James Newsome

http://www.jimnewsome.net
jim@jimnewsome.net

Education

Carnegie Mellon University, Pittsburgh, PA
M.S. in Electrical and Computer Engineering, December 2003
Ph.D. in Electrical and Computer Engineering, December 2008
GPA: 3.94

University of Michigan, Ann Arbor, MI
BSE in Computer Engineering, May 2002
GPA: 3.88

Employment

jimnewsome.net *Sole Proprietor*
Pittsburgh, PA August 2012 –
I am now an independent developer. I am available for freelance and consulting in cyber-security and software development.

Carnegie Mellon University *Systems Scientist*
Pittsburgh, PA July 2010 – August 2012
Performed research and development in virtualization and trusted computing. I was especially involved in the development of the TrustVisor hypervisor, and the TEE-SDK. These tools allow a piece of code to run in isolation from the rest of the computer system, allowing security-critical Pieces of Application Code to run securely even if the operating system is infested with malware. We released these as open source as part of the xmhf project (http://xmhf.org).

Bosch Research and Technology Center *Research Engineer*
Pittsburgh, PA November 2008 – July 2010
Performed internal consulting on cryptography and design of secure networked embedded systems.

Intel Research Pittsburgh *Intern Researcher*
Pittsburgh, PA May 2004 – January 2005
May 2005 – August 2005
Researched and developed Polygraph, a system to automatically generate signatures for polymorphic worms. Work resulted in an open source release, and two publications (IEEE S&P 2005 and RAID 2006)

Peer-reviewed Publications

Building Verifiable Trusted Path on Commodity x86 Computers. Zongwei Zhou, Virgil Gligor, James Newsome, and Jon McCune. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2012.

CARMA: A Hardware Tamper-Resistant Isolated Execution Environment on Commodity x86 Platforms. Amit Vasudevan, Jonathan M. McCune, James Newsome, Adrian Perrig, and Leendert van Doorn. In *Proceedings of the ACM Symposium on Information, Computer and Communications Security (ASIACCS)* 2012.

Trustworthy Execution on Mobile Devices: What security properties can my mobile platform give me? Amit Vasudevan, Emmanuel Owusu, Zongwei Zhou, James Newsome, and Jonathan M. McCune. In *Proceedings of Trust and Trustworthy Computing* 2012.

Access Right Assignment Mechanisms for Secure Home Networks. Tiffany Hyun-Jin Kim, Lujo Bauer, James Newsome, Adrian Perrig, and Jesse Walker. In *Journal of Communications and Networks*, 13(2):175–186, 2011.

Challenges in Access Right Assignment for Secure Home Networks. Tiffany Hyun-Jin Kim, Lujo Bauer, James Newsome, Adrian Perrig, and Jesse Walker. In *Proceedings of the 5th USENIX Workshop on Hot Topics in Security (HotSec '10)*, 2010.

Measuring Channel Capacity to Distinguish Undue Influence. James Newsome, Stephen McCamant, and Dawn Song. In *Proceedings of the Fourth ACM SIGPLAN Workshop on Programming Languages and Analysis for Security (PLAS)*, June 2009.

Towards automatic discovery of deviations in binary implementations with applications to error de-

tection and fingerprint generation. David Brumley, Juan Caballero, Zhenkai Liang, James Newsome, Dawn Song. In *Proceedings of the 2007 USENIX Security Conference*, 2007.

Sweeper: a lightweight end-to-end system for defending against fast worms. Joseph Tucek, James Newsome, Shan lu, Chengdu Huang, Spiros Xanthos, David Brumley, Yuanuyan Zhou, and Dawn Song. In *Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems*, 2007.

Towards automatic generation of vulnerability-based signatures. David Brumley, James Newsome, Dawn Song, Hao Wang, and Somesh Jha. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, 2006.

Replayer: Automatic protocol replay by binary analysis. James Newsome, David Brumley, Jason Franklin, and Dawn Song. In Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS), October 2006.

Paragraph: Thwarting signature learning by training maliciously. James Newsome, Brad Karp, and Dawn Song. In *Proceedings of the 9th International Symposium On Recent Advances In Intrusion Detection* (RAID 2006), September 2006.

Vulnerability-specific execution filtering for exploit prevention on commodity software. James Newsome, David Brumley, Dawn Song. In *Proceedings of the 13th Annual Network and Distributed System Security Symposium (NDSS '06)*, February 2006.

Towards automatic generation of vulnerability-based signatures. David Brumley, James Newsome, Dawn Song, Hao Wang, and Somesh Jha. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 2006.

Polygraph: Automatically generating signatures for polymorphic worms. James Newsome, Brad Karp, and Dawn Song. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 2005.

Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software. James Newsome and Dawn Song. In *Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS '05)*, February 2005.

The Sybil attack in sensor networks: analysis & defenses. James Newsome, Runting Shi, Dawn Song, and Adrian Perrig. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, April 2004.

GEM: Graph EMbedding for routing and data-centric storage in wireless sensor networks. James Newsome and Dawn Song. In *Proceedings of ACM SenSys*, November 2003.

Technical Reports

Influence: A quantitative approach for data integrity. James Newsome and Dawn Song. Carnegie Mellon Cylab. Technical Report CMU-CyLab-08-005, February 2008.

Towards practical automatic generation of multipath vulnerabity signatures. David Brumley, Zhenkai Liang, James Newsome, and Dawn Song. Carnegie Mellon University School of Computer Science. Technical Report CMU-CS-07-150, 2007.

BitScope: Automatically Dissecting Malicious Binaries. David Brumley, Cody Hartwig, Min Gyung Kang, Zhenkai Liang, James Newsome, Pongsin Poosankam, Dawn Song, and Heng Yin. Carnegie Mellon School of Computer Science. Technical Report CMU-CS-07-133, March 2007.

Alias analysis for assembly. David Brumley and James Newsome. Carnegie Mellon University School of Computer Science. Technical Report CMU-CS-06-180, December 2006.

Sting: an End-to-End Self-healing system for Defending against Zero-day Worm Attacks on Commodity Software. James Newsome, David Brumley, and Dawn Song. Technical Report CMU-CS-05-191, 2005.

Book Chapters

Automatically Identifying Trigger-based Behavior in Malware. David Brumley, Cody Hartwig, Zhenkai Liang, James Newsome, Pongsin Poosankam, Dawn Song, and Heng Yin. In Botnet Anal-

ysis and Defense, vol. 36 of Advances in Information Security Series, Wenke Lee, Cliff Wang, and David Dagon (editors), pp. 65-88, Springer, 2008.

Sting: An End-to-End Self-Healing System for Defending against Internet Worms. David Brumley, James Newsome, and Dawn Song. In Malware Detection, Springer Publications, 2007.