responded to but can be responded to using information derived from only a small portion of the narrative structure.

The question now is how to choose and represent the narrative in a way that minimizes the effort required to make the "secret," minimizes the likelihood that these secrets are shared between multiple services and known to unauthorized individuals, and allows for a wide variety of challenges than can be quickly and easily responded to yet are varied enough to reduce the impact of replay attacks.

We next describe an example authentication scheme based on narrative elements, after which we describe alternative ways of approaching authentication through narrative.

4. A TEXT ADVENTURE-BASED SCHEME

While we cannot yet hope to capture the full complexity of narrative interactions in an authentication system, we can use narrative elements to form a very simple authentication system using a very old but still remarkably powerful type of game: text adventures. Text adventures are a form of interactive fiction similar to but more advanced than hypertext fiction works such as *Choose Your Own Adventure*. They are played by invoking text commands to move a character (commonly from first person perspective) about an environment. Interaction makes use of natural language, albeit with a severely restricted form and vocabulary. Nevertheless, modern interpreters are capable of taking a variety of text inputs and mapping them to a single, equivalent action.

Early text adventures were very difficult to program, particularly on the limited computers available in the late 1970's and early 1980's. Modern interactive fiction, however, can be built with much better tools. For example, the Inform interactive fiction programming language environment [14], with its built-in knowledge of the room-based structure, inventory control, navigation commands, and other standard elements of text adventures, is remarkably easy to use. With version 7, Inform even allows adventures to be created using a natural language-like structure that can almost be read as a story.

Inform 7 is flexible enough to create adventures that double as authentication mechanisms. In Listing 1, we show a dialog with a simple text adventure we call Stacker. Stacker is designed to give PIN-like security through a text adventure-type interface. The "player"—the authenticating user—must

```
Stacker

Stage
You can see a table here.

> > inventory

You are carrying:
  a bowl
  a hammer
  a ruler
  a paintbrush
  a soap
  a telephone
  a pair of scissors
  a cup
  a towel
  a camera

> > stack hammer

You put the hammer on the table.

> > stack paintbrush

You put the paintbrush on the hammer.

> > stack soap

You put the soap on the paintbrush.

> > stack cup

You put the cup on the soap.

As you drop the last item down, you
notice a small crease in the wall.
Pushing on it, you find a small doorway,
and walk through...

     *** You have won ***
```

Listing 1: An example of a successful authentication attempt for Stacker.

stack the objects in the correct order in order to authenticate. The source for Stacker is in Appendix A.

Note that in its current form Stacker is not a particularly good narrative, and as such it is not very memorable. However, this same structure could be used to tell any number of more memorable narrative fragments. The room could be described as a child's bedroom, and the "stacking" of objects could involve placing objects in the correct position on a shelf. If the user had strong memories of this place and the arrangement of objects (say, by recalling a childhood memory), the presented scene could be quite memorable.

Also, we can make this scheme more secure by making other kinds of changes [4]. Resistance to brute force attacks could be improved by increasing the number of objects to be arranged or by adding other characteristics such as orientation or size. Replay attacks could be minimized by randomizing the objects' characteristics in some way that would still appear to be equivalent to an authorized user (e.g., by changing their color) or the user could be asked to only rearrange a random subset of the objects in the room. Unfortunately all of these changes could impact the difficulty of the authentication task in terms of authentication time and the memorability of the required behavioral constraints.

Another aspect that can be incorporated into these types of schemes is the presence of "traps" that non-authorized users may not expect. Databases of related but distinct objects can be scattered throughout the user's text adventure. The act of inserting "blue shoes" into the front hallway of a user's scheme that involves picking up memorable items in a house has little effect for the authorized user, but this object must be also neglected by any would-be attackers. The intention here is the same effect as seeing pencils, rulers and scissors in a drawer when searching for a marker—to the intended user, these other objects are simply noise to be filtered out and ignored.

In an unfamiliar text adventure, every object requires inspection. In a text adventure resembling a memorable place and time (e.g., exploring grandmother's house as a child), automatically inserted generic but context-appropriate objects will confuse attackers while being ignored by authorized users. Continuing the example, objects such as a collectible spoon, walnut picture frames, or a pink glass vase could represent particularly fond childhood memories (and thus be used for authentication-related actions) while a decorative china plate or wall-mounted bottle opener might have no sentimental connection and thus would be ignored by the authorized user.

While it is possible for users to create text adventures for authentication purposes, we regard this proposal more as a proof-of-concept that narrative can be incorporated into authentication schemes. As we will now discuss, though, there are a number of other ways that narrative could potentially be incorporated, particularly if we assume certain types of technological advancement.

5. NARRATIVE SOURCES

There are two key design choices to be made when designing a narrative-based authentication scheme. The first is where the narrative should originate: from the user, from the computer, or from a collaboration between the two. The second is how the user should interact with a narrative. We cover the first choice here and the second in the following section.

5.1 User-Imparted Narratives

The most obvious source of narrative elements for authentication is the user. As with our Stacker game, the user can choose a scene, objects, characters, and plot so as to make a story or story fragment that is memorable. As with user choice with passwords, however, it is possible that users will tend to create simple authentication adventures and will reuse these "credentials," thus reducing the potential security benefit of moving towards narrative-based authentication. This is particularly a risk if the effort involved is on the scale of creating a text adventure from scratch for every resource for which one would use a password. Even with sophisticated authoring tools that could import narratives (e.g., from home movies), the work required would still likely be much, much greater than that required for generating a password. And this is all assuming that the number of interaction required for authentication would be small (to make authentication fast) while also being hard for an attacker to compromise (challenges that are hard to guess).

With user-imparted narratives, the user must be sophisticated and motivated enough to choose combinations of narrative elements that will be secure against adversaries. And again, as with passwords, we suspect this will be unlikely in practice.

5.2 Computer-Inferred Narratives

Rather than have the user create the narrative, another option is for the computer to generate one. A pure computer generated story is unlikely to be memorable (simply because it probably won't be very good), so purely randomly generated stories is probably not a promising strategy. However, there is no need for purely random stories when our computers have something much richer to draw upon: their record of interactions with users.

If we regard narrative as a type of summary of experience, it might be possible to automatically extract narratives from past records of user activity. For example, you could authenticate to your PC by answering questions regarding a "story" generated from user behavior over the previous few days. The difficult part of this task is to extract the interesting parts (from the user's perspective) while ignoring the boring parts. A person might have spent many hours on their computer the previous day, but most of that behavior might have been routine and so most of the recorded activity would not be memorable enough for authentication purposes; however, the user choosing to play a game for the first time, even if only for 15 minutes, might be the interesting activity that is worth incorporating into an authentication narrative.

Existing work in text summarization [1] and affect detection [5] might be of use here; however, this type of story extraction from user behavior records is largely an open problem. This is perhaps a good thing because computational extraction of interesting user activity could be a bit disconcerting to many users.

Inferring a narrative need not be based only on text summarization. Social media websites like Facebook and Foursquare already encourage personal location recordings via "check-ins." Foursquare's Time Machine [8] identifies destination preferences and when and where trips occur. Suggestions based on these discovered trends are part of their business model, but this type of summarization can initiate a computer inferred narrative that describes a recent holiday. Repeat visits to a particular location (e.g., hotel) and

check-ins at places with strong correlation to the user's profile (e.g., a highly rated restaurant by a foodie) can quickly form a narrative that resonates with a user's memories of the recent excursion.

5.3 Collaboratively Constructed Narratives

Potentially the most fruitful option for building narratives for authentication purposes is to create them through an interactive process between the computer and the user. Whether the story is fictional or non-fiction, whether it is based upon a pre-existing story generator or an extraction of usage patterns, combining input from both can help minimize the weaknesses of each. By having the "raw material" of the narrative not come from the user directly, it is less likely that it will have insufficient entropy to ensure security. Similarly, by having the user choose which of the pre-selected elements to use and which to ignore, the user can ensure the choices are sufficiently memorable to reduce the difficulty of authentication.

While the potential benefits might be clear, how this collaboration could be generated is ambiguous. Again, ideally we would allow the user to enter their opinions in free-form text to edit and supplement the automatically-generated narrative elements. The more flexibility given to the user, however, the more difficult the task becomes. In practice the dialog would probably involve highly constrained user choices at every stage, at least initially. Advances in natural language processing, however, might allow for more flexible collaborative story creation.

Building on the automatically constructed narrative that involves key locations visited, a user can fill in the narrative in a guided way. This creates a story worth remembering by adding causality and motivation. Automatically building a story about an unusual Sunday morning hardware store visit builds on novelty, but the story is not significant to the user until the reason is established. A user might not remember specific events at the hardware store, but by asking why, the story can shift to more memorable events—the kitchen sink clogging and overflowing.

This act of building upon a given scaffold of a narrative is a well-established component of Improvisational Theatre. The act of combining apparently disparate fragments of stories together is what makes the performance enjoyable, but also provides novelty. "Scaffolds" of story pieces for use in improv theatre are memorized, and can be combined quickly to the existing narrative by the actors in stage. Johnstone [11] codifies the act of creating good improvisational theatre, where initial fragments are first established, connected, then lastly reincorporated: an element introduced but cast aside early in the story reappears at a crucial moment to complete the scene. Collaboratively constructing a narrative in this way gives the user the opportunity to provide causal connection to disparate fragments, and the computer can suggest improvised reincorporations of linked fragments.

The format for such an improvised story can occur the same way Johnstone suggests improv theatre be brainstormed: via rounds of collaborative question-response sessions, linking fragments to other fragments. Computerized facsimiles of inquisitive counselors have existed for decades, involving simple natural language pattern matching techniques that explore by nondirectional questions based on user input (such as ELIZA [15]). A sufficiently clever ELIZA-bot could also provide necessary reintroduction of discarded fragments from earlier in the story, asking questions to draw the user into providing necessary connection points.

6. NARRATIVE AND MEMORY

To authenticate via a narrative, there needs to be a dialog between the computer and the user. This dialogue is essential because the narratives, on their own, will be too voluminous to be communicated by the user to the computer on every authentication. Generating appropriate challenges, however, requires that the narrative be represented in a way that facilitates such challenge generation. Again here the ideal interaction model would be a human-like dialog, where the computer and user would go back and forth sharing key details of the narrative until both parties were convinced they were referring to the same story.

The central problem, then, is what do we mean by the same story? Even for memorable stories, people will often forget or change small details in ways that other people might notice but that do not effect story "equality." Indeed, in general people will store in their heads different variations on narrative summaries of past events, simply because of their different observation viewpoints and past history. This is perhaps the central impediment to having a more free-form dialogue with a computer regarding a story: we don't know how to represent this almost-equivalence.

We can of course have the computer drive the conversation, limiting user choice to options that can be appropriately processed. This is how games do interactive narrative today, and it can be effective in some circumstances. However, such limits greatly restrict the space of possibilities, thus making guessing and other attacks more feasible. If the restricted questions are drawn from a sufficiently rich narrative source, it should still be possible to make the authentication mechanism resistant to attack.

A more fundamental problem, though, is that efforts to create a narrative structure themselves reduce the entropy of the source. Story patterns such as the monomyth [6] and, more generally, mythological archetypes could provide the basis for a kind of semantic dictionary attack on stories, particularly if those stories are pure fiction. We hypothesize that the extent to which the underlying narrative is drawn from information only known to the user and system being authenticated to, the more secure the subsequent authentication will be. Nevertheless, we should expect that, just as movie critics can predict the endings of movies after watching them for only a few minutes, attackers may become skilled at guessing appropriate story elements.

One consolation of such a reduction in security, though, is that it will likely require the attackers to address key problems in artificial intelligence, namely how to represent narrative and conversation. Any such advances could then be incorporated into better narrative-based authentication schemes.

7. CONCLUSION

In this paper we propose that what-you-know authentication schemes be built using narrative elements much as they are used in computer games. Stories could be created by the user (in the form of a text adventure or other interactive system), could be generated using records of user behavior, or could be semi-automatically created through automated selection of narrative elements that are then re-

fined by the user. Authentication would then involve a user demonstrating that they know the narrative stored by the computer, ideally through free-form dialogue about the narrative but more likely through answers to a small set of restricted questions. It is likely that if deployed, attackers will learn to exploit narrative-based authentication by exploiting the predictable structure of narratives. Such advances, however, would likely also enable the creation of better narrative-based authentication schemes. While simple schemes based upon existing text adventure technology can be implemented today, the most promising strategies will require significant research into extracting narrative elements from records of user behavior and transforming those elements into appropriate challenges for the user. While this research is non-trivial, and while any potential scheme will require extensive user testing, it is also research that could impact computer games and artificial intelligence in addition to the field of computer security. We thus hope other researchers attempt to develop these ideas further.

Acknowledgements: This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Discovery grants program, the ISSNet strategic network, and the GRAND NCE.

8. REFERENCES

[1] R. Barzilay, M. Elhadad, et al. Using lexical chains for text summarization. In *Proceedings of the ACL workshop on intelligent scalable text summarization*, pages 10–17. Madrid, Spain, 1997.

[2] R. Biddle, S. Chiasson, and P. C. Van Oorschot. Graphical passwords: Learning from the first twelve years. *ACM Computing Surveys (CSUR)*, 44(4):19, 2012.

[3] S. Brostoff and M. A. Sasse. Are passfaces more usable than passwords? a field trial investigation. In *People and Computers XIV—Usability or Else!*, pages 405–424. Springer, 2000.

[4] C. Brown. A meta-scheme for authentication using text adventures. Master's thesis, School of Computer Science, Carleton University, Ottawa, Canada, December 2010.

[5] R. A. Calvo and S. D'Mello. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, 1(1):18–37, 2010.

[6] J. Campbell. *The hero with a thousand faces*, volume 17. New World Library, 2008.

[7] S. Cohan and L. M. Shires. *Telling Stories – a theoretical analysis of narrative fiction*. New Accents. Routledge, 1988.

[8] Foursquare Labs, Inc. Say hello to the foursquare time machine. `http://blog.foursquare.com/2013/06/13/say-hello-to-the-foursquare-time-machine/`, 2013. Accessed July 29, 2013.

[9] A. S. Glassner. *Interactive Storytelling: Techniques for 21st Century Fiction*. AK Peters Series. Taylor & Francis, 2004.

[10] B. Ip. Narrative structures in computer and video games: part 1: context, definitions, and initial findings. *Games and Culture*, 6(2):103–134, March 2011.

[11] K. Johnstone. *Impro: improvisation and the theatre*. Theatre Arts. Routledge, 1979.

[12] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675, 2009.

[13] J. Majewski. Theorising video game narrative. Master's thesis, Bond University, Robina, Australia, 2003.

[14] G. Nelson et al. Inform 7. `http://inform7.com/`, 2006-2013. Accessed April 12, 2013.

[15] J. Weizenbaum. ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine. *Commun. ACM*, 9(1):36–45, 1966.

[16] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon. Passpoints: Design and longitudinal evaluation of a graphical password system. *International Journal of Human-Computer Studies*, 63(1):102–127, 2005.

[17] Wikipedia. Eyewitness memory. `http://en.wikipedia.org/wiki/Eyewitness_memory`. Accessed April 12, 2013.

APPENDIX

A. STACKER

```
"Stacker" by "Carson Brown"

The Stage is a room.

Use no scoring.

The bowl is a supporter carried by the
    player. The bowl is not fixed in place.
The hammer is a supporter carried by the
    player. The hammer is not fixed in
    place.
The ruler is a supporter carried by the
    player. The ruler is not fixed in
    place.
The paintbrush is a supporter carried by
    the player. The paintbrush is not
    fixed in place.
The soap is a supporter carried by the
    player. The soap is not fixed in place.
The telephone is a supporter carried by
    the player. The telephone is not fixed
    in place.
A pair of scissors is a supporter carried
    by the player. The pair of scissors is
    not fixed in place.
The cup is a supporter carried by the
    player. The cup is not fixed in place.
The towel is a supporter carried by the
    player. The towel is not fixed in
    place.
The camera is a supporter carried by the
    player. The camera is not fixed in
    place.

The table is a supporter in the stage.

The Proper Stack is a list of objects that
    varies. The Proper Stack is usually
    {table, hammer, paintbrush, soap, cup}.
```

```
The Chosen Stack is a list of objects that
    varies. The Chosen Stack is usually
    {table}. [The exact stack the player
    makes.]

The Last Object is an object that varies.
    The Last Object is usually the table.

After putting a supporter on the Last
    Object:
  Add the noun to the Chosen Stack;
  Now the Last Object is the noun;
  Now the noun is fixed in place;
  Continue the action.

Understand "stack [something]" as
    stacking. Stacking is an action
    applying to one thing.

Check stacking:
  if the noun is carried by the player:
    try putting the noun on the Last
    Object.

Every turn:
  [say "Your Chosen Stack is [Chosen
    Stack].";]
  [say "The Proper Stack is [Proper
    Stack].";]
  If Chosen Stack is the Proper Stack:
    say "As you drop the last item down,
    you notice a small crease in the wall.
    Pushing on it, you find a small
    doorway, and walk through...";
    end the game in victory.
```

Listing 2: The complete Inform story file for Stacker.