Sokratis K. Katsikas
Javier Lopez
Michael Backes
Stefanos Gritzalis
Bart Preneel (Eds.)

LNCS 4176

Information Security

9th International Conference, ISC 2006
Samos Island, Greece, August/September 2006
Proceedings

Springer

Sokratis K. Katsikas   Javier Lopez
Michael Backes   Stefanos Gritzalis
Bart Preneel (Eds.)

Information Security

9th International Conference, ISC 2006
Samos Island, Greece, August 30 – September 2, 2006
Proceedings

Springer

Volume Editors

Sokratis K. Katsikas
University of the Aegean, Mytilene, Greece
E-mail: ska@aegean.gr

Javier Lopez
University of Malaga, Spain
E-mail: jlm@lcc.uma.es

Michael Backes
Saarland University, Saarbrücken, Germany
E-mail: backes@cs.uni-sb.de

Stefanos Gritzalis
University of the Aegean, Samos, Greece
E-mail: sgritz@aegean.gr

Bart Preneel
Katholieke Universiteit Leuven, Belgium
E-mail: bart.preneel@esat.kuleuven.be

Library of Congress Control Number: 2006931359

CR Subject Classification (1998): E.3, E.4, D.4.6, F.2.1, C.2, J.1, C.3, K.4.4, K.6.5

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN 0302-9743
ISBN-10 3-540-38341-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-38341-3 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11836810 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 4176

Preface

This volume contains the papers presented at the 9th Information Security Conference (ISC 2006) held on Samos Island, Greece, during August 30 – September 2, 2006. The Conference was organized by the University of the Aegean, Greece.

ISC was first initiated as a workshop, ISW in Japan in 1997, ISW 1999 in Malaysia, ISW 2000 in Australia and then changed to the current name ISC when it was held in Spain in 2001 (ISC 2001). The latest conferences were held in Brazil (ISC 2002), UK (ISC 2003), USA (ISC 2004), and Singapore (ISC 2005).

ISC 2006 provided an international forum for sharing original research results and application experiences among specialists in fundamental and applied problems of information security.

In response to the Call for Papers, 188 papers were submitted. Each paper was reviewed by three members of the PC, on the basis of their significance, novelty, and technical quality. Of the papers submitted, 38 were selected for presentation, with an acceptance rate of 20%.

We would like to express our gratitude to the members of the Program Committee, as well as the external reviewers, for their constructive and insightful comments during the review process and discussion that followed. Moreover, we would like to thank all the members of the Organizing Committee for their continuous and valuable support. We also wish to express our thanks to Alfred Hofmann and his colleagues from Springer, for their co-operation and their excellent work during the publication process. Finally, we would like to thank all the people who submitted their papers to ISC 2006, including those whose submissions were not selected for publication, and all the delegates from around the world, who attended the ISC 2006 9th Information Security Conference. Without their support the conference would not have been possible.

August 2006

Sokratis K. Katsikas
Javier Lopez
Michael Backes
Stefanos Gritzalis
Bart Preneel

ISC 2006 9th Information Security Conference

General Co-chairs

Sokratis K. Katsikas	University of the Aegean, Greece
Javier Lopez	University of Malaga, Spain

Program Committee Co-chairs

Michael Backes	Saarland University, Germany
Stefanos Gritzalis	University of the Aegean, Greece
Bart Preneel	Katholieke Universiteit Leuven, Belgium

Program Committee

A. Acquisti	Carnegie Mellon University, USA
N. Asokan	Nokia Research Center, Finland
V. Atluri	Rutgers University, USA
T. Aura	Microsoft Research, UK
F. Bao	Institute for Infocomm Research, Singapore
J. Baras	University of Maryland, USA
D. Basin	ETH Zurich, Switzerland
G. Bella	University of Catania, Italy
J. Benaloh	Microsoft Research, USA
E. Bertino	CERIAS, Purdue University, USA
A. Biryukov	University of Luxembourg, Luxembourg
M. Burmester	Florida State University, USA
S. De Capitani di Vimercati	University of Milan, Italy
D. Catalano	ENS, France
D. Chadwick	University of Kent, UK
R. Cramer	CWI and Leiden University, The Netherlands
B. Crispo	Vrije Universiteit Amsterdam, The Netherlands
G. Danezis	Katholieke Universiteit Leuven, Belgium
A. Datta	Stanford University, USA
E. Dawson	Queensland University of Technology, Australia
S. Furnell	University of Plymouth, UK
V. Gligor	University of Maryland, USA
D. Gollmann	Hamburg University of Technology, Germany
H. Handschuh	Spansion, France
D. Hofheinz	CWI, The Netherlands
D. Hutter	DFKI, Germany

J. Ioannidis	Columbia University, USA
A. Juels	RSA Laboratories, USA
T. Karygiannis	NIST, USA
S. Kokolakis	University of the Aegean, Greece
C. Lambrinoudakis	University of the Aegean, Greece
H. Lipmaa	Cybernetica AS & University of Tartu, Estonia
M. Mambo	University of Tsukuba, Japan
H. Mantel	RWTH Aachen, Germany
W. Mao	HP Labs, China
F. Massacci	University of Trento, Italy
M. Merabti	Liverpool John Moores University, UK
C. Mitchell	Royal Holloway, University of London, UK
A. Odlyzko	University of Minnesota, USA
E. Okamoto	University of Tsukuba, Japan
J. A. Onieva	University of Malaga, Spain
R. Oppliger	eSECURITY Technologies, Switzerland
G. Pernul	University of Regensburg, Germany
A. Pfitzmann	Dresden University of Technology, Germany
V. Rijmen	Graz University of Technology, Austria
P.Y.A. Ryan	University of Newcastle upon Tyne, UK
K. Sakurai	Kyushu University, Japan
P. Samarati	University of Milan, Italy
D. Serpanos	University of Patras, Greece
P. Tuyls	Philips Research and K.U. Leuven, The Netherlands and Belgium
J. Villar	Universitat Politecnica Catalunya, Spain
M. Yung	Columbia University and RSA Laboratories, USA
Y. Zheng	University of North Carolina at Charlotte, USA
J. Zhou	Institute for Infocomm Research, Singapore

External Reviewers

Asnar, W.
Balopoulos, Theodoros
Batina, Lejla
Bergmann, Mike
Bin Abd Razak, Shukor
Cardenas, Alvaro A.
Carvounas, Christophe
Cascella, Roberto
Chen, Haibo
Clauí, Sebastian
Cremonini, Marco
Das, Tanmoy Kanti
de Medeiros, Breno
De Win, Bart
Dobmeier, Wolfgang
Doser, Juergen
Fehr, Serge
Fergus, Paul
Fernandez, Gerardo
Fouque, Pierre-Alain
Fuchs, L.
Fukushima, Kazuhide
Geneiatakis, Dimitris
Gonzalez, Juanma
Goubin, Louis
Gouget, Aline
Guajardo, Jorge

Gymnopoulos, Lazaros
Hankes Drielsma, Paul
Henricksen, Matt
Her, Yong-Sork
Herranz, Javier
Hilty, Manuel
Holmstroem, Ursula
Hori, Yoshiaki
Kambourakis, George
Karyda, Maria
Katzenbeisser, Stefan
Kiltz, Eike
Koepf, Boris
Kolter, Jan
Koshutanski, Hristo
Krausser, Tina
Kunihiro, Noboru
Kopsell, Stefan
Lano, Joseph
Lee, Soo Bum
Lindqvist, Janne
Llewellyn-Jones, David
Meckl, Norbert S.
Muschall, B.
Naccache, David
Naliuka, Katerina
Neven, Gregory
Padr, Carles
Papadaki, Maria
Peng, Kun
Qiang, Weizhong
Schillinger, Rolf
Schlaeger, Christian
Schrijen, Geert-Jan
Seys, Stefaan
Skoric, Boris
Stefanidis, Kyriakos
Steinbrecher, Sandra
Sudbrock, Henning
Taban, Gelareh
Tu, Feng
Ueshige, Yoshifumi
van Le, Tri
Vercauteren, Frederik
Vigano, Luca
Volkamer, Melanie
Westfeld, Andreas
Wolf, Christopher
Woo, Chaw-Seng
Wu, Yongdong
Yatshukin, Artsiom
Yudistira, D.
Zannone, Nicola
Zhao, Yunlei
Zhong, Xiang
Zhou, Bo
Zhou, Juxiang
Zhu, Xusong

Lecture Notes in Computer Science

For information about Vols. 1–4051

please contact your bookseller or Springer

Vol. 4180: M. Kohlhase, OMDoc – An Open Markup Format for Mathematical Documents [version 1.2]. XIX, 428 pages. 2006. (Sublibrary LNAI).

Vol. 4176: S.K. Katsikas, J. Lopez, M. Backes, S. Gritzalis, B. Preneel (Eds.), Information Security. XIV, 548 pages. 2006.

Vol. 4163: H. Bersini, J. Carneiro (Eds.), Artificial Immune Systems. XII, 460 pages. 2006.

Vol. 4162: R. Královič, P. Urzyczyn (Eds.), Mathematical Foundations of Computer Science 2006. XV, 814 pages. 2006.

Vol. 4159: J. Ma, H. Jin, L.T. Yang, J.J.-P. Tsai (Eds.), Ubiquitous Intelligence and Computing. XXII, 1190 pages. 2006.

Vol. 4155: O. Stock, M. Schaerf (Eds.), Reasoning, Action and Interaction in AI Theories and Systems. XVIII, 343 pages. 2006. (Sublibrary LNAI).

Vol. 4153: N. Zheng, X. Jiang, X. Lan (Eds.), Advances in Machine Vision, Image Processing, and Pattern Analysis. XIII, 506 pages. 2006.

Vol. 4152: Y. Manolopoulos, J. Pokorný, T. Sellis (Eds.), Advances in Databases and Information Systems. XV, 448 pages. 2006.

Vol. 4151: A. Iglesias, N. Takayama (Eds.), Mathematical Software - ICMS 2006. XVII, 452 pages. 2006.

Vol. 4146: J.C. Rajapakse, L. Wong, R. Acharya (Eds.), Pattern Recognition in Bioinformatics. XIV, 186 pages. 2006. (Sublibrary LNBI).

Vol. 4144: T. Ball, R.B. Jones (Eds.), Computer Aided Verification. XV, 564 pages. 2006.

Vol. 4139: T. Salakoski, F. Ginter, S. Pyysalo, T. Pahikkala, Advances in Natural Language Processing. XVI, 771 pages. 2006. (Sublibrary LNAI).

Vol. 4138: X. Cheng, W. Li, T. Znati (Eds.), Wireless Algorithms, Systems, and Applications. XVI, 709 pages. 2006.

Vol. 4137: C. Baier, H. Hermanns (Eds.), CONCUR 2006 – Concurrency Theory. XIII, 525 pages. 2006.

Vol. 4134: K. Yi (Ed.), Static Analysis. XIII, 443 pages. 2006.

Vol. 4133: J. Gratch, M. Young, R. Aylett, D. Ballin, P. Olivier (Eds.), Intelligent Virtual Agents. XIV, 472 pages. 2006. (Sublibrary LNAI).

Vol. 4130: U. Furbach, N. Shankar (Eds.), Automated Reasoning. XV, 680 pages. 2006. (Sublibrary LNAI).

Vol. 4129: D. McGookin, S. Brewster (Eds.), Haptic and Audio Interaction Design. XII, 167 pages. 2006.

Vol. 4128: W.E. Nagel, W.V. Walter, W. Lehner (Eds.), Euro-Par 2006 Parallel Processing. XXXIII, 1221 pages. 2006.

Vol. 4127: E. Damiani, P. Liu (Eds.), Data and Applications Security XX. X, 319 pages. 2006.

Vol. 4124: H. de Meer, J.P. G. Sterbenz (Eds.), Self-Organising Systems. XIV, 261 pages. 2006.

Vol. 4121: A. Biere, C.P. Gomes (Eds.), Theory and Applications of Satisfiability Testing - SAT 2006. XII, 438 pages. 2006.

Vol. 4119: C. Dony, J.L. Knudsen, A. Romanovsky, A. Tripathi (Eds.), Advanced Topics in Exception Handling Components. X, 302 pages. 2006.

Vol. 4117: C. Dwork (Ed.), Advances in Cryptology - CRYPTO 2006. XIII, 621 pages. 2006.

Vol. 4116: R. De Prisco, M. Yung (Eds.), Security and Cryptography for Networks. XI, 366 pages. 2006.

Vol. 4115: D.-S. Huang, K. Li, G.W. Irwin (Eds.), Computational Intelligence and Bioinformatics, Part III. XXI, 803 pages. 2006. (Sublibrary LNBI).

Vol. 4114: D.-S. Huang, K. Li, G.W. Irwin (Eds.), Computational Intelligence, Part II. XXVII, 1337 pages. 2006. (Sublibrary LNAI).

Vol. 4113: D.-S. Huang, K. Li, G.W. Irwin (Eds.), Intelligent Computing, Part I. XXVII, 1331 pages. 2006.

Vol. 4112: D.Z. Chen, D. T. Lee (Eds.), Computing and Combinatorics. XIV, 528 pages. 2006.

Vol. 4111: F.S. de Boer, M.M. Bonsangue, S. Graf, W.-P. de Roever (Eds.), Formal Methods for Components and Objects. VIII, 447 pages. 2006.

Vol. 4110: J. Díaz, K. Jansen, J.D.P. Rolim, U. Zwick (Eds.), Approximation, Randomization, and Combinatorial Optimization. XII, 522 pages. 2006.

Vol. 4109: D.-Y. Yeung, J.T. Kwok, A. Fred, F. Roli, D. de Ridder (Eds.), Structural, Syntactic, and Statistical Pattern Recognition. XXI, 939 pages. 2006.

Vol. 4108: J.M. Borwein, W.M. Farmer (Eds.), Mathematical Knowledge Management. VIII, 295 pages. 2006. (Sublibrary LNAI).

Vol. 4106: T.R. Roth-Berghofer, M.H. Göker, H. A. Güvenir (Eds.), Advances in Case-Based Reasoning. XIV, 566 pages. 2006. (Sublibrary LNAI).

Vol. 4104: T. Kunz, S.S. Ravi (Eds.), Ad-Hoc, Mobile, and Wireless Networks. XII, 474 pages. 2006.

Vol. 4099: Q. Yang, G. Webb (Eds.), PRICAI 2006: Trends in Artificial Intelligence. XXVIII, 1263 pages. 2006. (Sublibrary LNAI).

Vol. 4098: F. Pfenning (Ed.), Term Rewriting and Applications. XIII, 415 pages. 2006.

Vol. 4097: X. Zhou, O. Sokolsky, L. Yan, E.-S. Jung, Z. Shao, Y. Mu, D.C. Lee, D. Kim, Y.-S. Jeong, C.-Z. Xu (Eds.), Emerging Directions in Embedded and Ubiquitous Computing. XXVII, 1034 pages. 2006.

Vol. 4096: E. Sha, S.-K. Han, C.-Z. Xu, M.H. Kim, L.T. Yang, B. Xiao (Eds.), Embedded and Ubiquitous Computing. XXIV, 1170 pages. 2006.

Vol. 4094: O. H. Ibarra, H.-C. Yen (Eds.), Implementation and Application of Automata. XIII, 291 pages. 2006.

Vol. 4093: X. Li, O.R. Zaïane, Z. Li (Eds.), Advanced Data Mining and Applications. XXI, 1110 pages. 2006. (Sublibrary LNAI).

Vol. 4092: J. Lang, F. Lin, J. Wang (Eds.), Knowledge Science, Engineering and Management. XV, 664 pages. 2006. (Sublibrary LNAI).

Vol. 4091: G.-Z. Yang, T. Jiang, D. Shen, L. Gu, J. Yang (Eds.), Medical Imaging and Augmented Reality. XIII, 399 pages. 2006.

Vol. 4090: S. Spaccapietra, K. Aberer, P. Cudré-Mauroux (Eds.), Journal on Data Semantics VI. XI, 211 pages. 2006.

Vol. 4089: W. Löwe, M. Südholt (Eds.), Software Composition. X, 339 pages. 2006.

Vol. 4088: Z.-Z. Shi, R. Sadananda (Eds.), Agent Computing and Multi-Agent Systems. XVII, 827 pages. 2006. (Sublibrary LNAI).

Vol. 4087: F. Schwenker, S. Marinai (Eds.), Artificial Neural Networks in Pattern Recognition. IX, 299 pages. 2006. (Sublibrary LNAI).

Vol. 4085: J. Misra, T. Nipkow, E. Sekerinski (Eds.), FM 2006: Formal Methods. XV, 620 pages. 2006.

Vol. 4084: M.A. Wimmer, H.J. Scholl, Å. Grönlund, K.V. Andersen (Eds.), Electronic Government. XV, 353 pages. 2006.

Vol. 4083: S. Fischer-Hübner, S. Furnell, C. Lambrinoudakis (Eds.), Trust and Privacy in Digital Business. XIII, 243 pages. 2006.

Vol. 4082: K. Bauknecht, B. Pröll, H. Werthner (Eds.), E-Commerce and Web Technologies. XIII, 243 pages. 2006.

Vol. 4081: A. M. Tjoa, J. Trujillo (Eds.), Data Warehousing and Knowledge Discovery. XVII, 578 pages. 2006.

Vol. 4080: S. Bressan, J. Küng, R. Wagner (Eds.), Database and Expert Systems Applications. XXI, 959 pages. 2006.

Vol. 4079: S. Etalle, M. Truszczyński (Eds.), Logic Programming. XIV, 474 pages. 2006.

Vol. 4077: M.-S. Kim, K. Shimada (Eds.), Geometric Modeling and Processing - GMP 2006. XVI, 696 pages. 2006.

Vol. 4076: F. Hess, S. Pauli, M. Pohst (Eds.), Algorithmic Number Theory. X, 599 pages. 2006.

Vol. 4075: U. Leser, F. Naumann, B. Eckman (Eds.), Data Integration in the Life Sciences. XI, 298 pages. 2006. (Sublibrary LNBI).

Vol. 4074: M. Burmester, A. Yasinsac (Eds.), Secure Mobile Ad-hoc Networks and Sensors. X, 193 pages. 2006.

Vol. 4073: A. Butz, B. Fisher, A. Krüger, P. Olivier (Eds.), Smart Graphics. XI, 263 pages. 2006.

Vol. 4072: M. Harders, G. Székely (Eds.), Biomedical Simulation. XI, 216 pages. 2006.

Vol. 4071: H. Sundaram, M. Naphade, J.R. Smith, Y. Rui (Eds.), Image and Video Retrieval. XII, 547 pages. 2006.

Vol. 4070: C. Priami, X. Hu, Y. Pan, T.Y. Lin (Eds.), Transactions on Computational Systems Biology V. IX, 129 pages. 2006. (Sublibrary LNBI).

Vol. 4069: F.J. Perales, R.B. Fisher (Eds.), Articulated Motion and Deformable Objects. XV, 526 pages. 2006.

Vol. 4068: H. Schärfe, P. Hitzler, P. Øhrstrøm (Eds.), Conceptual Structures: Inspiration and Application. XI, 455 pages. 2006. (Sublibrary LNAI).

Vol. 4067: D. Thomas (Ed.), ECOOP 2006 – Object-Oriented Programming. XIV, 527 pages. 2006.

Vol. 4066: A. Rensink, J. Warmer (Eds.), Model Driven Architecture – Foundations and Applications. XII, 392 pages. 2006.

Vol. 4065: P. Perner (Ed.), Advances in Data Mining. XI, 592 pages. 2006. (Sublibrary LNAI).

Vol. 4064: R. Büschkes, P. Laskov (Eds.), Detection of Intrusions and Malware & Vulnerability Assessment. X, 195 pages. 2006.

Vol. 4063: I. Gorton, G.T. Heineman, I. Crnkovic, H.W. Schmidt, J.A. Stafford, C.A. Szyperski, K. Wallnau (Eds.), Component-Based Software Engineering. XI, 394 pages. 2006.

Vol. 4062: G. Wang, J.F. Peters, A. Skowron, Y. Yao (Eds.), Rough Sets and Knowledge Technology. XX, 810 pages. 2006. (Sublibrary LNAI).

Vol. 4061: K. Miesenberger, J. Klaus, W. Zagler, A.I. Karshmer (Eds.), Computers Helping People with Special Needs. XXIX, 1356 pages. 2006.

Vol. 4060: K. Futatsugi, J.-P. Jouannaud, J. Meseguer (Eds.), Algebra, Meaning, and Computation. XXXVIII, 643 pages. 2006.

Vol. 4059: L. Arge, R. Freivalds (Eds.), Algorithm Theory – SWAT 2006. XII, 436 pages. 2006.

Vol. 4058: L.M. Batten, R. Safavi-Naini (Eds.), Information Security and Privacy. XII, 446 pages. 2006.

Vol. 4057: J.P.W. Pluim, B. Likar, F.A. Gerritsen (Eds.), Biomedical Image Registration. XII, 324 pages. 2006.

Vol. 4056: P. Flocchini, L. Gąsieniec (Eds.), Structural Information and Communication Complexity. X, 357 pages. 2006.

Vol. 4055: J. Lee, J. Shim, S.-g. Lee, C. Bussler, S. Shim (Eds.), Data Engineering Issues in E-Commerce and Services. IX, 290 pages. 2006.

Vol. 4054: A. Horváth, M. Telek (Eds.), Formal Methods and Stochastic Models for Performance Evaluation. VIII, 239 pages. 2006.

Vol. 4053: M. Ikeda, K.D. Ashley, T.-W. Chan (Eds.), Intelligent Tutoring Systems. XXVI, 821 pages. 2006.

Vol. 4052: M. Bugliesi, B. Preneel, V. Sassone, I. Wegener (Eds.), Automata, Languages and Programming, Part II. XXIV, 603 pages. 2006.

¥58L四元

Table of Contents

Software Security

Extending .NET Security to Unmanaged Code 1
Patrick Klinkoff, Christopher Kruegel, Engin Kirda, Giovanni Vigna

Transparent Run-Time Prevention of Format-String Attacks Via Dynamic Taint and Flexible Validation 17
Zhiqiang Lin, Nai Xia, Guole Li, Bing Mao, Li Xie

Privacy and Anonymity

Low Latency Anonymity with Mix Rings 32
Matthew Burnside, Angelos D. Keromytis

Breaking Four Mix-Related Schemes Based on Universal Re-encryption 46
George Danezis

Weak k-Anonymity: A Low-Distortion Model for Protecting Privacy 60
Maurizio Atzori

Protecting Data Privacy Through Hard-to-Reverse Negative Databases 72
Fernando Esponda, Elena S. Ackley, Paul Helman, Haixia Jia, Stephanie Forrest

Block Ciphers and Hash Functions

Related-Key Rectangle Attack on 42-Round SHACAL-2 85
Jiqiang Lu, Jongsung Kim, Nathan Keller, Orr Dunkelman

On the Collision Resistance of RIPEMD-160 101
Florian Mendel, Norbert Pramstaller, Christian Rechberger, Vincent Rijmen

Digital Signatures

Blind Ring Signatures Secure Under the Chosen-Target-CDH Assumption 117
Javier Herranz, Fabien Laguillaumie

Multi-party Concurrent Signatures 131
Dongvu Tonien, Willy Susilo, Reihaneh Safavi-Naini

Formal Security Model of Multisignatures 146
Yuichi Komano, Kazuo Ohta, Atsushi Shimbo, Shinichi Kawamura

Cryptanalysis of Variants of UOV 161
Yuh-Hua Hu, Chun-Yen Chou, Lih-Chung Wang, Feipei Lai

Stream Ciphers

TRIVIUM: A Stream Cipher Construction Inspired by Block Cipher Design Principles 171
Christophe De Cannière

Cryptanalysis of the Bluetooth E_0 Cipher Using OBDD's 187
Yaniv Shaked, Avishai Wool

Encryption I

A Partial Key Exposure Attack on RSA Using a 2-Dimensional Lattice 203
Ellen Jochemsz, Benne de Weger

On the Integration of Public Key Data Encryption and Public Key Encryption with Keyword Search 217
Joonsang Baek, Reihaneh Safavi-Naini, Willy Susilo

Collusion-Free Policy-Based Encryption 233
Walid Bagga, Refik Molva

Pervasive Computing

Using Multiple Smart Cards for Signing Messages at Malicious Terminals 246
István Zsolt Berta

Diverging Keys in Wireless Sensor Networks 257
Michał Ren, Tanmoy Kanti Das, Jianying Zhou

Encryption II

A Generic Transformation from Symmetric to Asymmetric Broadcast Encryption 270
Ulrich Huber, Ahmad-Reza Sadeghi

Transparent Image Encryption Using Progressive JPEG 286
Thomas Stütz, Andreas Uhl

Network Security

Preserving TCP Connections Across Host Address Changes 299
Vassilis Prevelakis, Sotiris Ioannidis

A Security Architecture for Protecting LAN Interactions 311
André Zúquete, Hugo Marques

Simulation of Internet DDoS Attacks and Defense 327
Igor Kotenko, Alexander Ulanov

SNOOZE: Toward a Stateful NetwOrk prOtocol fuzZEr 343
Greg Banks, Marco Cova, Viktoria Felmetsger, Kevin Almeroth, Richard Kemmerer, Giovanni Vigna

Watermarking and DRM

Rights Protection for Data Cubes 359
Jie Guo, Yingjiu Li, Robert H. Deng, Kefei Chen

An Efficient Probabilistic Packet Marking Scheme (NOD-PPM) 373
Huifang Yin, Jun Li

Intrusion Detection and Worms

Resistance Analysis to Intruders' Evasion of Detecting Intrusion 383
Jianhua Yang, Yongzhong Zhang, Shou-Hsuan Stephen Huang

A Wireless Intrusion Detection System for Secure Clustering and Routing in Ad Hoc Networks 398
Luciano Bononi, Carlo Tacconi

Anomaly Intrusion Detection Based on Clustering a Data Stream 415
Sang-Hyun Oh, Jin-Suk Kang, Yung-Cheol Byun, Taikyeong T. Jeong, Won-Suk Lee

Robust Reactions to Potential Day-Zero Worms Through Cooperation and Validation 427
K. Anagnostakis, S. Ioannidis, A.D. Keromytis, M.B. Greenwald

Key Exchange

An Authentication and Key Exchange Protocol for Secure Credential Services 443
SeongHan Shin, Kazukuni Kobara, Hideki Imai

A Non-malleable Group Key Exchange Protocol Robust Against Active Insiders 459
Yvo Desmedt, Josef Pieprzyk, Ron Steinfeld, Huaxiong Wang

Security Protocols and Formal Methods

Formalising Receipt-Freeness 476
H.L. Jonker, E.P. de Vink

Enhancing the Security and Efficiency of 3-D Secure 489
Mohammed Assora, Ayoub Shirvani

Designing and Verifying Core Protocols for Location Privacy 502
David von Oheimb, Jorge Cuellar

Information Systems Security

Delegation in a Distributed Healthcare Context: A Survey of Current Approaches 517
Mila Katzarova, Andrew Simpson

Managing Information Systems Security: Critical Success Factors and Indicators to Measure Effectiveness 530
Jose M Torres, Jose M Sarriegi, Javier Santos, Nicolás Serrano

Author Index 547

Extending .NET Security to Unmanaged Code

Patrick Klinkoff[1], Christopher Kruegel[1], Engin Kirda[1], and Giovanni Vigna[2]

[1] Secure Systems Lab
Technical University Vienna
{pk, chris, ek}@seclab.tuwien.ac.at
[2] Department of Computer Science
University of California, Santa Barbara
vigna@cs.ucsb.edu

Abstract. The number of applications that are downloaded from the Internet and executed on-the-fly is increasing every day. Unfortunately, not all of these applications are benign, and, often, users are unsuspecting and unaware of the intentions of a program. To facilitate and secure this growing class of mobile code, Microsoft introduced the .NET framework, a new development and runtime environment where machine-independent byte-code is executed by a virtual machine. An important feature of this framework is that it allows access to native libraries to support legacy code or to directly invoke the Windows API. Such native code is called *unmanaged* (as opposed to *managed* code). Unfortunately, the execution of unmanaged native code is not restricted by the .NET security model, and, thus, provides the attacker with a mechanism to completely circumvent the framework's security mechanisms.

The approach described in this paper uses a sandboxing mechanism to prevent an attacker from executing malicious, unmanaged code that is not permitted by the security policy. Our sandbox is implemented as two security layers, one on top of the Windows API and one in the kernel. Also, managed and unmanaged parts of an application are automatically separated and executed in two different processes. This ensures that potentially unsafe code can neither issue system calls not permitted by the .NET security policy nor tamper with the memory of the .NET runtime. Our proof-of-concept implementation is transparent to applications and secures unmanaged code with a generally acceptable performance penalty. To the best of our knowledge, the presented architecture and implementation is the first solution to secure unmanaged code in .NET.

1 Introduction

With the growth of the Internet, applications are increasingly downloaded from remote sources, such as Web sites, and executed on-the-fly. Often, little or no knowledge exists about the author or her intentions. Therefore, users are susceptible to executing potentially malicious programs on their computers. Malicious programs contain code that executes in any unauthorized or undesirable way.

S.K. Katsikas et al. (Eds.): ISC 2006, LNCS 4176, pp. 1–16, 2006.

To secure users and increase the proliferation of mobile code, Microsoft recently introduced a new development and runtime framework called .NET [5]. This framework leverages the previous experiences gathered with the Java virtual machine concepts and includes a fine-grained security model that allows one to control the level of access associated with software built upon .NET. These applications are referred to as composed of *managed* code. The model significantly limits the damage that can be caused by malicious code. To address the important problem of backward compatibility and legacy code support, .NET also offers a mechanism to tie in native libraries. These libraries, however, execute outside of the .NET security model, and therefore are called *unmanaged code*. As a consequence, the usage of this feature in .NET applications may allow an attacker to completely circumvent the framework's security mechanisms, leading to the unrestricted execution of arbitrary code. This security problem is important because the use of unmanaged code will probably be common in future Windows .NET applications. Millions of lines of legacy native Windows code exist that will need to be integrated and supported over the next decade. Also, software engineering research [10] has shown that it is not realistic to expect existing applications to be entirely rewritten from scratch in order to take advantage of the features of a new language.

This paper describes our approach to extend the current .NET security model to native (unmanaged) code invoked from .NET. To this end, we use a sandboxing mechanism that is based on the analysis of Windows API and system call invocations to enforce the .NET security policy. Our approach ensures that all unmanaged code abides by the security permissions granted by the framework. Our primary contributions are as follows:

- Extension of existing sandboxing methods to .NET unmanaged code invocations.
- Two-step authorization of system calls by placing the security layer in the Windows API and the enforcement mechanisms in a loadable kernel driver.
- Separation of untrusted native library and trusted managed code into two separate processes by way of .NET remoting.

The paper is structured as follows. The next section provides an overview of the .NET framework and its security-relevant components. Section 3 introduces the design of our proposed system. Section 4 discusses the evaluation of the security and performance of the system and shows that our approach is viable. Section 5 presents related work. Finally, Section 6 outlines future work and concludes the paper.

2 Overview of the .NET Framework

Microsoft's .NET framework is an implementation of the Common Language Infrastructure (CLI) [6], which is the open, public specification of a runtime environment and its executable code. A part of the CLI specification describes the Common Type System (CTS), which defines how types are declared and used in the runtime. An important property of the .NET framework is that it is type-safe. Type safety ensures that memory accesses are performed only in well-defined ways, and no operation will be applied to a variable of the wrong type. That is, any declared variable will always reference an object of either that type or a subtype of that type. In particular, type safety prevents a non-pointer from being dereferenced to access memory. Without type safety, a program could construct an integer value that corresponds to a target address, and then use it as a pointer to reference an arbitrary location in memory. In addition to type safety, .NET also provides memory safety, which ensures that a program cannot access memory outside of properly allocated objects. Languages such as C are neither type-safe nor memory-safe. Thus, arbitrary memory access and type casts are possible, potentially leading to security vulnerabilities such as buffer overflows.

The runtime environment can enforce a variety of security restrictions on the execution of a program by relying on type and memory safety. This makes it possible to run multiple .NET programs with different sets of permissions in the same process (on the same virtual machine). To specify security restrictions, the CLI defines a security model that is denoted as Code Access Security (CAS) [9]. CAS uses *evidence* provided by the program and security policies configured on the machine to generate permissions set associated with the application. Security relevant operations (for example, file access) create corresponding permission objects, which are tested with respect to the granted permission set. If the permission is not found in the granted set, the action is not permitted and a security exception is thrown. Otherwise, the operation continues.

Managed code executes under the control of the runtime, and, therefore, has access to its services (such as memory management, JIT compilation, or type and memory safety). In addition, the runtime can also execute *unmanaged code*, which has been compiled to run on a specific hardware platform and cannot directly utilize the runtime. In general, developers will prefer managed code to benefit from the services offered by the runtime. However, there are cases in which unmanaged code is needed. For example, the invocation of unmanaged code is necessary when there are external functions that are not written in .NET. Arguably, the most important library of unmanaged functions is the Windows API, which contains thousands of routines that provide access to most aspects of the Windows operating system.

To support interoperability with existing code written in languages such as C or C++ (e.g., the Windows API), the CLI uses a mechanism called *platform invoke service* (P/Invoke). This service allows for invocation of code residing in native libraries. Because code in native libraries can modify the security state of the user's environment, the .NET permission to call native code is equal to full trust [18]. Furthermore, native code launched by P/Invoke is run within the same process as the .NET CIL, and, as a consequence, malicious native code could modify the state of the .NET runtime itself. Microsoft suggests to only allow P/Invoke to be used to execute highly-trusted code. Unfortunately, users generally cannot determine the trust level of an application and will likely grant access also to non-trustworthy applications.