Decoy Applications for Continuous Authentication on Mobile Devices

Malek Ben Salem
Cyber Security Laboratory
Accenture Technology Labs
Arlington, VA, USA
malek.ben.salem@accenture.com

Jonathan Voris
Allure Security
Technology Inc.
New York, NY, USA
jon@alluresecurity.com

Salvatore J. Stolfo
Allure Security
Technology Inc.
New York, NY, USA
sal@alluresecurity.com

1. INTRODUCTION

Mobile devices and applications carry a great deal of sensitive and personally identifiable information, which makes them very lucrative targets for attackers. Authentication on these devices is vulnerable to smudge attacks [1]. Furthermore, their small size, light weight, and ubiquity makes them easily stolen. According to the Cloud Security Alliance, data loss from lost, stolen, or decommissioned mobile devices is the single largest threat to mobile computing [5]. The nature of user interaction with mobile devices calls for novel authentication approaches that are robust and secure, usable, and inexpensive. In a mobile context, security solutions must be flexible as well as resource efficient to ensure compatibility with a broad platform base.

We propose the use of *decoy apps* on mobile devices to continuously authenticate users once the user is logged in– *i.e.* throughout the user session– and to detect suspicious activity by a masquerader, or unauthorized user posing as the owner and legitimate user of the mobile device. Decoy apps are authentic-looking apps that hold fake but enticing information to the potential masquerader. They may be installed manually by the device owner or automatically through some app distribution and installation service. Once installed on the mobile device, their only function is to act as bait to the masquerader. They are not to be used by the device owner, and therefore any access to decoy apps is highly indicative of potential masquerade activity. We conjecture that decoy apps can be used to continuously authenticate users once logged-in to the mobile device throughout an entire user session. Access to any decoy app could be a trigger for de-authenticating the user. Furthermore, we posit that even if a masquerader were aware decoy apps are loaded on the device, they would lack the user's knowledge of which apps are real or decoys. Figure 1 displays a notional view of the conundrum faced by the attacker.

In this paper, we present an approach for deploying decoy apps to (de-)authenticate mobile device users. The remainder of this paper is organized as follows. First we will briefly describe how mobile decoy apps can be created, distributed and used for authentication in Section 2. Section 3 describes the usability of decoy apps as a continuous (de-)authentication and masquerade detection mechanism on mobile devices. We discuss the costs associated with their use and deployment in Section 4, and conclude the paper in Section 5.

Symposium on Usable Privacy and Security (SOUPS) 2014, July 9–11, 2014, Menlo Park, CA.

2. CONTINUOUS AUTHENTICATION USING DECOY APPS

Decoy applications can be generated in a variety of ways, allowing an organization to select the technique that best fits their operational requirements. One option is to program specific fake applications which contain spurious data and issue alerts when accessed. For example, decoy e-mail or banking applications could be planted on a device and seeded with realistic but inauthentic transaction information. Device owners would naturally avoid these applications as they know the authentic e-mail or banking app, but they would make prime targets for a curious adversary, issuing an alert to the true device owner in the process. This option produces believable applications, but is time consuming if many varied decoys are needed. As an alternative, seldom used applications can be transformed into decoys by injecting existing programs with "beaconing" functionality [3]. If an organization utilizes device client security monitoring software, another option is to leverage this platform to "tag" applications as decoys. These techniques scale much more easily, but require additional effort in terms of application monitoring and analysis. We conjecture that the use of decoy apps would be very effective in detecting any suspicious activity on the mobile device. Prior studies on the use of decoy files on desktops have shown that honeyfiles are very effective at detecting masquerade activities with a high accuracy and a low latency [2].

An important consideration of decoy design is how an application, or the device's operating system itself, should respond to signs of suspicious activity. To further increase the effectiveness of the decoy apps, it is important to include variability in the decoy apps that are deployed as well as in the fake information that they carry.

3. USABILITY

Besides ease of deployment, an important aspect of the usability of decoy apps is their associated error rate, in particular the false positive rate. A high false positive rate, preventing a user from accessing their mobile device when needed, may adversely affect the usability and adoption of

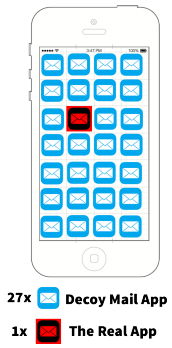

Figure 1: A Notional Decoy App Screen Layout

decoy apps as a continuous authentication mechanism. The right mitigation strategy when a decoy app gets accessed plays a prominent role in our research to reduce errors. Hence, we consider challenging the user in such situation, and incorporating other modalities for authentication whenever a user is challenged, such as image or voice verification, or perhaps swiping a digital pattern image using a mouse or touchscreen to add another layer of authentication. A 100 mobile phone user study is being planned that will provide insight into alternative mitigation strategies.

We note that every user is constantly being trained on mobile devices to be reminded about security: For instance, the phone locks after a few minutes of no user activity. Many people are accustomed to getting alerts from their credit card companies for any transactions that are suspicious. The important point is that even if an authentic user clicks on a decoy app, and is alerted, this will remind the user that they are protected and that their security protection works.

4. COST OF DECOY APPS

Mobile device usage has increased dramatically in recent years. During 2013, global mobile data traffic increased by 81% [4], and current estimates indicate that smartphones and tablets started to outnumber desktop computers in the workplace [6]. Considering these explosive growth figures, the costs of deploying security solutions in scale across an organization's network of mobile assets are particularly critical.

A core advantage of decoy applications is their cost effectiveness both with respect to infrastructural costs as well as resource costs at the device level. When coupled with a distribution service and/or mobile host sensor, decoy applications can be installed on a wide variety of devices with minimal user interaction or administrator involvement. Unlike cumbersome anti-virus and firewall systems, which require consistent upkeep and monitoring, decoy programs can be easily monitored for access, and their contents can be periodically refreshed with little transmission overhead.

Decoy apps require little computational power and consume small amounts of battery power, as they don't need to perform any work aside from triggering an alarm and mimicking typical application behavior. Decoys do not require much storage space either; A typical Android application only consumes several megabytes of storage capacity, for example. We conjecture that a very limited number of highly attractive and conspicuous decoy apps will be needed to detect an attacker's intrusion; our experiments using decoy files for masquerade detection on desktops have already shown that when placing 30 decoy files among thousands of authentic files, the probability of detecting an intruder within 10 minutes is at least 90% (at the 98% confidence level) [2].

5. CONCLUSION

Decoy applications are a natural (de-) authentication solution for mobile platforms when a phone is lost or stolen. They are easily composable with other mobile security mechanisms. Decoy programs incur little monitoring overhead. Generating them is efficient and flexible, and their resource-friendly nature is especially advantageous. In case of masquerade attacks, which should logically be rare events, an alert about a decoy app being touch is the only real way of knowing about the intrusion. We have planned an IRB-approved user study of mobile application usage in order to assess the efficacy of decoy applications in a realistic setting, to measure error rates, and to identify best practices for decoy application design, placement, and distribution.

6. ACKNOWLEDGMENTS

This material is based on work supported by the Defense Advanced Research Projects Agency (DARPA) through the Active Authentication II Program with contract award number FA8750-13-C-0259. The views expressed are those of the author(s) and do not reflect the official policy or position of the Department of Defense or the U.S. Government. Approved for Public Release, Distribution Unlimited.

7. REFERENCES

[1] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith. Smudge attacks on smartphone touch screens. In *Proceedings of the 4th USENIX Conference on Offensive Technologies*, WOOT'10, pages 1–7, Berkeley, CA, USA, 2010. USENIX Association.

[2] M. Ben-Salem and S. J. Stolfo. Decoy document deployment for effective masquerade attack detection. In *Proceedings of the Eighth Conference on Detection of Intrusions and Malware & Vulnerability Assessment*, DIMVA '11, pages 35–54, Heidelberg, July 2011. Springer.

[3] B. M. Bowen, M. Ben-Salem, S. Hershkop, A. D. Keromytis, and S. J. Stolfo. Designing host and network sensors to mitigate the insider threat. *IEEE Security & Privacy*, 7(6):22–29, Nov. 2009.

[4] Cisco Systems. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018. Available online at `http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html`, 2014.

[5] D. Hubbard, C. Garlati, F. Kasprzykowski, D. Lingenfelter, J.-M. Brook, A. Decker, E. Fisher, A. Lum, S. Michalove, G. Sanchidrian, S. Wilke, A. Alva, L. J. Santos, K. Scoboria, E. Scoboria, and J. Yeoh. Top Threats to Mobile Security, 2012.

[6] Mary Meeker and Liang Wu. 2013 Internet Trends available online at `http://www.kpcb.com/insights/2013-internet-trends`, 2013.