White Paper Addendum

SyScan 2014 talk

"Scientific Best-Practices for Recurring Problems in Computer Security R & D"

Solicitation Ref. SysCan 2014

March 23, 2014

Prepared for
Mr. Thomas Lim
CEO, SyScan Pte. Ltd.
102F Pasir Panjang Road,
#08-02
Singapore 118530

Prepared By:
Dr. Daniel Bilar
Director of Research
Siege Technologies, LLC
540 North Commercial Street
Manchester, NH 03101, USA

Section I. Overview

Section I-A. Introduction

This WP highlight current and upcoming cyber R & D topics as an addendum to the side channel discussions at the SyScan 2014 talk. We mention concurrency concerns, resilience at design time, weird machines and systemic computer security.

Section I-B. Cyber R & D Topics

Section I-B.1 Concurrency Issues

Even though we increasingly rely on concurrent execution, such programs are much more difficult to write, test, and more importantly, debug. This has led to serious *concurrency errors* in many widespread concurrent programs [2], enabling feasible *concurrency attacks*. Many sequential defense techniques, if unaware of concurrent programming, are ineffective against such attacks (see Table 1) and are prone to general TOCTOU attacks if the check and use are not made atomic against concurrently running code.

Findings	Implications
A majority (24 out of 46) of the concurrency attacks corrupt pointer data.	Existing memory safety tools, once made aware of concurrency, may be able to prevent concurrency attacks that corrupt pointer data.
9 concurrency attacks *directly* corrupt scalar data, such as user identifiers, without compromising memory safety.	Few existing defenses handle attacks that directly corrupt scalar data.
Many existing defenses become unsafe in the face of concurrency errors	These defenses must consider concurrent execution.
The exploitability of a concurrency error highly depends on the duration of its *vulnerable window* (*i.e.*, the timing window within which the concurrency error may occur).	New defense techniques may reduce the exploitability of concurrency errors by reducing the duration of the vulnerable window.

Table 1: Investigations into concurrency attacks. Table from [3]

This problem is compounded in our age of data analytics in which massive parallelism (on multi-cores, grids) is tasked to negotiate the data deluge. A promising way forward is the "race-and-repair," or "end-to-end nondeterministic" computing model of the Renaissance kind [7]. This type of non-deterministic computing abandons consistency (the C of Brewer's CAP “Consistency, Availability, and Partition Tolerance, choose any two”) in favor of speed. Concurrency attacks will be much more pronounced in such a regime. Careful study of previous SyScan work work will yield valuable insights and usable techniques in terms of prevention and detection, respectively [14] [15].

Section I-C. Side Channel Leaks Resilience

As noted in the talk, side channel leaks lead to innovative attacks against data structures, protocols, underlying algorithms, infrastructure components and human interactivity [5]. There have been recent advances in quantifying such leaks [2]. Known defenses overwhelmingly rely on multi-level high entropy ‘masking’ implementations; FIPS 140-3 requires consideration of certain side channel attacks (mainly crypto, smart cards, PDA), but was conceived in an era before ubiquitous sensor coverage and

Big Data Analytics and is unlikely to be effective against side channels that occur in distributed infrastructure, multi-core and concurrent programming.

In 2012, Goldwasser and Rothblum [6] articulated theoretical advances in handling side channel leaks at design time: They proved that for any computationally unbounded A observing the results of computationally unbounded leakage functions will learn no more from its observations than it could given blackbox access only to the input-output behavior of P. This result is *unconditional* and does not rely on any secure hardware components. Although side channels cannot be eliminated in general (given irreversible computating architectures), their work gives guidance on how to resisting leakage at *design time* and offers progress towards *formulation of automatic approaches* that generate "leakage-resilience" programs for a wide range of side channel attacks.

Section I-D. Compositional Security / 'Weird Machines'

It is well known that secure composition problem has flummoxed the security community for years. It turns out that a straightforward "halting problem" is at the root of the composition problem: Secure composition requires parser computational equivalence which is undecidable for many language classes.

The Minimal Computational Power Principle is a first-principles language theoretical approach to realizing *provable compositional security of individual modules* by restricting the power of the individual parsers and recognizers. In the opinion of this author, LANGSEC represents the most fundamental intuition in computer security since Thompson (1984) "Reflections on Trusting Trust". Though not a complete compositional security panacea (concurrency attacks are beyond its purview, as well as software-generated hardware attacks) based on LANGSEC's Minimal Power Principle, practical, automatable, compositional security is provably achievable at design time.

Formal input verification (i.e. that an input to a parser/recognizer constitutes a valid expression in a input-handler protocol's grammar), as well as verifying semantics of input transformations is neglected compositional computer security aspect. Exploit writers, who like Molière's Jourdain have been speaking prose all their life without knowing it, take advantage of parser power (e.g. HTML5+CSS shown to be able to implement CA Rule 110, hence botnets are possible within the browser context) and parser discrepancies[9]; in extremis, "weird machines" can be instantiated [8]. These weird machines may exhibit identifying distinct side channel signatures, as noted in the talk.

Section I-E. Systemic Computer Security

Aggregate behavior of simple agents was studied in the past (e.g. Bell Lab's Core War in the 1960s, Conway's Life in the 1970s and Koza's LISP programs in the 1990s). These remained curiosities with no real-world ramifications. This detachment changed in the 21st century with the computerization of vast swaths of life, specifically the advent of automated black-box algorithmic trading. In 2012, Johnson studied phenomenological 'signatures' of interacting autonomous computer agents in real-world dynamic (trading) system; identifying an all-machine time regime characterized by frequent `black swan' events with ultrafast durations (<650ms for crashes, <950ms for spikes; causing 18,000 extreme price changes events [10].

Formal models of *strategic* interactions of self interested agents exist in simplified settings that characterize phenomenological system properties in a Nash equilibrium [11]. However, in real-world interactions, human agents do not (and realistically cannot) compute Nash equilibria. Algorithmic agents could, but it will be of no use for complicated (i.e. real-life) games whose free parameter space induce high-dimensional chaotic attractors; making 'rational learning' effectively random [12].

As a result, event 'signatures' induced by the collective behavior of autonomous programs may herald a necessary evolution of computer security in our AI-headed world: Since aggregate behavior of even simple agents is highly unpredictable (and not consistent across time scales), no useful a priori security guarantees anent the dynamics can be given.

These findings, together with the rapid AI-ization (and UAS-fication of everyday life with the specter of autonomous action chains looming [13]) makes it imperative that dynamic system warning bell signatures be identified and 'circuit breakers' designed. *Systemic computer security* will make its debut: The study of signatures in (side channel) phase space, and the requisite design of circuit breakers and rectifiers when warning bells appear.

Section II. References

[1] S. Hong and M.Kim, "Classification of Race Bug Detection Techniques forMulti-threaded Programs", *Technical Report*, Computer Science Department, KAIST (South Korea), 2012 http://swtv.kaist.ac.kr/publications/race-classification.pdf

[2] S. Lu et al, "Learning from mistakes: a comprehensive study on real world concurrency bug characteristics", *ASPLOS 13,* March 2008 http://web1.cs.columbia.edu/~junfeng/10fa-e6998/papers/concurrency-bugs.pdf

[3] J. Yang et al, "Concurrency attacks", *USENIX HotPar*, 2012 https://www.usenix.org/system/files/conference/hotpar12/hotpar12-final44.pdf

[4] J. Demme et al, "Side-channel Vulnerability Factor: A Metric for Measuring Information Leakage", *ICSA*, 2012 http://www.cs.columbia.edu/~simha/preprint_isca12_svf.pdf

[5] R. Ensafi et al, "Students Who Don't Understand Information Flow Should be Eaten: An Experience Paper", *NSPW*, September 2012 https://www.usenix.org/system/files/conference/cset12/cset12-final23.pdf

[6] S. Goldwasser and G. Rothblum, "How to Compute in the Presence of Leakage," *FOCS*, October 2012, pp.31-40 http://eccc.hpi-web.de/report/2012/010/download/

[7] D. Ungar, "Everything You Know (about Parallel Programming) Is Wrong!: A Wild Screed about the Future ", *SplashCon* , October 2011

http://splashcon.org/2011/program/dls/245-invited-talk-2

[8] L. Sassaman et al, "The Halting Problems of Network Stack Insecurity", *;login* 36:6, December 2011, https://c59951.ssl.cf2.rackcdn.com/3094-105516-Sassaman.pdf

[9] S. Jana and V. Shmatikov, "Abusing File Processing in Malware Detectors for Fun and Profit, *IEEE S & P*, Oakland (CA), 2012, http://www.cs.utexas.edu/~shmat/shmat_oak12av.pdf

[10] N. Johnson et al, "Abrupt rise of new machine ecology beyond human response time." *Nature Scientific reports* 3, 2013 http://dx.doi.org/10.1038/srep02627

[11] Y. Vorobeychik et al, "Noncooperatively Optimized Tolerance: Decentralized Strategic Optimization in Complex Systems", *Phys. Review Letters* 107 (10), 2011, http://link.aps.org/doi/10.1103/PhysRevLett.107.108702

[12] T. Galla and J. Farmer, "Complex dynamics in learning complicated games", *PNAS* 110 (4), 2013 http://www.pnas.org/content/110/4/1232.full.pdf+html

[13] R. Arkin et al, "Moral decision making in autonomous systems: Enforcement, moral emotions, dignity, trust, and deception", *Proceedings of the IEEE* 100.3, 2012, pp. 571-589 https://smartech.gatech.edu/bitstream/handle/1853/40769/IEEE-ethicsv17.pdf?sequence=1

[14] M. Jurczyk and G. Coldwind, "Identifying and Exploiting Windows Kernel Race Conditions via Memory Access Patterns", *SyScan (Singapore)*, April 2013 http://j00ru.vexillium.org/?p=1695

[15] G. Wicherksy "Taming the ROPe on Sandy Bridge", *SyScan (Singapore)*, April 2013 http://www.syscan.org/index.php/download/get/3c6891f2e90e661ea23224cd8f419262/SyScan2013_DAY1_SPEAKER05_Georg_WIcherski_Taming_ROP_ON_SANDY_BRIDGE_syscan.zip

Section III. Appendix

About the author: Daniel is Director of Research and Senior Principal Scientist for Siege Technologies, a 2009 boutique cyber R & D company specializing in offensive cyber-security supporting the US DoD and IC. Daniel has researched, published and lectured world-wide on highly evolved malware detection and classification, cyber warfare and quantitative compositional risk analysis/ management of computer systems. He holds a Ph.D. (2003) in Engineering Sciences from Dartmouth College (NH), a Master's of Engineering (1997) in Operations Research and Industrial Engineering from Cornell University (NY) and a Bachelor's of Arts (1995) in Computer Science from Brown University (RI). Contact him at dbilar@acm.org

The opinions in this WP are his own and not necessarily endorsed by Siege Technologies.