Varun Almaula and Taurin Tan-atichat
March 23, 2007
CSE 227

Security Research Project
Deterring Shoulder-Surfing Using Inverting Graphical Authentication

Introduction and Motivations

Graphical passwords are gaining popularity as an authentication mechanism. Users of such systems authenticate themselves by identifying some subset of image information from a larger set of image information presented on a display. Graphical passwords are often categorized as a recognition-based method as opposed to traditional recall-based methods, such as PINs and text passwords [1]. Humans have a high aptitude to distinguish and process information in images, and thus recall-based methods have the potential to provide accurate and user-friendly authentication. Additionally, graphical passwords are also thought to be harder to defeat by some members of the security community. Unfortunately, some of the early graphical password systems were also quite vulnerable to shoulder-surfing attacks. Since all visual information is presented onscreen, any input methods that directly indicate the user's action onscreen (e.g. using the mouse to point to the correct image) make it quite simple for a shoulder-surfer to defeat the system. Tari et al. introduced the Passfaces system to try to combat shoulder-surfing [2]. Results from this study showed improved capability of defeating shoulder-surfing by mapping image selection from a 3x3 grid to a number pad entry (1-9) method. Humans were generally unable to observe both the graphical images presented on screen and the user keyboard input simultaneously. However, the Passfaces system was not tested against any sort of electronic (camera or video) based shoulder-surfing attack.

The work described in this paper aims to identify the severity of the vulnerability that camera and video recording pose to systems such as Passfaces while introducing a new method of authentication that deters such attacks. The proliferation of camera and video enabled cell phones greatly increases this threat as shoulder-surfing can be made less invasive, automated, and anonymous. A not too farfetched scenario may include an attacker using a cell phone behind someone in an ATM line. The attacker can act as if he is using his cell phone to dial a call or write a text message (looking unassuming to the ATM user), while instead he can be taking photos or video recording the user's entire ATM session. An automated attack with a remote camera is equally disconcerting.

In a graphical password system, there should be great care in trying to decouple the presentation of visual information from user input indicating response to the presented information. For example, in the Passfaces system if it is possible to hide the keyboard from an attacker during user input, it would reduce the likelihood of defeating the system. Likewise, hiding the visual information presented on screen from the attacker would also reduce the likelihood of defeating the system. If an attacker possesses only the keystrokes of the user, they have no visual information to match it against (especially since the positions of faces in the Passfaces grid randomizes every session). If the attacker posseses only the presented visual information, they will be unable to determine what the user selected to authenticate themselves (however, a determined attacker may be able to use visual information collected from several authentication sessions of the same user to defeat the system). An ideal system would hide both the visual information and the method of input from an attacker.

Our proposed system utilizes the difficulty with which cameras and video recording devices have in adjusting critical settings (focus, aperture, shutter speed, white balance, ISO, etc.) in reaction to a scene in which the amount of light has been almost instantaneously inverted (a very bright scene suddenly turns dark or vice versa). In comparison, the human visual system is far more adept at adjusting to such a change. By displaying a graphical password in this context, cameras and video recording devices will not be able accurately capture what was displayed to the user. Additionally to prevent humans and devices from successfully shoulder-surfing, graphical passwords will be in the form of a set of small grayscale icons, making it extremely difficult for eavesdroppers or devices to recognize images.

While this project is a classic example of the tradeoff between usability and security, we try to focus on a proof of concept for the system. The goal is to demonstrate that cameras and video recording devices can be successfully defeated by using inverting images (the devices should not be able to change camera settings fast enough to capture visual information from the screen). In any practical authentication system, usability is of high concern, but the prototype may not strike the perfect balance.

Related Work

There has been some work done on trying to prevent camera recordings from taking place within controlled environments. Truong et al. discovered that a camera's image sensor (CCD) is retroreflective; this allows easy detection of a camera being present within a vicinity [3]. After locating a camera, a thin beam of visible white light or a laser could be focused on the CCD and disrupt filming. Naimark also describes how a camera could be easily neutralized with a simple consumer laser pointer [4]. However both of these approaches require additional hardware to detect cameras and project something to attempt to render the recording useless. We attempt to defeat cameras entirely in software.

For the purposes of this study we assume our adversary has access to consumer-grade cameras such as a commodity digital camera or a cell phone camera. We purposely omit high-end devices since they are rather expensive and would rouse suspicion if attempting to shoulder-surf with them as they are typically bulky. We take advantage of the fact that most consumer-grade cameras use automatic exposure to control the amount of light that reaches the camera's CCD. Automatic exposure's biggest weakness is that it is slow to adjust to sudden light changes since it requires electro-mechanical operations [5]. This will be our main tactic at combating cameras since we do not foresee a change away from this traditional automatic exposure any time soon.

System Description

The system is designed with both human and electronic shoulder-surfing attacks in mind. To defeat human attacks, the previously successful method of displaying images that correspond to number pad input is modeled in our system. To defeat electronic attacks, the system tries to obscure the visual information from the recording device by using inverting images.

When the user begins the authentication session a black screen is presented. The black screen is displayed for a relatively long time, usually between 1-5 seconds. Next, the screen switches to a white background, and a set of grayscale icons are presented to the user. This screen (with the valuable visual information) is presented for a very short time (hundreds of milliseconds) in comparison to the black screen. The icons then disappear and a blank white screen appears for several hundred more milliseconds. The black screen is then displayed again. The user, having seen the set of icons, enters the key associated with the position of the icon on the screen that is contained in the user's password. This sequence will repeat for the length of the graphical password. The amount of time spent displaying the black screen before switching to the white screen is always on the order of seconds, but the number of seconds is random. This helps prevent a shoulder-surfing attack with a camera; it is quite difficult to time depressing the shutter release button of a camera with the visual information presented onscreen. A camera with a rapid, consecutive shot mode is also unlikely to capture the correct frame and would also likely fill up its memory buffer quite rapidly. Also note that after the screen with the icons is displayed, a white blank screen is displayed for an extended period of time. This is to ensure that there will not be quick flickering of black screens to white screens to black screens which may cause problems for people with epilepsy.

Implementation Details

The prototype system uses no additional hardware and is a small standalone Java GUI software application. It has been developed and is used on a Dell Latitude laptop with a high resolution LCD. To keep implementation relatively simple, the password length is three grayscale icons. Icons are displayed in sets of 4 (a 2x2 grid) and maps to user input with the directional keys of a keyboard. The prototype will repeatedly display an icon set until the user is able to distinguish the password icon and input it with the keyboard. To allow tuning of the system, the amount of time the black and white screens are displayed is configurable as is the ‘darkness’ of the grayscale icons.

The original prototype easily defeats the camera and video recording functions of the Motorola V710 and Sony Ericsson Z525a cell phones. However, the video recording function of standard digital cameras (Canon PowerShot A610 and SD400) are more difficult to defeat. These cameras can adjust their settings ‘on-the-fly’ quite rapidly. The initial focus and settings lock (half depressing the shutter release button) does not seem to hinder the camera’s ability to quickly change settings during a video capture. Empirically we find that displaying the white screen (with the icon set) for around 300-400ms can defeat the camera in most cases. However, frame by frame analysis shows that a single frame of the white screen with icons is usually captured by the camera. Specifically, we find that the Java GUI application does not draw the entire screen in one frame when switching from the black screen to the white screen; there is usually a frame of video that captures the upper half of the screen drawn black, and the lower half of the screen drawn white (with half of the grayscale icons showing). Because the transition is not quite as dramatic (entirely black screen to entirely white screen) the camera is able to adjust its settings fast enough to discern the images on the lower half of the screen.

This is unacceptable to us, so we remedy this situation as follows; in between displaying the black screen and white screen (with icons), we insert a short amount of time where an entirely white screen (without icons) is displayed. However, this entirely white screen cannot be displayed for too long since the camera will be able to adjust its settings to the bright white screen and capture the grayscale icons in subsequent frames. After tweaking the timings, we are able to adjust the settings such that the entirely white screen need only be displayed for about one frame (or less than 50 milliseconds). In combination with adjusting the darkness of grayscale icons, we are now able to consistently defeat the video recording feature of our test cameras from a variety of zoom settings, shoulder-surfing locations, and levels of ambient light.

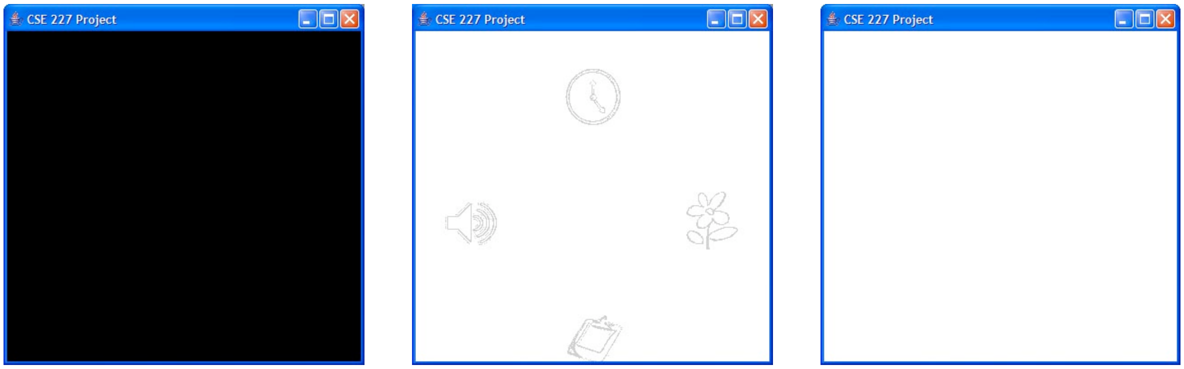

Figure 1. Screen shots of black screen, icons, and white screen.

Results & Analysis

To test our prototype, we used a Dell Latitude laptop as the authentication terminal. The aforementioned Canon A610 and SD400 cameras were used for video recording. These cameras have a relatively high quality 640x480, 30fps video recording mode and both feature the Digic II processor used in Canon's higher end pro digital SLR cameras. To analyze video, frame by frame analysis was completed in VirtualDub. Three test subjects completed our user study. The subjects were all new graduates (22-23 years old) with backgrounds in Computer Science, Mechanical Engineering, and Management Science. The subjects alternated being users of the authentication system and potential shoulder-surfers. A verbal description of the authentication system (along with their password) was given to the users, and a detailed hands-on walkthrough of the digital camera's video recording function was given to shoulder-surfers. Shoulder-surfers were told to orient themselves and the camera in whatever way was most beneficial to steal the password.

The first trial statically placed the grayscale icons onscreen and the user would enter the keys associated with their password icons. This was set up to crudely model the Passfaces interface and no image inverting was used. We found that with frame by frame analysis of the video, there was a 100% defeat rate of the static version of the authentication system. Attackers were easily able to capture both the visual information onscreen and the user input on keyboard with a single camera and analyze the video to discern the password.

The subsequent trials used our dynamic inverting image method of authentication. We found that our users authenticated themselves without error in every trial (100% input accuracy). When the UI was tuned to an 'easy setting' (the white screen with icons is displayed well over 400ms and the grayscale icons are quite dark), we found shoulder-surfers were able to pick up one out of every three icon sets displayed. However, the attackers had to fill the entire camera view with the screen to capture the icons, leaving nothing to capture the users input. On a slightly more difficult UI setting (the white screen with icons is displayed less than 300ms and the grayscale icons are lightened), we found that shoulder-surfers had a 0% defeat rate of the authentication system. Frame by frame analysis showed the camera was only able to capture a blank white screen. For this UI setting, our users were still able to authenticate themselves on every trial.

Our trials and results show that our system easily defeats cell phone recording devices and still picture captures of cell phones and cameras. Furthermore, when the system is tweaked correctly, it is also possible to consistently defeat the video recording function of most digital cameras. Our users are still able to authenticate themselves with high accuracy and relative ease. For our tests, the camera angle, initial camera settings, zoom, ambient light, and camera view of the system do not seem to affect the deterrence capabilities of our system.

Open Questions and Future Research

The results from our prototype are quite promising, but the implementation and testing of our authentication system brings a slew of open questions and research possibilities. To aim for completeness rather than a thorough discussion of each point, we present and discuss these concerns in bullet form:

- Usability vs. Security Tradeoffs
    - What is the appropriate length of graphical passwords?
    - Should the user be given multiple chances (display the same icon set repeatedly) until some keyboard input is provided?
    - How many icons should be displayed in a screen (how much graphical information can we expect the user to process)?
    - If the grid increases in size and a numerical pad is used, is 'chicken-pecking' a straight-line numerical entry significantly worse than a separate numerical pad (which is not available on laptops)?
    - Does the use of simple grayscale icons limit the password space?
        - Can color images/backgrounds be used as well in our system?
        - What is the best type of password icons (text, low quality icons, real images)?
    - Is our system usably for all age groups (elderly)? Those with vision problems? Those who might be agitated by inverting images?
        - What are the psychological effects of not being able to accurately authenticate oneself in our system or not being able to handle the rapid switching between images and grayscale icons?
    - What is the best tradeoff between deterrence and usability for the display time of the black screen and the white (with icons)? What is the shortest amount of time we can present the password icons while leaving the system usable?
- Device and Technology Trends
    - How will camera improvements defeat our system?
        - Improvements in megapixels? Optical zoom? CCD technology and light metering? Camera processor?
            - Anyone of these could potentially break the system, but will these improvements ever match the ability of the human visual system?
    - How do SLR cameras and high quality HD video recording cameras fare against our system? (Granted, a shoulder-surfer carrying a large high quality video recorder or camera would be quite noticeable)
    - How does the output display affect the system?
        - CRT vs. LCD? Screen size? Resolution? Brightness (cd/m^2)? Graphics card 2-D draw capabilities? Java vs. other GUI drawing system?
- System Environment
    - How much attention would this authentication system draw in a busy environment during an authentication session?
    - How does the amount of ambient light change the system's ability to deter devices?
        - Perhaps screen settings could change with the amount of ambient light perceived through a laptop's built in web camera.
            - If the room is dim, the system would work as described before (black screen that switches to a quick flash of a

white screen with grayscale icons). If the room is very bright, the system would switch (white screen that switches to a quick flash of a black screen with gray-scale icons).

- The web camera may also be able to detect the red-light produced by the AF-assist (Auto-Focus assist) feature of most cameras or even human eyes ‘peeping’ in the background.

- What if two cameras are present, each with different preset settings (one preset to bright scenes, another preset to dark scenes)?
- How does the camera view (the view is filled entirely by the authentication screen vs. the view is partially filled by the authentication screen and also contains the surrounding background) change the system’s ability to deter should-surfing?

- Applications
    - Where is this system most appropriate?
        - Probably too much work to log into an email or online forum account.
        - But might be tolerable for online financial transactions (banking websites), or even ATMs.
    - Can the general idea be used for other applications?
        - At a government authentication ‘booth’, bright lights may illuminate behind the user to prevent human and electronic shoulder-surfing behind the subject.

Conclusion

In this project we have demonstrated that a novel, but simple, tactic can be used to successfully defend against both human and electronic methods of shoulder-surfing. A rapid change of image brightness in software defeats most common cameras; the system has excellent results against cell phone cameras and current popular digital cameras. The ability of the human visual system to react to these types of image changes faster than cameras allows the system to retain its usability for the subjects of our user study. The proof of concept has been a success, but the ultimate utility of such a system requires a deeper investigation of various system, user, and environment parameters.

References

1. Xiaoyuan Suo, Ying Zhu, G. Scott. Owen, "Graphical Passwords: A Survey," acsac, pp. 463-472, 21st Annual Computer Security Applications Conference (ACSAC'05), 2005.
2. Furkan Tari, A. Ant Ozok, Stephen H. Holden, “A Comparison of Perceived and Real Shoulder-surfing Risks between Alphanumeric and Graphical Passwords,” Symposium on Usable Privacy and Security (SOUPS), 2006.
3. Khai N. Truong, Shwetak N. Patel, Jay W. Summet, Gregory D. Abowd,”Preventing Camera Recording by Designing a Capture Resistant Environment,” Ubicomp 2005, 2005.
4. Michael Naimark, “How to ZAP a Camera: Using Lasers to Temporarily Neutralize Camera Sensors,” http://www.naimark.net/projects/zap/howto.html, 2002.
5. Jim Stinson, “Camera Work: Letting in Some Light,” http://www.videomaker.com/article/1405/, 1995.