Handout 2
Articles for Oral Reports in CompSci 725

v0.9: first published version, 12 July 2012
v0.91: fixed URL in [20], 19 July 2012
v1.0: articles identified by surname of first author and year
v1.01: removed empty bibitem [1]

Giovanni Russello
Clark Thomborson

August 8, 2012

A large fraction (15%) of your marks in CompSci 725 will be awarded for your performance of an oral presentation during a lecture period, during weeks 6 to 12 of the semester. Your presentation will be based on your careful reading and analysis of a professional publication that appears on the list at the end of this handout.

Students will deliver their oral presentations according to a randomly-selected "Order" that is determined at random, by the instructor, at the end of the first week of lectures.

Students with low Order numbers will present before students with higher numbers. Lower-numbered students also have a better chance of being able to present on their first-choice article.

During the first week of lectures, you should be deciding which of the (approximately) two dozen articles on the reference list of this handout you'd like to use as the basis for your oral presentation. You should also make some backup choices, in case you draw a high number.

Deadline: 5pm Monday 23 July : All students should send an email to Clark (`mailto:cthombor@cs.auckland.ac.nz`), indicating their first, second, and third choice of article for their oral presentation. Articles on this year's reading list can be uniquely identified by the surname of the first author and the year of publication. So, for example, your preference list might be "1. Armbrust 2010; 2. Beresford 2011; 3. Bletsch 2012".

Clark will digest your emails on Tuesday, and before lecture on Wednesday 25 July he will webpost the first draft of an "Articles to be Presented" handout.

Your attendance at lecture on Wednesday 25 July will be very important, because on that day we will be finalising the "Articles to be Presented" using the algorithm specified below.

Further information on the oral presentations and term papers will be supplied later in the term. You will also be given some tuition – we understand that, for many of you, this will be your first experience at constructing, and delivering, an oral presentation on a technical subject. We also understand that English is not the native language for many of our students, and we will not be marking you on the fine points of English grammar or spelling. However we do insist that technical words be spelled and used correctly in your oral and written reports. Your technical content must be clearly understandable.

1 Algorithm for Assigning Students to Articles

Please don't worry if you don't understand the algorithm, it's not examinable, and I'll be explaining it as we go!

1.1 Before the first round

All articles are in "Category 0". Formally, $\forall y : Category(y) \equiv 0$.

Category 0	no one assigned to this article.
Category 1	1 student is assigned to this article.
Category 2	2 students are assigned to this article.
Category 3	3 students are assigned to this article.

All students are "Type 0". Formally, $\forall x : Type(x) \equiv 0$.

Type 0	student is not assigned to any article.
Type 1	student is assigned to a category 1 article.
Type 2	student is assigned to a category 2 article.
Type 3	student is assigned to a category 3 article.

All students x have been assigned a unique integer Order(x) in the range $1..N$, where N is the number of students enrolled. The ordering integer determines a student's priority for article selection, and it also defines the order in which they will present their selected article to the class.

The maximum number of articles M has been fixed by the instructors at approximately $0.4N$, so that most articles will have three presenters. For example, if there are $N = 36$ students enrolled, $M = 16$.

Note 1: A student x who enrols at any time after the Order() is fixed will be assigned an ordering integer which is larger than that of any other student in the class.

Note 2: Articles may be added to the recommended list at any time by the instructors. Suggestions from students are welcome, however the suggested article must meet with the instructors' approval.

1.2 Round 1

Students send an email, as defined in the first section of this handout, to Clark (`mailto:cthombor@cs.auckland.ac.nz`), indicating their first, second, and third choice of article for their oral presentation.

For each article y on the list of suggested articles, let $X(y)$ be the set of students who want to present this article. Clark will perform the following computational steps.

```
foreach y in List
  if (|{z: Type(z) > 0}| < M)
    x1 = argmin( Order(X(y)) ); Article(x1) = y;
    Type(x1) = 1; Category(y) = 1;
  endif
  if (|{z: Type(z) > 0}| <= M) and (X>1)
    x2 = argmin( Order(X(y)/x1) ); Article(x2) = y;
    Type(x2) = 2; Category(y) = 2;
  endif
  if (|{z: Type(z) > 0}| <= M) and (X>2)
    x3 = argmin( Order(X(y)/{x1,x2}) ); Article(x3) = y;
    Type(x3) = 3; Category(y) = 3;
  endif
endfor
```

Postconditions for Round 1:

1. $\forall x : \text{Type}(x) \in \{0, 1, 2, 3\}$
2. $\forall p : \text{Category}(p) \in \{0, 1, 2, 3\}$
3. No more than M different articles will be presented: $|\{p : \text{Category}(p) > 0\}| \leq M$

1.3 Round 2: In class, Wednesday 25 July

Each student x, in assigned Order starting with student #1, must choose one of the following actions:

1. If student x has made a selection (i.e. if $\text{Type}(x) \equiv 1$), they may "hold" this article.
2. Student x may select any article y in category 1 or 2. Do:

   ```
   Category(y)++; Article(x) = y; Type(x) = Category(y);
   ```

3. If M articles haven't already been selected $((|\{p : \text{Category} > 0\}| < M))$, then student x may select any article y in category 0. Do:

```
Category(y) = 1; Article(x) = y; Type(x) = 1;
```

Postcondition: all students who attended this lecture have chosen an article.

1.4 Round 3: In class, Monday 30 July

Students are allowed to "swap" their article with other students, in a controlled fashion. Also any students who haven't chosen an article must do so.

Each student x, in assigned order starting with student #1, must choose one of the following actions.

1. If (Type(x) $\equiv 0$), this student must select a article y in this round, either by choosing one of the Round-2 actions listed above or by choosing one of the actions below.

2. Student x may select a article y in Category 3, but *only if* one of the students z who is currently assigned to y is

   (a) willing to move to a different article w in category 1 or 2, or

   (b) if fewer than M articles have been selected and student z is willing to move to a article w in category 0.

Note 1: if more than one student is willing to move from y, then the student with the lowest number is the "volunteer" z.

Note 2: once a student (even one with the highest order #) is assigned a article, they cannot be forced to "move" to a different article.

Postconditions for Rounds > 2:

1. All registered students must present an article: $\forall x : \text{Type}(x) \in \{1, 2, 3\}$.

2. No more than M articles will be presented: $|\{p : \text{Category}(p) > 0\}| \leq M$

Termination Condition: Rounds will continue (to a maximum of 10) until a fixed-point is reached, i.e. until a Round makes no changes to the Article() assignments.

References

[1] J. Bellamy-McIntyre, C. Luterroth, and G. Weber, "OpenID and the enterprise: A model-based analysis of single sign-on authentication," in *Enterprise Distributed Object Computing Conference (EDOC)*, 2011, pp. 129–138. [Online]. Available: http://ieeexplore.ieee.org.ezproxy.auckland.ac.nz/stamp/stamp.jsp?tp=&arnumber=6037567

Abstract. Single sign-on (SSO) protocols allow one person to use the same login credentials for several organizations. Enterprises face increasing competitive pressure to position themselves with regard to SSO, yet the ramifications of a move to SSO are not fully understood. In this paper we discuss OpenID, a relatively new SSO protocol that is gaining traction on the web. We apply enterprise application modelling techniques to OpenID in order to obtain well-founded decision aids for enterprises: we show how published modelling approaches can be used to analyse risks in OpenID, and show that these can identify security problems with common OpenID practice. Finally, we propose analysis principles that condense important general insights of authentication modelling.

[2] A. R. Beresford, A. Rice, N. Skehin, and R. Sohan, "MockDroid: trading privacy for application functionality on smartphones," in *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '11. New York, NY, USA: ACM, 2011, pp. 49–54. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1145/2184489.2184500

Abstract. MockDroid is a modified version of the Android operating system which allows a user to 'mock' an application's access to a resource. This resource is subsequently reported as empty or unavailable whenever the application requests access. This approach allows users to revoke access to particular resources at run-time, encouraging users to consider the trade-off between functionality and the disclosure of personal information whilst they use an application. Existing applications continue to work on MockDroid, possibly with reduced functionality, since existing applications are already written to tolerate resource failure, such as network unavailability or lack of a GPS signal. We demonstrate the practicality of our approach by successfully running a random sample of twenty-three popular applications from the Android Market.

[3] T. Bletsch, X. Jiang, and V. Freeh, "Mitigating code-reuse attacks with control-flow locking," in *Proceedings of the 27th Annual Computer Security Applications Conference*, ser. ACSAC '11. New York, NY, USA: ACM, 2011, pp. 353–362. [Online]. Available: http://doi.acm.org.ezproxy.auckland.ac.nz/10.1145/2076732.2076783

Abstract. Code-reuse attacks are software exploits in which an attacker directs control flow through existing code with

a malicious result. One such technique, return-oriented programming, is based on "gadgets"(short pre-existing sequences of code ending in a ret instruction) being executed in arbitrary order as a result of a stack corruption exploit. Many existing codereuse defenses have relied upon a particular attribute of the attack in question (e.g., the frequency of ret instructions in a return-oriented attack), which leads to an incomplete protection, while a smaller number of efforts in protecting all exploitable control flow transfers suffer from limited deployability due to high performance overhead. In this paper, we present a novel cost-effective defense technique called con- trol flow locking, which allows for effective enforcement of control flow integrity with a small performance overhead. Specifically, instead of immediately determining whether a control flow violation happens before the control flow transfer takes place, control flow locking lazily detects the violation after the transfer. To still restrict attackers' capability, our scheme guarantees that the deviation of the normal control flow graph will only occur at most once. Further, our scheme ensures that this deviation cannot be used to craft a malicious system call, which denies any potential gains an attacker might obtain from what is permitted in the threat model. We have developed a proof-of-concept prototype in Linux and our evaluation demonstrates desirable effectiveness and competitive performance overhead with existing techniques. In several benchmarks, our scheme is able to achieve significant gains.

[4] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, A.-R. Sadeghi, and B. Shastry, "Towards taming privilege-escalation attacks on Android," in *19th Annual Network & Distributed System Security Symposium (NDSS 2012)*. The Internet Society, 2012. [Online]. Available: http://www.internetsociety.org/sites/default/files/07_3.pdf

Abstract. Android's security framework has been an appealing subject of research in the last few years. Android has been shown to be vulnerable to application-level privilege escalation attacks, such as confused deputy attacks, and more recently, attacks by colluding applications. While most of the proposed approaches aim at solving confused deputy attacks, there is still no solution that simultaneously addresses collusion attacks. In this paper, we investigate the problem of designing and implementing a practical security framework for Android to protect against confused deputy and collusion attacks. We realize that defeating collusion attacks calls for a rather system-centric solution as opposed to application-dependent policy enforcement. To support our design decisions, we conduct a heuristic analysis of Android's system behavior (with popular apps) to identify attack patterns, classify different adversary models, and point out the challenges to be tackled. Then we propose a solution for a system-centric and policy-driven runtime monitoring of communication channels between applications at multiple layers: 1) at the middleware we control IPCs between applications and indirect communication via Android system components. Moreover, inspired by the approach in QUIRE, we establish semantic links between IPCs and enable the reference monitor to verify the call-chain; 2) at the kernel level we realize mandatory access control on the file system (including Unix domain sockets) and local Internet sockets. To allow for runtime, dynamic low-level policy enforcement, we provide a callback channel between the kernel and the middleware. Finally, we evaluate the efficiency and effectiveness of our framework on known confused deputy and collusion attacks, and discuss future directions.

[5] S. Bugiel, L. Davi, A. Dmitrienko, S. Heuser, A.-R. Sadeghi, and B. Shastry, "Practical and lightweight domain isolation on Android," in *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, ser. SPSM '11. New York, NY, USA: ACM, 2011, pp. 51–62. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1145/2046614.2046624

Abstract: In this paper, we introduce a security framework for practical and lightweight domain isolation on Android to mitigate unauthorized data access and communication among applications of different trust levels (e.g., private and corporate). We present the design and implementation of our framework, TrustDroid, which in contrast to existing solutions enables isolation at different layers of the Android software stack: (1) at the middleware layer to prevent inter-domain application communication and data access, (2) at the kernel layer to enforce mandatory access control on the file system and on Inter-Process Communication (IPC) channels, and (3) at the network layer to mediate network traffic. For instance, (3) allows network data to be only read by a particular domain, or enables basic context-based policies such as preventing Internet access by untrusted applications

while an employee is connected to the company's network. Our approach accurately addresses the demands of the business world, namely to isolate data and applications of different trust levels in a practical and lightweight way. Moreover, our solution is the first leveraging mandatory access control with TOMOYO Linux on a real Android device (Nexus One). Our evaluation demonstrates that TrustDroid only adds a negligible overhead, and in contrast to contemporary full virtualization, only minimally affects the battery's life-time.

[6] M. Conti, V. T. N. Nguyen, and B. Crispo, "CRePE: Context-related policy enforcement for Android," in *Information Security - 13th International Conference*, ser. Lecture Notes in Computer Science, vol. 6531. Springer, 2011, pp. 331–345. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1007/978-3-642-18178-8_29

Abstract. Most of the research work for enforcing security policies on smartphones considered coarse-grained policies, e.g. either to allow an application to run or not. In this paper we present CRePE, the first system that is able to enforce fine-grained policies, e.g. that vary while an application is running, that also depend on the context of the smartphone. A context can be defined by the status of some variables (e.g. location, time, temperature, noise, and light), the presence of other devices, a particular interaction between the user and the smartphone, or a combination of these. CRePE allows context-related policies to be defined either by the user or by trusted third parties. Depending on the authorization, third parties can set a policy on a smartphone at any moment or just when the phone is within a particular context, e.g. within a building, or a plane.

[7] M. Dietz, S. Shekhar, Y. Pisetsky, A. Shu, and D. S. Wallach, "QUIRE: Lightweight provenance for smart phone operating systems," in *20th USENIX Security Symposium*. USENIX Association, 2011. [Online]. Available: http://static.usenix.org/events/sec11/tech/full_papers/Dietz7-26-11.pdf

Abstract. Smartphone apps are often granted to privilege to run with access to the network and sensitive local resources. This makes it difficult for remote endpoints to place any trust in the provenance of network connections originating from a user's device. Even on the phone, different apps with distinct privilege sets can communicate with one another. This can allow one app to trick another into improperly exercising its privileges (resulting in a confused deputy attack). In Quire, we engineered two new security mechanisms into Android to address these issues. First, Quire tracks the call chain of on-device IPCs, allowing an app the choice of operating with the reduced privileges of its callers or exercising its full privilege set by acting explicitly on its own behalf. Second, a lightweight signature scheme allows any app to create a signed statement that can be verified by any app on the same phone. Both of these mechanisms are reflected in network RPCs. This allows remote systems visibility into the state of the phone when the RPC was made. We demonstrate the usefulness of Quire with two example applications: an advertising service that runs advertisements separately from their hosting applications, and a remote payment system. We show that Quire's performance overhead is minimal.

[8] W. Enck, P. Gilbert, B. gon Chun, L. P. Cox, J. Jung, P. McDaniel, and A. Sheth, "TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones," in *9th USENIX Symposium on Operating Systems Design and Implementation.* USENIX Association, 2010, pp. 393–407. [Online]. Available: http://www.usenix.org/events/osdi10/tech/full_papers/Enck.pdf

Abstract. Today's smartphone operating systems frequently fail to provide users with adequate control over and visibility into how third-party applications use their private data. We address these shortcomings with TaintDroid, an efficient, system-wide dynamic taint tracking and analysis system capable of simultaneously tracking multiple sources of sensitive data. TaintDroid provides realtime analysis by leveraging Android's virtualized execution environment. TaintDroid incurs only 14% performance overhead on a CPU-bound microbenchmark and imposes negligible overhead on interactive third-party applications. Using TaintDroid to monitor the behavior of 30 popular third-party Android applications, we found 68 instances of potential misuse of users' private information across 20 applications. Monitoring sensitive data with TaintDroid provides informed use of third-party applications for phone users and valuable input for smartphone security service firms seeking to identify misbehaving applications.

[9] M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, "RiskRanker: Scalable and accurate zero-day Android malware detection," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, ser. MobiSys '12. New York, NY, USA:

ACM, 2012, pp. 281–294. [Online]. Available: http://doi.acm.org.ezproxy.auckland.ac.nz/10.1145/2307636.2307663

> Abstract. Smartphone sales have recently experienced explosive growth. Their popularity also encourages malware authors to penetrate various mobile marketplaces with malicious applications (or apps). These malicious apps hide in the sheer number of other normal apps, which makes their detection challenging. Existing mobile anti-virus software are inadequate in their reactive nature by relying on known malware samples for signature extraction. In this paper, we propose a proactive scheme to spot zero-day Android malware. Without relying on malware samples and their signatures, our scheme is motivated to assess potential security risks posed by these untrusted apps. Specifically, we have developed an automated system called RiskRanker to scalably analyze whether a particular app exhibits dangerous behavior (e.g., launching a root exploit or sending background SMS messages). The output is then used to produce a prioritized list of reduced apps that merit further investigation. When applied to examine 118,318 total apps collected from various Android markets over September and October 2011, our system takes less than four days to process all of them and effectively reports 3281 risky apps. Among these reported apps, we successfully uncovered 718 malware samples (in 29 families) and 322 of them are zero-day (in 11 families). These results demonstrate the efficacy and scalability of RiskRanker to police Android markets of all stripes.

[10] C. Jackson, D. Simon, D. Tan, and A. Barth, "An evaluation of extended validation and picture-in-picture phishing attacks," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, S. Dietrich and R. Dhamija, Eds. Springer Berlin / Heidelberg, 2007, vol. 4886, pp. 281–293. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1007/978-3-540-77366-5_27

> Abstract. In this usability study of phishing attacks and browser anti-phishing defenses, 27 users each classified 12 web sites as fraudulent or legitimate. By dividing these users into three groups, our controlled study measured both the effect of extended validation certificates that appear only at legitimate sites and the effect of reading a help file about security features in Internet Explorer 7. Across all groups, we found that picture-in-picture attacks showing a fake browser window were as effective as the best other phishing technique,

the homograph attack. Extended validation did not help users identify either attack. Additionally, reading the help file made users more likely to classify both real and fake web sites as legitimate when the phishing warning did not appear.

[11] T. Kasper, M. Silbermann, and C. Paar, "All you can eat or Breaking a real-world contactless payment system," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, R. Sion, Ed. Springer Berlin / Heidelberg, 2010, vol. 6052, pp. 343–350. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1007/978-3-642-14577-3_28

> Abstract. We investigated a real-world contactless payment application based on MIFARE Classic cards. In order to analyze the security of the payment system, we combined previous cryptanalytical results and implemented an improved card-only attack with customized low-cost tools, that is to our knowledge the most efficient practical attack to date. We found several flaws implying severe security vulnerabilities on the system level that allow for devastating attacks including identity theft and recharging the amount of money on the cards. We practically verify and demonstrate the attacks on the commercial system.

[12] W. Mazurczyk and J. Lubacz, "LACK – a VoIP steganographic method," *Telecommunication Systems*, vol. 45, pp. 153–163, 2010, 10.1007/s11235-009-9245-y. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1007/s11235-009-9245-y

> Abstract. The paper presents a new steganographic method called LACK (Lost Audio PaCKets Steganography) which is intended mainly for VoIP. The method is presented in a broader context of network steganography and of VoIP steganography in particular. The analytical results presented in the paper concern the influence of LACK's hidden data insertion procedure on the method's impact on quality of voice transmission and its resistance to steganalysis.

[13] A. Munshi, P. Dell, and H. Armstrong, "Insider threat behavior factors: A comparison of theory with reported incidents," in *45th Hawaii International Conference on System Science (HICSS)*, Jan. 2012, pp. 2402–2411. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1109/HICSS.2012.326

> Abstract. Almost all organizations and sectors are currently faced with the problem of insider threats to vital computer

assets. Internal incidents can cause more than just financial losses, the costs can also include loss of clients and damage to an organization's reputation. Substantial academic research investigating internal threats has been conducted. This paper examines a number of theoretical models drawn from academic literature to identify a set of factors that are thought to be behavior factors associated with insider threats. These factors are then critiqued using empirical evidence from reported incidents, resulting in insights into areas where the theoretical perspectives of academic literature are both supported and unsupported by actual case evidence. The paper concludes with recommendations for future research directions for academic researchers.

[14] S. Ransbotham and S. Mitra, "Choice and chance: A conceptual model of paths to information security compromise," *Information Systems Research*, vol. 20, no. 1, pp. 121–139, 2009. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1287/isre.1080.0174

Abstract. No longer the exclusive domain of technology experts, information security is now a management issue. Through a grounded approach using interviews, observations, and secondary data, we advance a model of the information security compromise process from the perspective of the attacked organization. We distinguish between deliberate and opportunistic paths of compromise through the Internet, labeled choice and chance, and include the role of countermeasures, the Internet presence of the firm, and the attractiveness of the firm for information security compromise. Further, using one year of alert data from intrusion detection devices, we find empirical support for the key contributions of the model. We discuss the implications of the model for the emerging research stream on information security in the information systems literature.

[15] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, ser. CCS '09. ACM, 2009, pp. 199–212. [Online]. Available: http://doi.acm.org.ezproxy.auckland.ac.nz/10.1145/1653662.1653687

Abstract. Third-party cloud computing represents the promise of outsourcing as applied to computation. Services, such as Microsoft's Azure and Amazon's EC2, allow users to instantiate virtual machines (VMs) on demand and thus purchase precisely the capacity they require when they require it. In turn, the use of virtualization allows third-party cloud providers to maximize the utilization of their sunk capital costs by multiplexing many customer VMs across a shared physical infrastructure. However, in this paper, we show that this approach can also introduce new vulnerabilities. Using the Amazon EC2 service as a case study, we show that it is possible to map the internal cloud infrastructure, identify where a particular target VM is likely to reside, and then instantiate new VMs until one is placed co-resident with the target. We explore how such placement can then be used to mount cross-VM side-channel attacks to extract information from a target VM on the same machine.

[16] G. Russello, M. Conti, B. Crispo, and E. Fernandes, "MOSES: supporting operation modes on smartphones," in *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*, ser. SACMAT '12. New York, NY, USA: ACM, 2012, pp. 3–12. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1145/2295136.2295140

Abstract: Smartphones are very effective tools for increasing the productivity of business users. With their increasing computational power and storage capacity, smartphones allow end users to perform several tasks and be always updated while on the move. As a consequence, end users require that their personal smartphones are connected to their work IT infrastructure. Companies are willing to support employee-owned smartphones because of the increase in productivity of their employees. However, smartphone security mechanisms have been discovered to offer very limited protection against malicious applications that can leak data stored on them. This poses a serious threat to sensitive corporate data. In this paper we present MOSES, a policy-based framework for enforcing software isolation of applications and data on the Android platform. In MOSES, it is possible to define distinct *security profiles* within a single smartphone. Each security profile is associated with a set of policies that control the access to applications and data. One of the main characteristics of MOSES is the dynamic switching from one security profile to another.

[17] G. Russello, B. Crispo, E. Fernandes, and Y. Zhauniarovich, "YAASE: Yet another Android security extension," in *PASSAT/SocialCom 2011,*

Privacy, Security, Risk and Trust. IEEE Computer Society, 2011, pp. 1033–1040. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1109/PASSAT/SocialCom.2011.151

> Abstract. Three hundred and fifty thousand Android phones are activated each day. The open philosophy adopted by Google makes it easy for third-parties to develop and distribute applications. Unfortunately, the same applies to malicious applications that pose a real threat to users' privacy. The limited security model implemented on the Android Platform has failed in thwarting these attacks. In this paper, we present Yet Another Android Security Extension (YAASE) that provides a fine-grained security mechanism while protecting the user from malicious applications that attempt to leak sensitive information via network access or by privilege spreading through collusion. We have implemented YAASE and evaluated its performance overhead. Preliminary results show the approach is indeed feasible.

[18] M. I. Sharif, A. Lanzi, J. T. Giffin, and W. Lee, "Impeding malware analysis using conditional code obfuscation," in *Proceedings of the Network and Distributed System Security Symposium.* The Internet Society, 2008. [Online]. Available: http://www.isoc.org/isoc/conferences/ndss/08/papers/19_impeding_malware_analysis.pdf

> Abstract. Malware programs that incorporate trigger-based behavior initiate malicious activities based on conditions satisfied only by specific inputs. State-of-the-art malware analyzers discover code guarded by triggers via multiple path exploration, symbolic execution, or forced conditional execution, all without knowing the trigger inputs. We present a malware obfuscation technique that automatically conceals specific trigger-based behavior from these malware analyzers. Our technique automatically transforms a program by encrypting code that is conditionally dependent on an input value with a key derived from the input and then removing the key from the program. We have implemented a compiler-level tool that takes a malware source program and automatically generates an obfuscated binary. Experiments on various existing malware samples show that our tool can hide a significant portion of trigger based code. We provide insight into the strengths, weaknesses, and possible ways to strengthen current analysis approaches in order to defeat this malware obfuscation technique.

[19] S.-T. Sun, E. Pospisil, I. Muslukhov, N. Dindar, K. Hawkey, and K. Beznosov, "What makes users refuse web single sign-on?: An empirical investigation of OpenID," in *Proceedings of the Seventh Symposium on Usable Privacy and Security*, ser. SOUPS '11. ACM, 2011, pp. 4:1–4:20. [Online]. Available: http://doi.acm.org.ezproxy.auckland.ac.nz/10.1145/2078827.2078833

> Abstract. OpenID is an open and promising Web single sign-on (SSO) solution. This work investigates the challenges and concerns web users face when using OpenID for authentication, and identifies what changes in the login flow could improve the users' experience and adoption incentives. We found our participants had several behaviors, concerns, and misconceptions that hinder the OpenID adoption process: (1) their existing password management strategies reduce the perceived usefulness of SSO; (2) many (26%) expressed concerns with single-point-of-failure related issues; (3) most (71%) held the incorrect belief that the OpenID credentials are being given to the content providers; (4) half exhibited an inability to distinguish a fake Google login form, even when prompted; (5) many (40%) were hesitant to consent to the release of their personal profile information; and (6) many (36%) expressed concern with the use of SSO on websites that contain valuable personal information or, conversely, are not trustworthy. We also found that with an improved accordance and privacy control, more than 60% of study participants would use Web SSO solutions on the websites they trust.

[20] M. Theoharidou, S. Kokolakis, M. Karyda, and E. Kiountouzis, "The insider threat to information systems and the effectiveness of ISO17799," *Computers & Security*, vol. 24, no. 6, pp. 472–484, 2005. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1016/j.cose.2005.05.002

> Abstract. Insider threat is widely recognised as an issue of utmost importance for IS security management. In this paper, we investigate the approach followed by ISO17799, the dominant standard in IS security management, in addressing this type of threat. We unfold the criminology theory that has designated the measures against insider misuse suggested by the standard, i.e. the General Deterrence Theory, and explore the possible enhancements to the standard that could result from the study of more recent criminology theories. The paper concludes with supporting the argument for

a multiparadigm and multidisciplinary approach towards IS security management and insider threat mitigation.

[21] A. van Overeem and J. van Oosten, “Towards a pan European e-ID interoperability infrastructure,” in *42nd Hawaii International Conference on System Sciences (HICSS ’09)*. IEEE Computer Society, Jan. 2009, pp. 1–10. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1109/HICSS.2009.466

Abstract. The proliferation of e-Services in most European Countries has been favorable to the emergence of common identity providers and national identity management infrastructures in these countries. The STORK project aims to interconnect all of these identity management infra-structures to form a Pan- European federated e-Identity space. In this paper we show that due to two different identity concepts in use by the European countries, this objective is a far from trivial challenge. Based on our analysis we present two scenarios: homogeneous interoperability for countries with alike identity concepts and heterogeneous interoperability for countries with different identity concepts. For the latter case we present three solution directions to overcome technical limitations and challenges. The STORK project is co-funded by the European Union and will deliver real solutions by implementing five demo projects.

[22] Y. Wang, G. Norcie, S. Komanduri, A. Acquisti, P. G. Leon, and L. F. Cranor, “‘I regretted the minute I pressed share’: A qualitative study of regrets on Facebook,” in *Proceedings of the Seventh Symposium on Usable Privacy and Security*, ser. SOUPS ’11. ACM, 2011, pp. 10:1–10:16. [Online]. Available: http://doi.acm.org.ezproxy.auckland.ac.nz/10.1145/2078827.2078841

Abstract. We investigate regrets associated with users’ posts on a popular social networking site. Our findings are based on a series of interviews, user diaries, and online surveys involving 569 American Facebook users. Their regrets revolved around sensitive topics, content with strong sentiment, lies, and secrets. Our research reveals several possible causes of why users make posts that they later regret: (1) they want to be perceived in favorable ways, (2) they do not think about their reason for posting or the consequences of their posts, (3) they misjudge the culture and norms within their social circles, (4) they are in a “hot” state of high emotion when posting, or under the influence of drugs or alcohol, (5) their postings are seen by an unintended audience, (6) they do not foresee how their posts could be perceived by people within their intended audience, and (7) they misunderstand or misuse the Facebook platform. Some reported incidents had serious repercussions, such as breaking up relationships or job losses. We discuss methodological considerations in studying negative experiences associated with social networking posts, as well as ways of helping users of social networking sites avoid such regrets.

[23] Y. Zhou, X. Zhang, X. Jiang, and V. W. Freeh, "Taming information-stealing smartphone applications (on Android)," in *Trust and Trustworthy Computing - 4th International Conference*, ser. Lecture Notes in Computer Science, vol. 6740. Berlin, Heidelberg: Springer, 2011, pp. 93–107. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1007/978-3-642-21599-5_7

Abstract. Smartphones have been becoming ubiquitous and mobile users are increasingly relying on them to store and handle personal information. However, recent studies also reveal the disturbing fact that users' personal information is put at risk by (rogue) smartphone applications. Existing solutions exhibit limitations in their capabilities in taming these privacy-violating smartphone applications. In this paper, we argue for the need of a new privacy mode in smartphones. The privacy mode can empower users to flexibly control in a fine-grained manner what kinds of personal information will be accessible to an application. Also, the granted access can be dynamically adjusted at runtime in a fine-grained manner to better suit a user's needs in various scenarios (e.g., in a different time or location). We have developed a system called TISSA that implements such a privacy mode on Android. The evaluation with more than a dozen of information-leaking Android applications demonstrates its effectiveness and practicality. Furthermore, our evaluation shows that TISSA introduces negligible performance overhead.