Analysis and Detection of Computer Viruses and Worms: An Annotated Bibliography

Prabhat K. Singh, Arun Lakhotia
Center for Advanced Computer Studies
University of Louisiana, Lafayette, LA 70504,
{pks3539, arun}@cacs.louisiana.edu

Abstract:

This annotated bibliography reviews research in analyzing and detecting computer viruses and worms. This document focuses on papers that give information about techniques and systems detecting malicious code.

The format of the entries is as follows:
Book/Thesis:
Authors, "*article Title*", Publisher, City, State, Year
Electronic Media:
Authors, "article Title", *URL*
Conference Proceedings:
Authors, "Article Title", *Conference Title*, Edition, City, State, pp., Year
Technical Reports:
Authors, "Title", TR. #, Dept., Univ.

Leonard M. Adleman, "An Abstract Theory of Computer Viruses," Lecture Notes in Computer Science, Vol. 403, Advances in Computing-Crypto '88.
This paper applies formal computability theory to viruses. It presents definition for computer viruses based on set theory. Viruses have been broken up into benign, disseminating, malicious, and Epeian categories. It proves that "detecting viruses is quite untractable". It identifies several areas of possible research including complexity theoretic and program size theoretic aspects of computer viruses, protection mechanisms and development of other models.

J. Bergeron & M. Debbabi & M. M. Erhioui & B. Ktari. "Static Analysis of Binary Code to Isolate Malicious Behaviors," Proceedings of the IEEE 4th International Workshop on Enterprise Security (WETICE'99), Stanford University, California, USA, June 16-18, 1999.
This paper addresses the problem of static slicing on binary executables for the purpose of detecting malicious code in commercial off-the shelf software components. The paper first defines a malicious code. To analyze malicious code, the executable is first disassembled and passed through a series of transformations. These transformations aid in getting a high level imperative representation of the code. This leads to improved analyzability while preserving the original semantics. Next, the program is sliced to extract code segments critical from standpoint of security. The behavior of these segments is reviewed for malicious characteristics.

J. Bergeron, M. Debbabi et al., "Detection of Malicious Code in COTS software: A short Survey," First International Software Assurance Certification Conference (ISACC'99), Washington DC, Mar. 1999
This paper describes the main characteristics of malicious code and proposes taxonomy for the existing varieties. A formal definition of malicious code has been given. A new taxonomy that is oriented towards the goal of detecting malicious code has been defined. Different static, dynamic analysis methods and ad hoc techniques have been discussed. It discusses several techniques to detect malicious code in commercial-off–the-shelf software products. The paper concludes by looking at the advantages and disadvantages of static analysis over dynamic analysis methods.

J. Bergeron et al., "Static Detection of Malicious Code in Executable Programs," Symposium on Requirements Engineering for Information Security (SREIS'01), Indianapolis, Indiana, USA, March 5-6, 2001.
This paper approaches the problem of detection of malicious code in executable programs using static analysis. It involves three steps: the generation of intermediate representation, analyzing the control and data flows, and then doing static verification. Static verification consists of comparing a security policy to the

output of the analysis phase. A brief description of a prototype tool is also given.

Matt Bishop, "An Overview of Computer Viruses in a Research Environment," Technical Report bishop92overview, 1992
http://www.ja.net/CERT/JANET-CERT/
This paper analyzes virus in a general framework. A brief history of computer viruses is presented and any presence of threat relevant to research and development systems has been investigated. It examines several specific areas on vulnerability in research-oriented systems.

Vesselin Bontchev, "Vircing" the Invircible," May 1995, Not published in print, available at http://www.claws-and-paws.com/virus/papers/
This is a detailed technical evaluation of an existing antiviral software, called Invircible. It reflects on the degree of responsibility, that an antivirus company needs to shoulder, while it provides its product information to users. The author has detailed on the tests and procedures he has used to evaluate Invircible product's claimed features and proves that the claims are far from reality. The write-up is old in terms of providing techniques for antivirus software functionality but is still informative on giving ideas on designing antivirus software.

Vesselin Bontchev, "Macro Virus Identification Problems," 7th International Virus Bulletin. Conference, pp. 175-196, 1997
This paper discusses some interesting theoretical problems to anti-virus software. Two viral sets of macros can have common subsets or one of the sets could be a subset of the other. The paper discusses the problems caused by this. It emphasizes the difficulties that could be exploited by the virus writers and methods, which could be followed to tackle it.

Vesselin Bontchev, "Methodology of Computer Anti-Virus Research," Doctoral Thesis, Faculty of Informatics, University of Hamburg, 1998
This thesis is a detailed writing on computer viruses. It can be treated as a definitive text on understanding and dealing with computer viruses. The important topics discussed in this work include classification and analysis of computer viruses, state of art in anti-virus software, possible attacks against anti-virus software, test methods for anti-virus software systems and social aspects of virus problem. It also discusses useful applications of self-replicating software.

Vesselin Bontchev, "Analysis and maintenance of a clean virus library," Proc. 3rd Int. Virus Bull. Conf., pp. 77-89, 1993
This provides the methods adopted to facilitate the maintenance of large amounts of different virus samples for the sake of anti-virus research. The paper presents guidelines and procedures used to maintain virus collection at the university of Hamburg's Virus Test Center.

Vesselin Bontchev, "The "Pros" and "Cons" of WordBasic Virus Upconversion," Vesselin Bontchev, Proceedings of the 8th International Virus Bulletin Conference, pp. 153-172, 1998
This paper discusses the ethical problem faced by anti-virus researchers due to the automatic Upconversion of WordBasic Viruses to Visual Basic for Applications version 5. Since a macro virus written in one language has be automatically converted to another language it is yet another unique virus. Due to this inherent feature of MS Office 97, virus researchers have to create new virus to prepare an antidote. A side effect of this activity has reportedly been that these upconverts are created and "officially" listed as existing in some anti-virus product stimulates their creation and distribution by the virus exchange people. The author has given suggested solutions for this problem.

Vesselin Bontchev, "Future Trends in Virus Writing," 4th International Virus Bulletin Conference, pp. 65-82, 1994.
This paper summarizes some ideas that are likely to be used by virus writers in the future and suggests the kind of measures that could be taken against them.

Vesselin Bontchev, "Possible Virus Attacks Against Integrity Programs And How to Prevent Them," Proceedings of the 6th International Virus Bulletin Conference, pp. 97-127, 1996.
This paper discusses the ways of attacking one of the most powerful methods of virus detection on integrity checking programs. It demonstrates what can be done against these attacks.

David M. Chess, "Virus Verification and Removal Tools and Techniques," High Integrity Computing Lab, IBM T. J. Watson Research Center, Post Office Box 218, Yorktown Heights, NY, USA, November 18, 1991.
www.research.ibm.com/antivirus/SciPapers/Chess/CHESS/chess.html

This paper describes VERV, A Prototype Virus Verifier and Remover, and a Virus Description Language for VERV.

David Chess, "Future of Viruses on the Internet," Virus Bulletin Conference, San Francisco, California, October 1-3, 1997.
This paper discusses the role of the Internet in the Virus problem. It reasons for the availability of better-equipped crisis teams that may arise due to the continued growth of the Internet. Integrated mail systems and the rise in mobile program systems on the Internet have impacted the trends in virus spread. The deployment of network aware software systems on the Internet has contributed positively to the spread of network-aware virus. The paper briefly lists some generic features of the software, which aid in virus spread.

David M Chess and Steve R. White, "An Undetectable Computer Virus," Virus Bulletin Conference, September 2000
This paper extends Fred Cohen's demonstration on computer Viruses that there is no algorithm that can perfectly detect all possible viruses. This paper points out that there are computer viruses, which no algorithm can detect, even under somewhat more liberal definition of detection.

Fred Cohen, "A Formal Definition of Computer Worms and some related Results," Computers and Security, Vol. 11, pp. 641-652 (1992)
A formal definition for computer worms has been presented. The definition is based on Turing's model of computation.

Fred Cohen, "Computational Aspects of Computer Viruses," Computers and Security, Vol. 8, No. 4., page 325, 1 June 1989.
It presents a model for defining computer viruses. It formally defines a class of sets of transitive integrity-corrupting mechanisms called "viral-sets" and explores some of their computational properties.

Fred Cohen, "Computer Viruses-Theory and Experiments," Computers and Security, Volume 6 (1987), Number 1, pp. 22-35.
This paper brought the term "computer viruses" to general attention. It describes computer viruses and also describes several experiments in each of which all system rights were granted to an attacker in under an hour.

Fred Cohen, "Computer Viruses," Ph.D. thesis, University of Southern California, 1985.
This is the first formal work in the field of computer viruses.

Fred Cohen, "Models of Practical Defenses Against Computer Viruses", Computers and Security, Vol. 8, pp. 149-160 (1989)
This paper models complexity based virus detection mechanisms, that detect modifications and thereby prevent computer viruses from causing secondary infections. These models are then used to show how to protect information in both trusted and untrusted computing bases, show the optimality of these mechanisms, and discuss some of their features. The models indicate that we can cover changes at all levels of interpretation with a unified mechanism for describing interdependencies of information in a system and discuss the ramifications of this unification in some depth.

George I. Davida, Yvo G. Desmedt, and Brian J. Matt, "Defending Systems Against Viruses through Cryptographic Authentication," Proceedings of the 1989 IEEE Symposium on Computer Security and Privacy, pp. 312-318, 1989
This paper describes the use of cryptographic authentication for controlling computer viruses. The objective is to protect against viruses infecting software distributions, updates, and programs stored or executed on a system. The authentication scheme determines the source and integrity of an executable, relying on the source to produce virus-free software. The scheme presented relies on a trusted device, the authenticator, used to authenticate and update programs and convert programs between the various formats. In addition, each user's machine uses a similar device to perform run-time checking.

M. Debbabi et al., "Dynamic Monitoring of Malicious Activity in Software Systems," Symposium on Requirements Engineering for Information Security (SREIS'01), Indianapolis, Indiana, USA, March 5-6, 2001.
The authors discuss a dynamic monitoring mechanism, comprising of a watchdog system, which dynamically enforces a security policy. The authors reason this approach by stating that static analysis technique will not be able to detect malicious code inserted after the analysis has been completed. This paper discusses a dynamic monitor called DaMon. This is capable

of stopping certain malicious actions based on the combined accesses to critical resources (files, communication ports, registry, processes and threads) according to rudimentary specifications.

Denning, P., "The Science of Computing: The Internet Worm," American Scientist, Vol. 77, No. 2, Pages 126-128, March 1989.
A write-up on the November 1988 Internet Worm incident. This paper gives a brief note on how the Internet Worm worked. It also discusses the concerns arising due to the worm incident on the networks on which commerce, transportation, utilities, defense, space flight and other critical activities depended.

Mark W. Eichin and Jon A. Rochlis
"With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988," Massachusetts Institute of Technology, Cambridge, MA, February 9, 1989
This paper defines the Internet "Worm" as a "Virus." Reasoning has been presented to substantiate this classification. It discusses the goals of the teams working on the Virus, and the methods they employed, and summarizes what the virus did and did not actually do. The paper discusses in more detail the strategies it employed, the specific attacks it used, and the effective and ineffective defenses proposed by the community against it. It describes how the group at MIT found out and reacted to the "Virus" crisis of 1988. It discusses the flaws that were exploited to attack systems and propagate across the Internet. It also enumerates methods of preventing future attacks and problems.

Gleissner W, "A Mathematical Theory for the Spread of Computer Viruses," Computers and Security, Vol. 8, No. 1, Page 35, 1 February 1989.
No description available.

J. D. Howard. "An Analysis of Security Incidents on the Internet 1989-1995," Ph.D. Dissertation, Carnegie Mellon University: Carnegie Institute of Technology, April 1997
This dissertation analyses the trends in the Internet Security by investigating 4,299 security-related incidents on the Internet reported to the CERT Coordination Center (CERT/CC) from 1989 to 1995.

C. Ko, G. Fink, and K. Levitt. "Automated detection of vulnerabilities in privileged programs by execution monitoring," In Proc. 10th Annual Computer Security Application Conf., pp. 134-144, Orlando, FL, December 1994.
This paper uses concepts of solving Intrusion Detection Problems to detect vulnerabilities in programs during execution. Since the intended behaviors of privileged programs are benign, a program policy has been developed to describe this behavior, using a program policy specification language. Specifications of privileged programs in Unix have been presented, along with a prototype execution monitor, to analyze the audit trails with respect to this specification.

Jeffrey O Kephart and Steve R. White, "Measuring and Modeling Computer Virus Prevalence," Proceedings of the 1999 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, California, pp. 2-14, May 24-25, 1993
This paper introduces two new epidemiological models of computer virus spread. Only a small fraction of all well-known viruses have appeared in real incidents, partly because many viruses are below the theoretical epidemic threshold. Models of localized software exchange can explain the observed sub-exponential rate of viral spread.

Jeffrey O. Kephart, Gregory B. Sorkin, Morton Swimmer, and Steve R. White "Blueprint for a Computer Immune System," Virus Bulletin International Conference San Francisco, California, October 1-3, 1997
Since the internet will provide a fertile medium for new breeds of computer viruses, the authors have described a immune system for computers that senses the presence of a previously unknown pathogen that within minutes, automatically derives and deploys a prescription for detecting and removing the pathogen

Jeffrey O. Kephart and Steve R. White, "Directed-Graph Epidemiological Models of Computer Viruses," Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, California, May 20-22, 1991
This paper presents a detailed study of computer virus epidemics. It presents a theoretical view of the viral propagation using deterministic and stochastic approaches. It studies the conditions under which viral epidemics are likely to occur. It argues that an imperfect defense against a computer virus can still be highly effective in preventing widespread propagation provided that

infection rate does not exceed a well-defined threshold.

Jeffrey O. Kephart, Bill Arnold, "Automatic Extraction of Computer Virus Signatures," Proceedings of the 4th Virus Bulletin International Conference, R. Ford, ed., Virus Bulletin Ltd., Abingdon, England, pp. 178-184, 1994
This paper discusses the idea of automatically identifying viral signatures from machine code using statistical methods.

Paul Kerchen, Raymond Lo, John Crossley, Grigory Elkinbard, Karl Levitt, Ronald Olsson, "Static Analysis Virus Detection Tools For Unix Systems," Proceedings of the 13th National Computer Security Conference, pages 350-- 365, 1990.
This paper proposes two heuristic tools the use static analysis and verification techniques for detecting computer viruses in a UNIX environment. The tools should be used to detect infected programs before their installation. The first tool, "detector", searches for duplicate system calls in the compiled and linked program, the second tool, "Filter", uses static analysis to determine all of the files, which a program may write to. By finding out the files to which the program can or cannot write, the program can be identified as a malicious or benign.

Sandeep Kumar, Eugene Spafford, "Generic Virus Scanner in C++," Proceedings of the 8th Computer Security Applications Conference, pp. 210-219, Coast TR 92-01, 2-4 Dec 1992
This paper discusses a generic virus detection tool designed for recognizing viruses across different platforms. The paper initially discusses various methods of virus detection and then describes a generic signature scanner as an anti-virus tool.

Butler W. Lampson, "A Note on the Confinement Problem," Xerox, Palo Alto Center, Communications of the ACM, Vol. 16, No 10. , 1973
This paper explores the problem of confining a program during its execution so that it cannot leak information to any other program except it's caller. A few ways of the above mentioned information leakage problem have been given with solutions to prevent it.

Wenke Lee and Salvatore J. Stolfo "Data Mining Approaches for Intrusion Detection," Proceedings of the 7th USENIX Security Symposium, 2000
This paper discusses research in developing general and systematic methods for Intrusion Detection. Ideas from pattern recognition and machine learning have been used to discover program and user behavior. Discovered system features have been used to compute inductively learned classifiers that can identify anomalies and known intrusions.

R. W. Lo, K. N. Levitt, and R. A. Olsson. "MCF: A Malicious Code Filter," Computers and Security, 14(6): 541-566, 1995.
This paper discusses a programmable static Analysis tool called "Malicious Code Filter, MCF, to detect malicious code and security related vulnerabilities in system programs. The MCF uses *telltale* signs to determine whether a program is malicious without requiring a programmer to provide a formal specification. Program slicing techniques are used to reason about *telltale* malicious properties. By combining the *telltale* sign approach with program slicing, a small subset of a large program can be examined for malicious behavior. The paper also discusses how the approach can be defeated and then discusses a countermeasure.

John P. McDermott and William S. Choi, "Taxonomy of Computer Program Security Flaws," ACM Computing Surveys, 26(3): 211-254, 1994.
This paper defines security flaws as "any conditions or circumstances that can result in denial of service, unauthorized disclosure, unauthorized destruction of data, or unauthorized modification of data." The taxonomy defined in this paper organizes information about flaws so that as new flaws are added users will gain a fuller understanding of which parts of systems and which parts of the system life cycle are generating more security flaws than others. The methodology is similar to the one developed by Research in Secured Operating Systems (RISOS) project and Protection Analysis project conducted by Information Sciences Institute of the University of Southern California, both of whom attempted to characterize operating system security flaws.

John F. Morar, David M Chess, "Can Cryptography Prevent Computer Viruses?" Virus Bulletin Conference, September 2000

The relationship between cryptography and virus prevention is complex. Solutions to the virus prevention problem involving cryptography have been proposed, though these solutions do not contribute much to the prevention techniques prevalent at present. This paper discusses the role of encryption in the field of virus authoring and in the field of Anti-Virus research.

Maria M. Pozzo and Terence E. Gray, "An Approach to Containing Computer Viruses," Computers and Security, Volume 6 (1987), No. 4, pp. 321-331.
This paper presents a mechanism for containing the spread of computer viruses by detecting at run-time whether or not an executable has been modified since its installation. The detection strategy uses encryption and is held to be better for virus containment than conventional computer security mechanisms, which are based on the incorrect assumption that preventing modification of executables by unauthorized users is sufficient. Although this detection mechanism is most effective when all the executable on a system are encrypted, a scheme is presented that shows the usefulness of the encryption approach when this is not the case.

J. Reynolds, "The Helminthiasis of the Internet," RFC1135, Information Systems Institute, University of Southern California, Dec 1989
This RFC summarizes the infection and cure of the Internet Worm. It discusses the impact of the worm on the Internet community, ethics statements, role of the news media, rime in the computer world, and future prevention of such incidents. This RFC also reviews four publications that describe in detail, the computer program (a.k.a. Internet Worm or Internet Virus) that infected the Internet in the evening of November 2, 1988

Fred Schneider. "Enforceable Security Policies. Cornell University,"
http://cstr.cs.cornell.edu:80/Dienst/UI/1.0/Display/ncstrl.cornell/TR99-1759
A precise characterization is given for the class of security policies enforceable with mechanisms that work by monitoring system execution. Security automata are introduced for specifying exactly the class of security policies discussed. Techniques to enforce security policies specified by such automata are also discussed.

Mathew G. Schultz, Eleazar Eskin, Erez Zadok and Salvatore J. Stolfo, "Data Mining Methods for Detection of New Malicious Executables," Computer Science Department, Columbia University, New York, USA.
This paper presents a framework for detection of malicious executables with viral characteristics. The motivation for this work is that signatures for new viruses are not known and hence the data mining technique presented in this work will be able to solve this problem in a better way than the current signature-based methods of virus detection. The paper compares results of traditional signature based methods with the other learning algorithms. The Multi-Naïve Bayes method had the highest accuracy and detection rate over unknown programs and had double the detection rates of signature-based methods.

John F. Shoch and Jon A. Hupp, "The "Worm" Programs-Early Experience with a Distributed Computation," CACM, 25(3): 172-180, 1982
This is an exploratory paper for its time. This paper discusses issues found in the early exploration of distributed computing. Authors talk about the motivations and definitions for a worm program from the distributed computation perspective. Not much work had been done in building distributed systems in 1982. The authors wanted to obtain real experience (similar to Arpanet routing and Grapevine). The worm is a computation that lives on one or more machines. The piece on an individual computer is a segment. The segments maintain communications, so that if one fails another can be started in its place on another machine.
They also talk about the protocols and problems in controlling the growth of worm programs. Multi-casting is used to maintain communication. If a host is not heard for a period of time it is assumed dead and removed from the worm. A specified segment is given the responsibility for finding a new idle machine. The biggest problem is controlling growth while maintaining stable behavior. A few applications of the worm programs are also discussed.

Eugene H. Spafford, "The Internet Worm Program: An Analysis," ACM Computer communication review, 19(1), pp. 17-57, Jan 1989
This paper is an analytical commentary on the Internet Worm program, which infected the Internet on the evening of November 2nd1988. The paper defines Worms and Viruses. It discusses the flaws in computer systems that were exploited by the Worm to spread across the

Internet. Patches to these flaws are also discussed. A high level description of the functioning of the Worm program is also provided. The paper then carries a detailed analysis of the Worm.

Eugene H. Spafford, "Computer Viruses as Artificial Life," Department of Computer Sciences, Purdue University, West Lafayette, IN 47907-1398, COAST TR 94-02, 1994
This paper talks about how computer viruses operate, their history, and the various ways computer viruses are structured. It then examines how viruses meet properties associated with life as defined by some researchers in the area of artificial life and self-organizing systems. The paper concludes with some comments directed towards the definition of artificially "alive" systems and related experiments.

Gerald Tesauro, Jeffrey O. Kephart, Gregory B. Sorkin, "Neural Network for Computer Virus Recognition," IEEE Expert, vol. 11, no. 4, pp. 5-6, Aug 1996
This paper describes a neural network for generic detection of boot sector viruses that infect the boot sector of a floppy disk or a hard drive.

K Thompson, "Reflections on Trusting Trust," Comm. ACM 27(8), 761-763 (August 1984)
This ACM classic highlights the issues of trusting a third party. Thompson explains how a *backdoor* can be inserted in a C compiler, which in turn will insert a *backdoor* in the Unix "login" program. The *backdoor* may give unauthorized access into a system.

David Wagner, Drew Dean, "Intrusion Detection via Static Analysis," IEEE Symposium on Security and Privacy, May 2001
This paper describes static analysis methods for host based intrusion detection using program specification for its internal behavior. The specification is automatically derived for the program under analysis. It involves a combination of dynamic monitoring and static analysis to reduce false alarms during detection.

Ian Whalley, Bill Arnold, David Chess, John Morar, Alla Segal, Nortan Swimmer, "An Environment for controlled Worm Replication and Analysis or: Internet-inna-Box," Virus Bulletin Conference, September 2000
The paper outlines a functional prototype of a worm replication system. Techniques and mechanisms for constructing and utilizing an environment enabling the automatic examination of worms and network bases viruses have been described. The paper involves a very brief description of some well-known worms from the past and the present. It elaborates on techniques used by worms to spread across networks. Finally an anatomy of the worm replicator system is presented.

Steve R. White, "Open Problems in Computer Virus Research," Virus Bulletin Conference, Munich, Germany, October 1998
This paper identifies some challenging open issues on computer virus detection and protection. It lists out five problems in this field, namely, Development of New Heuristics for virus detection, the study of viral spread and epidemiology, deploying distributed digital immune system for detecting new viruses, detection of worm programs and proactive approaches towards detection of virus programs.

Tarkan Yetiser, "Polymorphic Viruses, Implementation, Detection and Protection,"
VDS Advanced Research Group, P.O. Box 9393, Baltimore, MD 21228, USA.
http://www.bocklabs.wisc.edu/~janda/polymorph.html
Discusses Polymorphic viruses and engines. It looks at general characteristics of polymorphism as currently implemented.