DETECTING APPLICATIONS WITH EXCESSIVE PRIVILEGES AND APPLICATIONS VULNERABLE TO PRIVILEGE ESCALATION ATTACK IN ANDROID

IMAN KASHEFI

A project report submitted in partial fulfillment of the
requirements for the award of the degree of
Master of Computer Science (Information Security)

Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia

JANUARY 2013

This project report is dedicated to my beloved mother, father, and sister who are the greatest assets in my life and also to Mariam for her endless support.

ACKNOWLEDGEMENT

First and foremost, I would like to express heartfelt gratitude to my supervisor **Associate Prof. Dr. Mazleena Salleh** for her constant support during my study at UTM. She inspired me greatly to work in this project. Her willingness to motivate me contributed tremendously to our project. I have learned a lot from her and I am fortunate to have her as my mentor and supervisor.

Besides, I would like to thank the authority of Universiti Teknologi Malaysia (UTM) for providing me with a good environment and facilities.

ABSTRACT

The rapid growth of capabilities and various services provided by smartphones transformed this device to a repository of private data and important resources and consequently an attractive target for attackers. Among the leaders in the world of smartphones, Android is a novel platform with rapidly growing market share. Number of Android users grows tremendously and preliminary study has shown that there are a number of the users that have little or no knowledge about the security of android based platforms. This is a serious issue because Android has delegated security decisions to the users themselves and furthermore there is no effective auditing on application development in android market. This research focuses on the most important attacks in Android which are concerned with the applications try to acquire excessive privileges by user approval, colluding together or even misusing other applications. The detection mechanisms proposed in this study addressed the mentioned attacks by proposing a method for detecting applications which are able to collude together to acquire excessive privileges and also a method to improve the precision of the existing mechanism for detecting applications vulnerable to be misused by privilege escalation attack. Excessive privileges are detected primarily by checking the application ability to share their permissions and then by comparing the acquired permissions against a set of predefined rules. Proposed mechanisms are integrated and implemented in form of an Android application by using Java (Android) language. The functionality of the implemented application is tested and validated by applying it on a series of applications downloaded from “Google play” and comparing the results with the existing methods. Experiments showed that the mechanism is able to detect applications vulnerable to privilege escalation attack accurately and also applications which are able to collude to obtain excessive permissions and were ignored by the existing methods.

ABSTRAK

Pertumbuhan pesat keupayaan dan perkhidmatan yang disediakan untuk telefon pintar menjadikan peralatan ini sebagai satu alat simpanan data peribadi dan sensitif. Ini sebaliknya telah membuatkan peralatan ini sebagai sasaran untuk penyerang. Pada masa kini, Android merupakan platform yang amat popular dalam pasaran yang berkembang pesat. Bilangan pengguna Android meningkat dengan pesat dan kajian awal telah menunjukkan bahawa terdapat beberapa bilangan pengguna mempunyai pengetahuan sedikit atau tiada langsung mengenai keselamatan platform berasaskan android. Ini merupakan satu isu yang serius kerana Android telah menyerahkan hal keselamatan telefon pintar kepada pengguna sendiri. Hal ini ditambahkan lagi apabila tiada audit yang berkesan untuk memperlihatkan pembangunan aplikasi dalam pasaran android. Sehubungan itu, kajian ini memfokuskan pada serangan yang serius dalam Android iaitu menitikberatkan aplikasi-aplikasi yang cuba memperolehi keistimewaan yang berlebihan dari keizinan pengguna, atau bersekutu sesama serta menyalahgunakan aplikasi lain. Mekanisma yang dicadangkan memperkenalkan metod untuk mengesan aplikasi di mana ia mampu untuk mengesan aplikasi yang ingin mendapat keistimewaan yang berlebihan dari sepatutnya dan juga metod yang akan meningkatkan ketepatan mekanisma sedia ada untuk aplikasi penegesanan kelemahan dari disalah guna untuk serangan peningkatan keistimewaan. Ini dilaksanakan dengan mengesan dengan menyemak kebolehkongsian keizinan dan membandingkan keizinan yang diperolehi terhadap satu set peraturan-peraturan yang telah ditetapkan. Mekanisma yang dicadangkan adalah bersepadu dan dilaksanakan dalam bentuk aplikasi android dengan menggunakan bahasa pengaturcaraan Java (Android). Fungsi apliksai yang dilaksanakan telah diuji dan disahkan dengan menjalankan pegujian semakan terhadap siri-siri aplikasi yang dimuat turun dari “Google Play” dan membandingkan keputusan yang terhasil dengan keputusan ujian metod yang sedia ada. Keputusan menunjukkan mekanisma cadangan berjaya mengesan dengan tepat aplikasi yang mempunyai kelemahan terhadap serangan peningkatan keistimewaan. Tambahan metod yang dicadang juga dapat mengesan aplikasi yang kebolehan untuk berpakat bagi mendapat kelebihan kebenaran, yang mana pada asalnya telah terlepas pandangan oleh method yang sedia ada.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AAA	-	Authentication, Authorization, and Accounting
API	-	Application Programming Interface
Apk	-	Android PacKage
Apps	-	Applications
AVD	-	Android Virtual Device
Dex	-	Dalvik Executable
DV	-	Virtual Machine
DVM	-	Dalvik Virtual Machine
ICC	-	Inter-Component Communication
IDE	-	Integrated Development Environment
IPC	-	Inter-Process Communication
MMS	-	Multimedia Messaging Service
MMU	-	Memory Management Unit
OS	-	Operating System
POSIX	-	Portable Operating System Interface
RPC	-	Remote Procedure Call
SDK	-	Software Development Kit
SIM	-	Subscriber Identity Module
SMS	-	Short Message Service
UID	-	User ID
URI	-	Uniform Resource Identifier

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

The rapid growth of capabilities and services that associated with smartphones has motivated enterprise to investigate toward these new generation platforms. Portability of traditional smart phone is integrated with the computational power of personal computers and the result is the everywhere and every time services such as mobile-health, mobile-banking, mobile-shopping, and social network services. With the tremendous growth in using smartphones, the security risk and attacks changed to a great concern [1, 2], the portability features of these devices make them the user's closest assistant even more than a PC or notebook, so the amount of data, and the level of importance of this data make them an attractive target for attackers while a great threat to customers [3].

Despite such advancement in application and services smartphones offer, the privacy of user sensitive data to third-party application is still a point of concern. Applications are granted privileges legitimately for accessing to user sensitive information, but they may use the user data in an improper way. Coming to a reasonable trade-off between functionality of running third-party application and maintaining user privacy is a significant challenge in smartphones platforms [4]. For instance, if a user install an application that have access to user's location information, he/she is not sure whether the data is being used in a proper way or the

application sends it to remote server for advertising reasons or other malicious purpose. In other words, users blindly trust that application and suppose that the application use them properly. Unfortunately recent researches [4, 5, 6, 7, 8] showed that currently there are various application with different malicious purpose uploaded in market, and users are attracted by their splendid advertisement. These applications have been developed with malicious purpose such as leaking user sensitive information [9].

Google Android is a novel smartphone platform with rapidly growing market share [10] and also could possess the first rank in mass-production of application development [3]. According to a recent report released by mobile security firm Lookout, the Android Market is growing at three times the rate of Apple's App store [11]. Dissimilar to Apple, Google has no mechanism in auditing application published in market. And from time to time, it may need to remove malicious application from the market after they are proved to contain malware [11]. Moreover, since everyone who has registered as Android developer has permission to upload his/her application to Android market, it changed to a potential place for attackers to get to their malicious intentions [12].

In such situation security concerns grows with the same rate of increasing Android applications and users, therefore, the need of achieving a higher level of security in Android platform has been arisen more than ever.

1.2 Problem Background

In brief, Android is basically a privilege-separated operating system [13, 11]. Android places each application in its own Dalvik virtual machine and they run with a unique system identity. In this way all applications are isolated from each other and from the system. In normal state, since applications run in separated virtual machine and with respect to specific Android security mechanism, they have no way

to negatively affect other applications, damage operating system, or cause leaking sensitive information stored on the phone [14]. While access of applications to each other components is controlled through Android specific security mechanism, granting Android built-in permissions to new installed applications and giving access to Android components takes place based on user discretion. To be able to do any actions, applications should ask for any required permissions at install time to be approved by user.

However, Android integrates many security mechanisms along with the important issue that the Android kernel is developed based on Linux which has a robust security infrastructure, yet the occurrence of attacks reveals that Android's permission framework has some vulnerability which are targeted by attackers. The most common attacks are caused by misusing of critical permissions that are approved by users and also application-level privilege escalation [12]. The major responsibility of maintaining the security of the device in right level is left to end-user and most of them are unaware of critical security issues that not respecting them may cause leaking out the user sensitive information or misusing of the device.

Requesting excessive permissions by an application and granting them by an unaware user violates the principle of "least privilege" and gives the malicious application the opportunity of taking advantages of extra privileges that are not needed for its normal functionality. Request for acquiring excessive permissions by applications is highly probable since Android has no effective audit on application developers; moreover, the majority of users do not have the knowledge of discerning the minimum necessary permissions needed by an App from extra permissions it may request. So this has become the main concern in the recent Android attacks. For example by installing every application without enough precision, a door for malware and Trojans such as unauthorized sending of text messages [15], malicious game updates [16], or location tracking and leaking of sensitive data in the background of running games can be opened [17].

Recent research showed that a malicious application can exploit an unprotected privileged application or seizes more permission to do harmful actions, i.e., a non-privileged caller with few permissions is not bounded to access component of an application with more privileges [18, 11]. Such attacks are known as privilege escalation attacks. In case of success of attack, a malicious application seizes more permission indirectly and through another application. Therefore an unprivileged application can perform its malicious intentions by employ other applications which have the supposed permissions. The attacks reported so far range from unauthorized phone calls [19] and text message sending [18] to illegal downloads of malicious files [20] and context-aware voice recording [21, 22]. Most attacks of this kind target privileged applications with vulnerable interface [23, 24]. However it is most seen that a group of malicious applications can collude and gain more permissions by accumulating their privilege. This attack is referred to as collusion attack [21]. In this way malicious applications benefit from a set of permissions which empowers them to perform malicious actions. Below are some instances of probable malicious actions [25]:

(i) Sending SMS/MMS to contact or anonymous. This kind of attack causes overcharging the user.
(ii) Depleting the phone battery by performing unnecessary process to interrupt normal services.
(iii) Communicating to charged site or pay per-minute telephone number and overcharge the user.
(iv) Installing malware code such as worms and viruses and disturbing the normal function of the device and spreading them to other devices through Bluetooth or other possible ways.
(v) Illegally accessing to user sensitive information, altering, or deleting them.
(vi) Leaking the user personal information out for malicious purpose.

In order to detecting and defeating these attacks and mitigate the consequent malicious actions, several researches have been conducted which are mostly propose some modification to Android OS to mitigate the shortcomings of the system. Most

proposed solutions need modifications to Android's middleware, the main middleware components such as the application installer, the reference monitor, the permission database, and the Dalvik virtual machine. While some others focus on proposing detection mechanisms to prevent suspicious and vulnerable applications from installing or make Android users aware of these threats in order to help them make reasonable decisions about granting critical permissions to applications which is an important preventing mechanism due to the significant role of the users in maintaining security in Android.

1.3 Problem Statement

Critical privileges acquired by applications through user approval, sharing permissions or using unprotected applications may cause serious attacks to users' privacy and sensitive information in Android devices. Therefore, a detecting system which addresses these attacks is necessary to make users aware of the possibility of such attacks in order to protect their information and privacy from malicious applications.

1.4 Objectives of the Project

The objectives of this project are listed as follows:

(i) To identify attacks result from applications with critical privileges and also the existing detection mechanism in Android smartphones.

(ii) To enhance the capabilities of existing detection mechanisms by designing and implementing an application for detecting applications that are potentially capable of conducting malicious activities or even are vulnerable to be misused by malicious applications.

(iii) To test and validate the functionality of implemented application in terms of enhancement in detection mechanism by applying it on a set of applications downloaded from the official Android market called “Google Play”.

1.5 Scope of the Project

This research investigates the shortcomings in Android permission system and attacks result from that in order to develop a useful application to detect applications that are potentially capable of conducting malicious activities solely or by colluding and also applications that are vulnerable to be misused by malicious applications. The scope of this study is defined as follows:

(i) Attacks results from applications with excessive privileges are considered; attacks result from granting extra permissions to a single application or a group of application that are capable of colluding together and sharing their privileges, and attacks result from unprotected components in applications.

(ii) The programming language used for developing the proposed application is Java(Android).

(iii) The application is applied on the number of applications obtained from “Google Play” in order to test and validate.

1.6 Significance of the Study

The outcome of this project is improving the security level in Android phones by increasing the user awareness of probable attacks and helping them making reasonable decisions regarding granting permissions to the applications, due to the

fact that users have the most significant role in securing Android phones and unfortunately most of them know little about security.

1.7 Organization of the Report

This study is divided into six chapters. Chapter 1 describes briefly about the overview of the project and understanding of the project's problem background. It also includes the project's scope, purpose of this research and objectives. Chapter 2 discusses about Android architecture, Android security mechanisms, Android application package and related works of this study. The methodology of this research is explained in detail in Chapter 3.

Chapter 4 contains explanations of design and implementation of this study. Chapter 5 explains results of this research based on applying the implemented application on a series of applications downloaded from "Google play" and comparing the results with the existing methods. Finally, Chapter 6 reviews and summarizes the whole project findings and suggests some recommendations.

REFERENCES

1. Cheng, J., H.Y. Wong, S., Yang, H., and Lu., S. (2007). SmartSiren: Virus Detection and Alert for Smartphones. *MobiSys '07: Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*. 11-13 June. New York, USA: ACM, 258-271.
2. Racic, R., Ma, D., and Chen, H. (2006). Exploiting MMS Vulnerabilities to Stealthily Exhaust Mobile Phone's Battery. *Proceedings of 2nd International Conference on Security and Privacy in Communication Networks*. August. Baltimore, MD, USA: IEEE, 1-10.
3. Nauman, M., Khan, S., and Zhang, X. (2010). Apex: Extending Android Permission Model and Enforcement with User-Defined Runtime Constraints. *Proceedings of the 5th ACM Symposium on Information.* 02 May. New York, USA: ACM, 328–332.
4. Enck, W., Gilbert, P., Chun, B.G., Cox, L.P., Jung, J., McDaniel, P., and Sheth, A.N. (2010). TaintDroid: an Information-Flow Tracking System for Real-time Privacy Monitoring on Smartphones. *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. October. Vancouver, BC, Canada: OSDI'10, Article No. 1-6.
5. Egele, M., Kruegel, C., Kirda, E., and Vigna, G. (2011). PiOS: Detecting Privacy Leaks in iOS Applications. *Proceedings of 18th Annual Network and Distributed System Security Symposium*. 6 – 9 February. San Diego, CA, USA: NDSS.
6. Bradley, T. (2011, March 2). DroidDream Becomes Android Market Nightmare. PCWorld, Retrieved December 15, 2012, from http://www.pcworld.com/businesscenter/article/221247/droiddream_becomes_android_market_nightmare.html

7. Thurm, S. and Kane, Y.I. (2010, December 17). Your Apps Are Watching You. The Wall Street Journal, Retrieved December 15, 2012, from http://online.wsj.com/article/SB10001424052748704694004576020083703574602.html
8. Mahaffey, K., Hering, J. (2010). App Attack: Surviving the Explosive Growth of Mobile Apps.
9. Zhou, Y., Zhang, X., Jiang, X., and Freeh, V.W. (2011). Taming Information-Stealing Smartphone Applications (on Android). *Proceedings of the 4th International Conference on Trust and Trustworthy Computing.* 22-24 June. Pittsburgh, PA, USA, TRUST: 93-107.
10. Egham. (2011, May 19). Gartner Says 428 Million Mobile Communication Devices Sold Worldwide in First Quarter 2011, a 19 Percent Increase Year-on-Year. Gartner Inc., Retrieved December 15, 2012, from http://www.gartner.com/it/page.jsp?id=1689814
11. Chan, P.P.F, Hui, L.C.K, and Yiu, S.M. A Privilege Escalation Vulnerability Checking System for Android Applications. (2011). *Proceedings of 2011 IEEE 13th International Conference on Communication Technology*. 25-28 September. Jinan, China: IEEE, 681-686.
12. Bugiel, S., Davi, L., Dmitrienko, A., Fischer, T., and Sadeghi, A.R. (2011). Xmandroid: A new Android Evolution to Mitigate Privilege Escalation Attacks. Tecnical report, TR-2011-04, Technische Universität Darmstadt, Darmstadt, Germany.
13. Security and permissions. Android Open Source Project, Retrieved December 15, 2012, from http://developer.android.com/guide/topics/security/security.html
14. Shabtai, A., Fledel, Y., Kanonov, U., Elovici, Y., Dolev, S., and Glezer, C. (2010). Google android: A Comprehensive Security Assessment. *IEEE Security and Privacy*. 8 (2), 35–44.
15. Leyden, J. (2010, August 10). First SMS Trojan for Android Is in the Wild. Theregister, Retrieved December 15, 2012, from http://www.theregister.co.uk/2010/08/10/android_sms_trojan/
16. Goodin, D. (2010, September 20). Android Bugs Let Attackers Install Malware without Warning. Theregister, Retrieved December 15, 2012, from http://www.theregister.co.uk/2011/09/20/google_android_vulnerability_patching/

17. Wyatt, T. (2010, December 29). Security Alert: Geinimi, Sophisticated new Android Trojan Found in Wild. Lookout Mobile Security, Retrieved December 15, 2012, from http://blog.mylookout.com/2010/12/geinimi_trojan/
18. Davi, L., Dmitrienko, A., Sadeghi, A.R., and Winandy, M. (2010). Privilege Escalation Attacks on Android. *Proceedings of the 13th International Conference on Information Security*. 25-28 October. Boca Raton, FL, USA, ISC: 346–360.
19. Enck, W., Ongtang, M., and McDaniel, P. (2008). Mitigating Android Software Misuse Before it Happens. Technical Report, NAS-TR-0094. Pennsylvania State University, USA.
20. Lineberry, A., Richardson, D. L., and Wyatt, T. (2010). These Aren't the Permissions You're Looking for. BlackHat USA 2010, Retrieved May 9, 2012, from http://dtors.files.wordpress.com/2010/08/blackhat-2010-slides.pdf
21. Schlegel R., Zhang, K., Zhou, X., Intwala, M., Kapadia, A., and Wang, X. (2011). Soundcomber: A Stealthy and Context-Aware Sound Trojan for Smartphones. *Proceedings of the 18th Annual Network and Distributed System Security Symposium*. 6-9 February. San Diego, CA, USA, NDSS: 17-33.
22. Kameir, R., Dalton, W., and Laird, J. (2011, Janurary 20). Android Trojan Captures Credit Card Details. ITProPortal, Retrieved December 15, 2012, from http://www.thinq.co.uk/2011/1/20/android-trojan-captures-credit-card-details/
23. Hardy, N. (1988). The Confused Deputy: (or Why Capabilities Might Have Been Invented). *ACM SIGOPS Operating Systems Review*. 22 (4), 36-38.
24. Dietz, M., Shekhar, S., Pisetsky, Y., Shu, A., and Wallach, D. S. (2011). Quire: Lightweight Provenance for Smartphone Operating Systems. *Proceedings of 20th USENIX Security Symposium*. 8 -12 August. San Francisco, CA, USA, USENIX.
25. Seung-Hee, O., Deok-Gyu, L., and Jongwook, H. (2008). The Mechanism and Requirements for Detecting the Cross-Service Attack. *Proceedings of* 3rd *IEEE Asia-Pacific Services Computing Conference*. 9-12 December. Yilan, Taiwan: APSCC, 656-661.
26. Android Security Overview, Security and Permissions. (2012). Retrieved December 15, 2012, from http://source.android.com/tech/security/#android-application-security

27. Shabtai, A., Fledel, Y., Kanonov, U., Elovici, Y., and Dolev,S. (2009). Google Android: A state-of-the-Art Review of Security Mechanisms. CoRR, abs/0912.5101.
28. Bugiel, S., Davi, L., Dmitrienko, A., Fischer, T., Sadeghi, A.R., and Shastry, B. (2012). Towards Taming Privilege-Escalation Attacks on Android. *Proceedings of 19th Annual Network & Distributed System Security Symposium.* 5-8 February. San Diego, California, USA, NDSS.
29. Enck, W., Ongtang, M., and McDaniel, P. (2009). Understanding Android Security. *IEEE Security & Privacy Magazine*. 7(1), 10-17.
30. Ongtang, M., McLaughlin, S., Enck, W., and McDaniel, P. (2009). Semantically Rich Application-Centric Security in Android. *Proceedings of Proceedings of the 2009 Annual Computer Security Applications Conference*. December. Washington, DC, USA, ACSAC: pages 340-349.
31. Conti, M., Nguyen, V.T.N, and Crispo, B. (2010). Crepe: Context Related Policy Enforcement for Android. *Proceedings of the 13th International Conference on Information Security*. 25 – 28 October. Boca Raton, FL, USA, ISC: 331– 345.
32. Beresford, A.R., Rice, A., and Skehin, N. (2011). MockDroid: Trading Privacy for Application Functionality on Smartphones. *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications, ser.* 1-2 March. New York, NY, USA: ACM, 49-54.
33. Hering, J., Mahaffey, K., and Burgess, J. (2010, 27 July). Introducing the App Genome Project. LOOKOUT, Retrieved December 15, 2012, from http://blog.mylookout.com/2010/07/introducing-the-app-genome-project/
34. Calo, R., Young, R., Smith, A., and Gelman, L. (2010). WhatApp. Retrieved May 6, 2012, from http://www.whatapp.org
35. Shabtai, A., Fledel, Y., and Elovici, Y. (2010). Securing Android-powered mobile devices using SELinux. *IEEESecurity and Privacy Magazine*. 8(3), 36-44.
36. Hornyack, P., Han, S., Jung, J., Schechter, S., and Wetherall, D. (2011). These Aren't the Droids You're Looking for: retrofitting android to protect data from Imperious Applications. *Proceedings of the 18th ACM Conference on Computer and Communications Security*. 17 – 21 October. New York, USA: ACM, 639-652.

37. Ongtang, M., Butler, K., and McDaniel, P. (2010). Porscha: Policy Oriented Secure Content Handling in Android. *Proceedings of the 26th Annual Computer Security Applications Conference*. 6-10 December. New York, NY, USA, ACM: 221-230.
38. Chin, E., Felt, A.P., Greenwood, K., and Wagner, D. (2011). Analyzing Inter-Application Communication in Android. *Proceedings of 9th Annual International Conference on Mobile Systems, Applications, and Services*. 28 June – 1 July. New York, NY, USA: ACM, 239-252.
39. Enck, W., Ongtang, M., and McDaniel, P. (2009). On Lightweight Mobile Phone Application. *Proceedings of the 16th ACM conference on Computer and Communications Security*. 9-13 November. Chicago, IL, ACM: 235 – 245.
40. Enck, W., Octeau, D., McDaniel, P., and Chaudhuri, S. (2011). A Study of Android Application Security. *Proceedings of the 20th USENIX Conference on Security*. 8-12 August. San Francisco, CA, USA, SEC: 21–21.
41. Felt, A., Chin, E., Hanna, S., Song, D., and Wagner, D. (2011). Android Permissions Demystified. *Proceedings of the 18th ACM conference on Computer and Communications Security*. 17 – 21 October. New York, USA, ACM: 627–638.
42. Portokalidis, G., Homburg, P., Anagnostakis, K., and Bos, H. (2010). Paranoid Android: versatile protection for smartphones. *Proceedings of the 26th Annual Computer Security Applications Conference*. 6-10 December. NY, USA, ACM: 347-356.
43. Felt, A. and Evnas, D. (2008). Privacy protection for social networking platforms. *Proceedings of the Workshop on Web 2.0 Security and Privacy*. 22 May. Oakland, CA, USA.
44. Au, K., Zhou, Y., Huang, Z., Gill, P., and Lie, D. (2011). Short paper: a Look at Smartphone Permission Models. *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, pages. 17- 21 October. Chicago, IL, USA, ACM: 63–68.
45. Vidas, T., Christin, N., and Cranor, L. (2011). Curbing Android Permission Creep. *Proceedings of Web 2.0 Security and Privacy Workshop*. 26 May. Oakland, California, USA, W2SP: 2.