US008074115B2

(12) **United States Patent**
Stolfo et al.

(10) **Patent No.: US 8,074,115 B2**
(45) **Date of Patent: Dec. 6, 2011**

(54) **METHODS, MEDIA AND SYSTEMS FOR DETECTING ANOMALOUS PROGRAM EXECUTIONS**

(75) Inventors: **Salvatore J. Stolfo**, Ridgewood, NJ (US); **Angelos D. Keromytis**, New York, NY (US); **Stelios Sidiroglou**, New York, NY (US)

(73) Assignee: **The Trustees of Columbia University in the City of New York**, New York, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **12/091,150**

(22) PCT Filed: **Oct. 25, 2006**

(86) PCT No.: **PCT/US2006/041591**

§ 371 (c)(1),
(2), (4) Date: **Jun. 15, 2009**

(87) PCT Pub. No.: **WO2007/050667**

PCT Pub. Date: **May 3, 2007**

(65) **Prior Publication Data**

US 2010/0023810 A1 Jan. 28, 2010

Related U.S. Application Data

(60) Provisional application No. 60/730,289, filed on Oct. 25, 2005.

(51) **Int. Cl.**
G06F 11/00 (2006.01)

(52) **U.S. Cl.** **714/38.1**

(58) **Field of Classification Search** 714/2–10, 714/25–29, 32, 33, 37–39, 47

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,968,113	A *	10/1999	Haley et al.	714/38
6,079,031	A *	6/2000	Haley et al.	714/38
6,154,876	A *	11/2000	Haley et al.	717/133
7,155,708	B2 *	12/2006	Hammes et al.	717/155
7,490,268	B2	2/2009	Keromytis et al.	
7,496,898	B1 *	2/2009	Vu	717/127
7,639,714	B2	12/2009	Stolfo et al.	
2005/0108562	A1 *	5/2005	Khazan et al.	713/200

OTHER PUBLICATIONS

Hangal et al., Tracking down software bugs using automatic anomaly detection, Proceedings of the 24the international conference on software engineering, May 2002, pp. 291-301.*

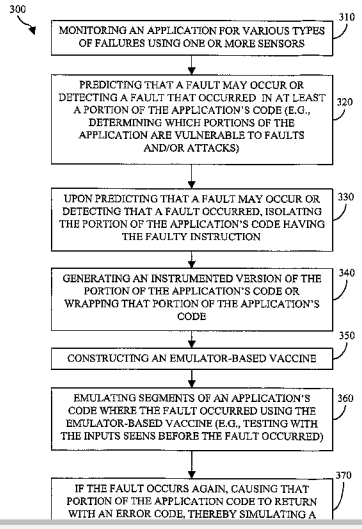
(Continued)

Primary Examiner — Nadeem Iqbal
(74) *Attorney, Agent, or Firm* — Byrne Poh LLP

(57) **ABSTRACT**

Methods, media, and systems for detecting anomalous program executions are provided. In some embodiments, methods for detecting anomalous program executions are provided, comprising: executing at least a part of a program in an emulator; comparing a function call made in the emulator to a model of function calls for the at least a part of the program; and identifying the function call as anomalous based on the comparison. In some embodiments, methods for detecting anomalous program executions are provided, comprising: modifying a program to include indicators of program-level function calls being made during execution of the program; comparing at least one of the indicators of program-level function calls made in the emulator to a model of function calls for the at least a part of the program; and identifying a function call corresponding to the at least one of the indicators as anomalous based on the comparison.

42 Claims, 8 Drawing Sheets



OTHER PUBLICATIONS

Chan et al., A machine learning approach to anomaly detection, Technical Report, Dept. of computer science, Florida institute of technology, Mar. 2003, pp. 1-13.*
M. Chew and D. Song, Mitigating Buffer Overflows by Operating System Randomization, Technical Report CMUCS-02-197, Carnegie Mellon University, Dec. 2002.
V. Prevelakis, A Secure Station for Network Monitoring and Control, In Proceedings of the 8th USENIX Security Symposium, Aug. 1999.
J. Reynolds, J. Just, L. Clough, and R. Maglich, On-Line Intrusion Detection and Attack Prevention Using Diversity, Generate-and-Test, and Generalization, In Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS), Jan. 2003.
H. Shacham, M. Page, B. Pfaff, E. Goh, N. Modadugu, and D. Boneh, on the Effectiveness of Address-Space Randomization, In Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS), pp. 298-307, Oct. 2004.
S. Sidiroglou, M. Locasto, S. Boyd, and A. Keromytis, Building A Reactive Immune System for Software Services, In Proceedings of the 11th USENIX Annual Technical Conference, Apr. 2005.
M. Stamp, Risk of Monoculture, Communications of the ACM, 47(3):120, Mar. 2004.
Using Network-Based Application Recognition and ACLs for Blocking the "Code Red" Worm, Technical report, Cisco Systems, Inc.
Aleph One, Smashing the stack for fun and profit, Phrack, 7(49), 1996.
K. Ashcraft and D. Engler, Detecting Lots of Security Holes Using System-Specific Static Analysis, In Proceedings of the IEEE Symposium on Security and Privacy, May 2002.
S. M. Bellovin, Distributed Firewalls, ;login: magazine, special issue on security, Nov. 1999.
M. Bhattacharyya, M. G. Schultz, E. Eskin, S. Hershkop, and S. J. Stolfo, MET: An Experimental System for Malicious Email Tracking, In Proceedings of the New Security Paradigms Workshop (NSPW), pp. 1-12, Sep. 2002.
Bulba and Kil3r, Bypassing StackGuard and StackShield, Phrack, 5(56), May 2000.
B. Chess, Improving Computer Security Using Extended Static Checking, In Proceedings of the IEEE Symposium on Security and Privacy, May 2002.
M. Christodorescu and S. Jha, Static Analysis of Executables to Detect Malicious Patterns, In Proceedings of the 12th USENIX Security Symposium, pp. 169-186, Aug. 2003.
F. Cohen, Computer Viruses: Theory and Practice, Computers & Security, 6:22-35, Feb. 1987.
C. Cowan, M. Barringer, S. Beattie, and G. Kroah-Hartman, Formatguard: Automatic protection from printf format string vulnerabilities, In Proceedings of the 10th USENIX Security Symposium, Aug. 2001.
C. Cowan, S. Beattie, C. Pu, P. Wagle, and V. Gligor, SubDomain: Parsimonious Security for Server Appliances, In Proceedings of the 14th USENIX System Administration Conference (LISA 2000), Mar. 2000.
C. Cowan, C. Pu, D. Maier, H. Hinton, J. Walpole, P. Bakke, S. Beattie, A. Grier, P. Wagle, and Q. Zhang, Stackguard: Automatic adaptive detection and prevention of buffer-overflow attacks, In Proceedings of the 7th USENIX Security Symposium, Jan. 1998.
D. Engler and K. Ashcraft, RacerX: Effective, Static Detection of Race Conditions and Deadlocks, Proceedings of ACM SOSP, Oct. 2003.
S. Forrest, A. Somayaji, and D. Ackley, Building Diverse Computer Systems, In Proceedings of the 6th HotOS Workshop, 1997.
M. Frantzen and M. Shuey, StackGhost: Hardware facilitated stack protection, In Proceedings of the 10th USENIX Security Symposium, pp. 55-66, Aug. 2001.
T. Garfinkel, Traps and Pitfalls: Practical Problems in System Call Interposition Based Security Tools, In Proceedings of the Symposium on Network and Distributed Systems Security (SNDSS), pp. 163-176, Feb. 2003.
I. Goldberg, D. Wagner, R. Thomas, and E. Brewer, A Secure Environment for Untrusted Helper Applications, In Proceedings of the 6th USENIX Security Symposium, Jul. 1996.
S. Ioannidis, A. Keromytis, S. Bellovin, and J. Smith, Implementing a Distributed Firewall, In Proceedings of the ACM Computer and Communications Security (CCS) Conference, pp. 190-199, Nov. 2000.
R. Janakiraman, M. Waldvogel, and Q. Zhang, Indra: A peer-topeer approach to network intrusion detection and prevention, In Proceedings of the IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), Workshop on Enterprise Security, Jun. 2003.
R. Jones and P. Kelly, Backwards-compatible bounds checking for arrays and pointers in C programs, In Third International Workshop on Automated Debugging, 1997.
J. Just, L. Clough, M. Danforth, K. Levitt, R. Maglich, J. C. Reynolds, and J. Rowe, Learning Unknown Attacks—A Start, In Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection (RAID), Oct. 2002.
J. Kephart, A Biologically Inspired Immune System for Computers, In Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, pp. 130-139. MIT Press, 1994.
M. Kodialam and T. V. Lakshman, Detecting Network Intrusions via Sampling: A Game Theoretic Approach, In Proceedings of the 22nd Annual Joint Conference of IEEE Computer and Communication Societies (INFOCOM), Apr. 2003.
D. Larochelle and D. Evans, Statically Detecting Likely Buffer Overflow Vulnerabilities, In Proceedings of the 10th Security Symposium, pp. 177-190, Aug. 2001.
E. Larson and T. Austin, High Coverage Detection of Input-Related Security Faults, In Proceedings of the 12th Security Symposium, pp. 121-136, Aug. 2003.
K. Lhee and S. J. Chapin, Type-Assisted Dynamic Buffer Overflow Detection. In Proceedings of the 11th Security Symposium, pp. 81-90, Aug. 2002.
M.-J. Lin, A. Ricciardi, and K. Marzullo, A New Model for Availability in the Face of Self-Propagating Attacks, In Proceedings of the New Security Paradigms Workshop, Nov. 1998.
A. J. Malton, The Denotational Semantics of a Functional Tree-Manipulation Language, Computer Languages, 19 (3):157-168, 1993.
T. C. Miller and T. de Raadt, strlcpy and strlcat: Consistent, Safe, String Copy and Concatenation, In Proceedings of the USENIX Annual Technical Conference, Freenix Track, Jun. 1999.
D. Moore, C. Shanning, and K. Claffy, Code-Red: a case study on the spread and victims of an Internet worm. In Proceedings of the 2nd Internet Measurement Workshop (IMW), pp. 273-284, Nov. 2002.
D. Moore, C. Shannon, G. Voelker, and S. Savage, Internet Quarantine: Requirements for Containing Self-Propagating Code, In Proceedings of the IEEE Infocom Conference, Apr. 2003.
C. Nachenberg, Computer Virus-Coevolution, Communications of the ACM, 50(1):46-51, 1997.
D. Nojiri, J. Rowe, and K. Levitt, Cooperative Response Strategies for Large Scale Attack Mitigation, In Proceedings of the 3rd DARPA Information Survivability Conference and Exposition (DISCEX), pp. 293-302, Apr. 2003.
D. S. Peterson, M. Bishop, and R. Pandey, A Flexible Containment Mechanism for Executing Untrusted Code, In Proceedings of the 11th USENIX Security Symposium, pp. 207-225, Aug. 2002.
M. Prasad and T. Chiueh, A Binary Rewriting Defense Against Stack-based Buffer Overflow Attacks, In Proceedings of the USENIX Annual Technical Conference, pp. 211-224, Jun. 2003.
V. Prevelakis and D. Spinellis, Sandboxing Applications, In Proceedings of the USENIX Technical Annual Conference, Freenix Track, pp. 119-126, Jun. 2001.
N. Provos, M. Friedl, and P. Honeyman, Preventing Privilege Escalation, In Proceedings of the 12th USENIX Security Symposium, pp. 231-242, Aug. 2003.
J. Reynolds, J. Just, E. Lawson, L. Clough, and R. Maglich, The Design and Implementation of an Intrusion Tolerant System, In Proceedings of the International Conference on Dependable Systems and Networks (DSN), Jun. 2002.
M. Rosenblum, E. Bugnion, S. Devine, and S. A. Herrod, Using the

R. Sekar, V. Venkatakrishnan, S. Basu, S. Bhatkar, and D. C. DuVaney, Model-Carrying Code: A Practice Approach for Safe Execution of Untrusted Applications, in Proceedings of ACM SOSP, Oct. 2003.
N. Nethercote and J. Seward, Valgrind: A Framework for Heavy-weight Dynamic Binary Instrumentation, PLDI '07, Jun. 2007.
J. F. Shoch and J. A. Hupp, The "worm" programs—early experiments with a distributed computation, Communications of the ACM, 22(3):172-180, Mar. 1982.
Song, R. Malan, and R. Stone.,A Snapshot of Global Internet Worm Activity, Technical report, Arbor Networks, Nov. 2001.
E. H. Spafford, The Internet Worm Program: An Analysis, Technical Report CSD-TR-823, Purdue University, 1988.
S. Staniford, V. Paxson, and N. Weaver, How to Own the Internet in Your Spare Time, In Proceedings of the 11th USENIX Security Symposium, pp. 149-167, Aug. 2002.
T. Toth and C. Kruegel, Connection-history Based Anomaly Detection, In Proceedings of the IEEE Workshop on Information Assurance and Security, Jun. 2002.
H. Toyoizumi and A. Kara, Predators: Good Will Mobile Codes Combat against Computer Viruses, In Proceedings of the New Security Paradigms Workshop (NSPW), pp. 13-21, Sep. 2002.
J. Twycross and M. M. Williamson, Implementing and testing a virus throttle, In Proceedings of the 12th USENIX Security Symposium, pp. 285-294, Aug. 2003.
G. Venkitachalam and B.-H. Lim, Virtualizing i/o devices on vmware workstation's hosted virtual machine monitor.
C. Wang, J. C. Knight, and M. C. Elder, on Computer Viral Infection and the Effect of Immunization, In Proceedings of the 16th Annual Computer Security Applications Conference (ACSAC), pp. 246-256, 2000.
A. Whitaker, M. Shaw, and S. D. Gribble, Scale and Performance in the Denali Isolation Kernel, In Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI), Dec. 2002.
J. Wilander and M. Kamkar, A Comparison of Publicly Available Tools for Dynamic Intrusion Prevention, In Proceedings of the Symposium on Network and Distributed Systems Security (SNDSS), pp. 123-130, Feb. 2003.
M. Williamson, Throttling Viruses: Restricting Propagation to Defeat Malicious Mobile Code, Technical Report HPL-2002-172, HP Laboratories Bristol, 2002.
C. C. Zou, L. Gao, W. Gong, and D. Towsley, Monitoring and Early Warning for Internet Worms, In Proceedings of the 10th ACM International Conference on Computer and Communications Security (CCS), pp. 190-199, Oct. 2003.
C. C. Zou, W. Gong, and D. Towsley, Code Red Worm Propagation Modeling and Analysis, In Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS), pp. 138-147, Nov. 2002.
S. Hangal and M. Lam, Tracking Down Software Bugs Using Automatic Anomaly Detection, ICSE '02. May 19-25, 2002, pp. 291-301.
P. Chan, M. Mahoney, and M. Arshad, A Machine Learning Approach to Anomaly Detection, Technical Report CS-2003-06, Department of Computer Sciences, Florida Institute of Technology, Mar. 29, 2003.
Interational Search Report and Written Opinion, International Application No. PCT/US06/41591, dated Jun. 25, 2008.
F. Apap, A. Honig, S. Hershkop, E. Eskin, and S. Stolfo, Detecting malicious software by monitoring anomalous windows registry accesses, Proceedings of the Fifth International Symposium on Recent Advances in Intrusion Detection (RAID 2002), 2002.
D. Denning, An intrusion detection model, IEEE Transactions on Software Engineering, SE-13:222-232, Feb. 1987.
E. Eskin, Anomaly detection over noisy data using learned probability distributions, Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000), 2000.
S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff, A sense of self for unix processes, Proceedings of the IEEE Symposium on Research in Security and Privacy, pp. 120-128, 1996.
N. Friedman and Y. Singer, Efficient bayesian parameter estimation
S. Hofmeyr, S. Forrest, and A. Somayaji, Intrusion detection using sequences of system calls, Journal of Computer Security, 6:151-180, 1998.
H. Javitz and A. Valdes, The nides statistical component: Description and justification, Technical Report, SRI International, Computer Science Laboratory, 1993.
W. Lee, S. Stolfo, and P. Chan, Learning patterns from unix processes execution traces for intrusion detection, AAAI Workshop on AI Approaches to Fraud Detection and Risk Management, pp. 50-56, 1997.
W. Lee, S. Stolfo, and K. Mok, A data mining framework for building intrusion detection models, IEEE Symposium on Security and Privacy, pp. 120-132, 1999.
W. Lee, S. Stolfo, and K. Mok, Data mining in work flow environments: Experiences in intrusion detection, Proceedings of the 1999 Conference on Knowledge Discovery and Data Mining (KDD-99), 1999.
M. Mahoney and P. Chan, Detecting novel attacks by identifying anomalous network packet headers, Technical Report CS-2001-2, 2001.
B. Scholkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, Estimating the support of a high dimensional distribution, Neural Computation, 13(7):1443-1472, 2001.
C. Warrender, S. Forrest, B. Pearlmutter, Detecting intrusions using system calls: Alternative data models, IEEE Symposium on Security and Privacy, pp. 133-145, 1999.
A. Honig, A. Howard, E. Eskin, and S. Stolfo, Adaptive model generation: An architecture for the deployment of data mining-based intrusion detection systems, in Data Mining for Security Applications, Kluwer, 2002.
S. White, Open problems in computer virus reseach, in Virus Bulletin Conference, 1998.
CERT Advisory CA-2003-21: W32/Blaster Worm, http://www.cert.org/advisories/CA-2003-20.html, Aug. 2003.
A. Baratloo, N. Singh, and T. Tsai, Transparent Run-Time Defense Against Stack Smashing Attacks, In Proceedings of the Annual Technical Conference, Jun. 2000.
E. G. Barrantes, D. H. Ackley, S. Forrest, T. S. Palmer, D. Stefanovic, and D. D. Zovi, Randomized Instruction Set Emulation to Distrupt Binary Code Injection Attacks, in 10th ACM Conference on Computer and Communications Security (CCS), Oct. 2003.
D. Bruening, T. Garnett, and S. Amarasinghe, An Infrastructure for Adaptive Dynamic Optimization, In Proceedings of the International Symposium on Code Generation and Optimization, pp. 265-275, 2003.
G. Candea and A. Fox, Crash-Only Software, in Proceedings of the 9th Workshop on Hot Topics in Operating Systems, May 2003.
H. Chen and D. Wagner, MOPS: an Infrastructure for Examining Security Properties of Software, In Proceedings of the ACM Computer and Communications Security (CCS) Conference, pp. 235-244, Nov. 2002.
S. A. Crosby and D. S. Wallach, Denial of Service via Algorithmic Complexity Attacks, In Proceedings of the 12th USENIX Security Symposium, pp. 29-44, Aug. 2003.
B. Demsky and M. C. Rinard, Automatic Detection and Repair of Errors in Data Structures, In Proceedings of the 18th Annual ACM SIGPLAN Conference on Object Oriented Programming, Systems, Languages, and Applications, Oct. 2003.
G. W. Dunlap, S. King, S. Cinar, M. A. Basrai, and P. M. Chen, ReVirt: Enabling Intrusion Analysis Through Virtual-Machine Logging and Replay, In Proceedings of the Symposium on Operating Systems Design and Implementation (OSDI), Feb. 2002.
C. Cowan et al., StackGuard: Automatic Adaptive Detection and Prevention of Buffer-Overow Attacks, In Proceedings of the 7th Security Symposium, Jan. 1998.
T. Garfinkel and M. Rosenblum, A Virtual Machine Introspection Based Architecture for Intrusion Detection, in 10th ISOC Symposium on Network and Distributed Systems Security (SNDSS), Feb. 2003.
T. Jim, G. Morrisett, D. Grossman, M. Hicks, J. Cheney, and Y. Wang,

G. S. Kc, A. D. Keromytis, and V. Prevelakis, Countering Code-Injection Attacks With Instruction-Set Randomization, in 10th ACM Conference on Computer and Communications Security (CCS), Oct. 2003.
S. T. King and P. M. Chen, Backtracking Intrusions, In 19th ACM Symposium on Operating Systems Principles (SOSP), Oct. 2003.
S. T. King, G. Dunlap, and P. Chen, Operating System Support for Virtual Machines, In Proceedings of the Annual Technical Conference, Jun. 2003.
V. Kiriansky, D. Bruening, and S. Amarasinghe, Secure Execution Via Program Shepherding, In Proceedings of the 11th Security Symposium, Aug. 2002.
D. Mosberger and T. Jin, httperf: A tool for measuring web server performance, In First Workshop on Internet Server Performance, pp. 59-67, ACM, Jun. 1998.
N. Nethercote and J. Seward, Valgrind: A Program Supervision Framework, In Electronic Notes in Theoretical Computer Science, vol. 89, 2003.
J. Newsome and D. Dong, Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software, In The 12th Annual Network and Distributed System Security Symposium, Feb. 2005.
J. Oplinger and M. S. Lam, Enhancing Software Reliability with Speculative Threads, In Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS X), Oct. 2002.
N. Provos, Improving Host Security with System Call Policies, In Proceedings of the 12th USENIX Security Symposium, pp. 257-272, Aug. 2003.
M. Rinard, C. Cadar, D. Dumitran, D. Roy, and T. Leu, A Dynamic Technique for Eliminating Buffer Overow Vulnerabilities (and Other Memory Errors), In Proceedings 20th Annual Computer Security Applications Conference (ACSAC), Dec. 2004.
M. Rinard, C. Cadar, D. Dumitran, D. Roy, T. Leu, and J. W Beebee, Enhancing Server Availability and Security Through Failure-Oblivious Computing, In Proceedings 6th Symposium on Operating Systems Design and Implementation (OSDI), Dec. 2004.
A. Rudys and D. S. Wallach, Transactional Rollback for Language-Based Systems, In ISOC Symposium on Network and Distributed Systems Security (SNDSS), Feb. 2001.
A. Rudys and D. S. Wallach, Termination in Language-based Systems, ACM Transactions on Information and System Security, 5(2), May 2002.
S. Sidiroglou and A. D. Keromytis, A Network Worm Vaccine Architecture. In Proceedings of the IEEE Workshop on Enterprise Technologies: Infrastructure for Collaborative Enterprises (WET-ICE), Workshop on Enterprise Security, pp. 220-225, Jun. 2003.
A. Smirnov and T. Chiueh, DIRA: Automatic Detection, Identication, and Repair of Control-Hijacking Attacks, In The 12th Annual Network and Distributed System Security Symposium, Feb. 2005.
G. E. Suh, J. W. Lee, D. Zhang, and S. Devadas, Secure program execution via dynamic information flow tracking, SIGOPS Oper. Syst. Rev., 38(5):85-96, 2004.
T. Toth and C. Kruegel, Accurate Buffer Overflow Detection via Abstract Payload Execution, In Proceedings of the 5th Symposium on Recent Advances in Intrusion Detection (RAID), Oct. 2002.
N. Wang, M. Fertig, and S. Patel, Y-Branches: When You Come to a Fork in the Road, Take It, In Proceedings of the 12th International Conference on Parallel Architectures and Compilation Techniques, Sep. 2003.
J. Yin, J.-P. Martin, A. Venkataramani, L. Alvisi, and M. Dahlin, Separating Agreement from Execution for Byzantine Fault Tolerant Services, in Proceedings of ACM SOSP, Oct. 2003.
A. Avizienis, The n-version approach to fault-tolerant software, IEEE Transactionson Software Engineering, 11 (12):1491-1501, 1985.
S. Bhatkar, D. C. DuVarney, and R. Sekar, Address Obfuscation: an Efficient Approach to Combat a Broad Range of Memory Error Exploits, In Proceedings of the 12th Security Symposium, pp. 105-120, Aug. 2003.
S. Brilliant, J. C. Knight, and N. G. Leveson, Analysis of Faults in an N-Version Software Experiment, IEEE Transactions on Software Engineering, 16(2), Feb. 1990.

* cited by examiner

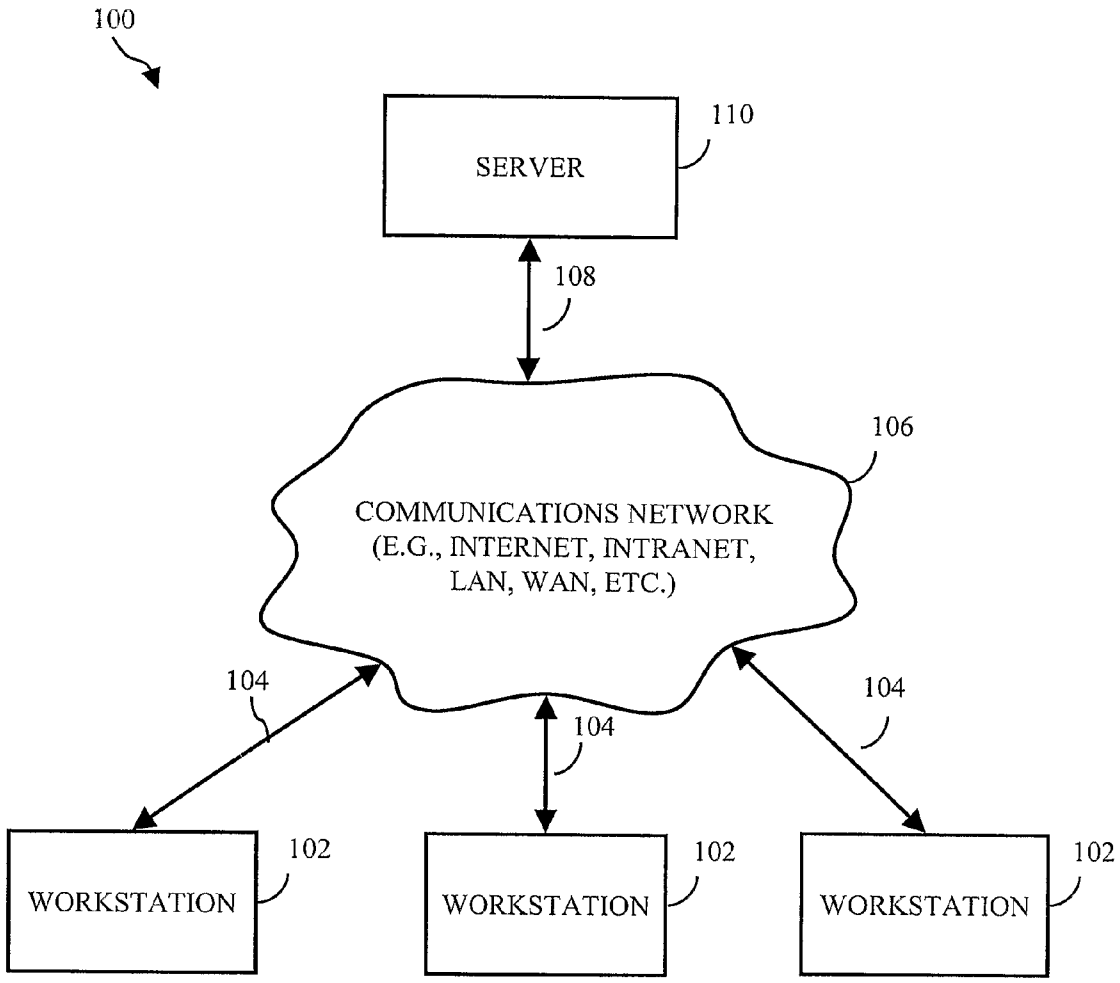


FIG. 1

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.