Computer Virus Response Using Autonomous Agent Technology

Christine M. Trently

Mitretek Systems
7525 Colshire Drive
McLean, VA 22102

voice: (703)-610-1677 fax: (703)-610-1699
email: *ctrently@mitretek.org*

Abstract

Automating the computer virus response offers the ability to prevent and recover from computer virus incidents with minimal input from and impact on the user. This paper proposes an automated computer virus response capability using autonomous agent technology. Although autonomous agent technology has not been exploited in the anti-virus industry, its use in virus response can permit computer system environments to mimic the biological immune system by identifying viruses, removing viruses, and reporting virus incidents. This paper describes the potential use of autonomous agent technology for automating computer virus response, describes the functionality to be realized through the automated response, and then discusses the issues to be addressed for any automated system for handling computer virus response in an enterprise environment. Future directions and considerations for this research are also included.

KEYWORDS: Autonomous Agent; Computer Virus; Automated Response; Immune System

Introduction

During the past decade, the computer virus problem has reached worldwide recognition and prevalence. The 1995 Datapro Information Services Survey of Computer Security Issues showed that 32% of the respondents were extremely concerned with computer viruses and malicious code [2]. There are thousands of DOS viruses and the number is growing at an average of 3 new viruses per day [16]. However, only about 10% of the existing DOS viruses [8] have been seen in actual computer virus incidents or “in the wild” (ITW).

When reviewing the vast amounts of information available on the nature of computer viruses and the various anti-virus software products available, it became evident that computer viruses will be not going away in the near future [4]. In the 1996 Computer Virus Prevalence Survey compiled by the National Computer Security Association (NCSA), the number of virus exposures rose approximately ten-fold in the last year from one virus exposure for every thousand personal computers (PCs) per month to ten virus exposures for every thousand PCs per month [10]. The current mechanisms for detecting and recovering from the growing number of computer viruses are time consuming and require extensive awareness and training for the user community. It is no longer practical, particularly as the connectivity and interoperability advancements increase, to expect the average user to be extensively computer literate.

One manner in which to view the computer virus problem is to continue the comparison to its biological counterpart. The generation of an immune system for computers [7] can be further expanded to include the duplication of the biological equivalent of white blood cells or antibodies to combat “infections” as the computer or network is exposed to known virus strains. The antibodies in the biological immune system combat those entities that are foreign to the system, and the antibodies are not dependent upon one central source for knowing what to combat and how. This gives the antibodies the ability to be distributed and active throughout the body. Without the ability to be distributed and autonomous, the antibodies would be highly susceptible to attack because one entity that could disable one antibody would be able to disable any or all of them [3]. With the use of autonomous agents, the biological function of antibodies or an immune system can be realized in the automated environment.

Needing to More Fully Automate the Computer Virus Response

Since there are approximately 7000 viruses in existence worldwide [16], fully automating the computer virus response to such a large number of viruses is unrealistic and unnecessary. As noted above, only about 10% of the viruses in existence have actually be reported “in the wild.” These are the viruses that can and should be handled in an automated fashion [8].

When looking at the effects of computer virus infections on an organization or enterprise, it is important to note that the costs associated with computer virus infections are growing as connectivity and interoperability increase and computer usage becomes more prevalent. These costs, which can be quite extensive in certain circumstances [10, 12], include the training of computer users in computer virus awareness and anti-virus product usage, the support of technical experts during a computer virus incident, and the interruption to productivity during an incident. In a 12 month period, 63% of the interruptions to processing in the microcomputer environment were attributed to computer viruses and malicious code [2].

The computer virus response within an enterprise includes:

- detecting and identifying the virus,
- collecting a sample of the virus (when possible),
- removing the virus,
- reporting the incident to an administrator or technical support, and
- keeping incident statistics.

These functions are currently performed by the user and require the user to be trained in the use of anti-virus products. Fully automating the response for ITW viruses [8] would seem to provide a considerable cost saving by eliminating the need for extensive training for the user and by reducing or eliminating the user productivity interruptions. An automated virus response could perform the detection, removal and reporting functions without interrupting or alarming the user [8]. Instead of notifying the user, an administrator is notified and the administrator can determine the extent of the incident as well as the need to inform the user. Automating the response, however, should not and does not abolish the need for general computer virus awareness information to be provided to any person using a computer.

A fully automated response, however, cannot be used in all computer virus incidents. The automated response, should, at least, detect and report all viruses, whether ITW, known or unknown. For those incidents dealing with previously unknown viruses, expert technical assistance will still be necessary.

Describing Autonomous Agent Technology

The term “agent” has been used and defined in a variety of ways. One such definition describes agents as “good viruses” [13] since the agent program acts in the background on behalf of the user and, in some instances, has the ability to replicate. Agents have also been compared to artificial life [9]. For this paper, however, autonomous agents are defined as a group of computer programs which utilize artificial intelligence techniques to fulfill a set of goals or tasks in a complex, dynamic environment [1]. Autonomous agent technology uses software designed to adapt its behavior based upon experience and from interactions with other agents in the environment. Each agent is designed to perform a simple, singular task. The collection of agents within an environment, however, can perform sophisticated, intelligent actions. In addition, the collection of agents can migrate throughout the computing environment performing tasks without any interference from or interaction with the user. The computing environment may be a single workstation or an entire network.

Agent Operating Environment

The operating environment for the autonomous agents needs to provide a mechanism for communication between the agents [5]. The agent operating environment can use the application programming interface (API) to pass information or parameters between the agents. In addition, the components of the agent operating environment need to be bound to various operating system functions [5]. These functions include such things as memory management, file management, and internal timing. The components of the agent operating environment also need to be bound to the available message transport service via the communications infrastructure to deploy and receive autonomous agents and their results. Once the components of the agent operating environment are established and bound to the communications infrastructure, the agents can perform their duties independently but have the results of their activities coordinated and managed.

Agent Coordination Engine

Since autonomous agents perform small, individual tasks, there is a need to coordinate the efforts performed and the results obtained by the agents [5, 6]. A centralized coordination engine running in the agent operating environment can provide the ability to coordinate and manage the flow and use of autonomous agents within a given system. The basic functions of an agent coordination engine (ACE) are depicted in Figure 1. The engine includes the ability to launch, authenticate, repair, and communicate with agents throughout the system. The functions of the ACE provide the autonomous agents with the ability to migrate throughout the computing environment to perform their tasks and report their results.

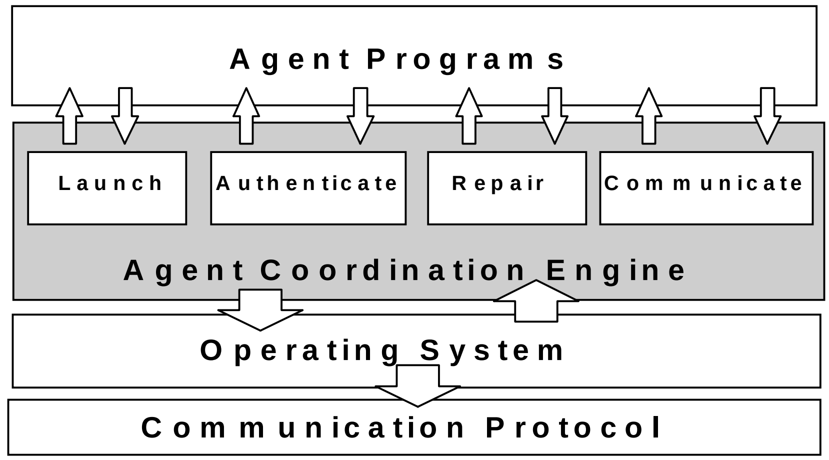

Figure 1 Centralized Agent Coordination

Launching Agents:
The coordination engine has the ability to launch or release agents into the computing environment. The engine will determine which, how many, and when agents are released into the environment. When the agent is launched or released, it is the responsibility of the ACE to ensure that the agent is informed of its scope and boundaries. The engine also verifies that the agents do not exceed their designated limitations.

Authenticating Agents:
In order to assure that the agents are performing the tasks they were designed and intended to perform, the coordination engine must ensure and verify the integrity of the agents used in the computing environment. Authenticating the agents consists of checking the state of the current agent with a known version. This can be accomplished through the use of such things as encryption, hashing or checksums.

Repairing Agents: In conjunction with the integrity of the autonomous agents ensured through authentication, the need to repair or disable damaged agents is necessary. If an agent is found to be damaged (corrupted), the coordination engine removes the damaged agent from service and repairs or replaces it. The repair process consists of replacing the damaged agent with an authenticated version of the agent available to the engine. In extreme cases the engine can notify the administrator that the agent needs to be reloaded from the original software.

Communication Agents: Since the autonomous agents independently perform their tasks, the coordination engine must provide a mechanism to coordinate the use and results of the agent's tasks. The results of the tasks need to be compiled to determine any further action that may be required, such as the release of additional agents.

With the agent operating environment established, the ACE acts to control the flow and use of autonomous agents within a given system. Acting in this manner, the agent operating environment and ACE closely resembles a biological immune system for computer virus response. In conjunction with the "biologically inspired immune system" [7], the use of autonomous agents suggests a more mobile and robust simulation of the immune system. With each agent performing a separate task, it can be suggested that the agents, in fact, act as biologically inspired "antibodies" for the computer system.

Using Autonomous Agents for Automated Virus Response

In a simplified description of the biological immune system, the antibodies detect entities which are foreign to it. Once a foreign body is detected and identified, it is destroyed by one or more antibodies. Acting as antibodies for a computer, autonomous agents need to perform similar functions for computer virus response. These functions, if initially performed from a known clean environment, can proactively prevent a virus infection at its source. This greatly reduces the risk of mass infections or epidemics which are currently experienced in many corporate environments [10]. As noted previously, these functions include the duties shown in Figure 2. Each portion of the automated response is described as part of the agent functions.

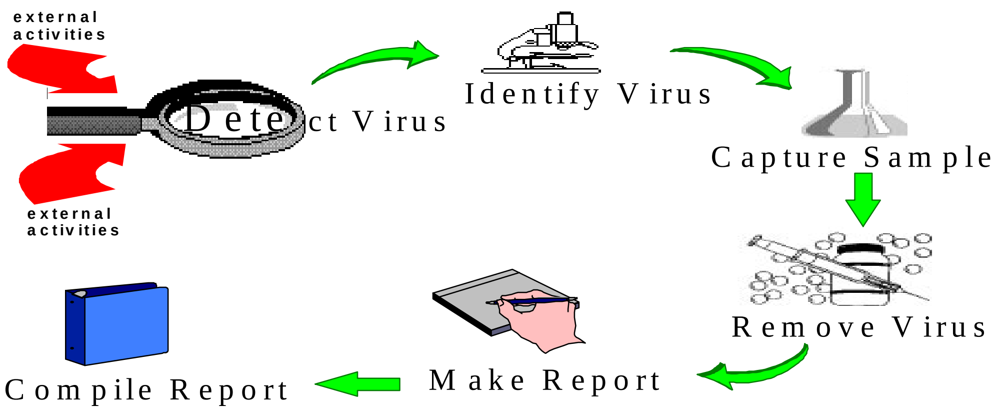

Figure 2 Automated Virus Response Duties

Detecting Viruses

To accomplish the detection of viruses, several autonomous agents are advisable to maintain the singular and simple task structure. Viruses come in three main forms: boot sector, file infector, and multi-partite. At a minimum, the automated response should include a separate agent for each type. Having separate agents for each type of virus allows the detection agents to continuously monitor different areas of the operating environment and to maintain the simple and singular tasks. In addition, each agent needs to be focused on a particular activity and can use different virus detection techniques. The current techniques for virus detection include scanning for known viruses using virus signatures, checking file integrity, and monitoring for suspicious behavior. The crucial activities for virus detection to monitor include:

- Inserting diskettes
- Receiving Mail
- Copying/Moving Files
- Creating/Saving Files

- Executing Files
- Opening Files

Once a virus is detected, the agent notifies the ACE along with the name/location of the suspected virus.

Identifying Viruses

Once a virus is detected (or suspected) using one of the virus detection techniques, agents must exist to positively identify the virus, if possible. In some cases, the detection agent may have a tentative identification; however, some of the detection techniques only detect a change, not the cause of the change. Again, to keep the agent task singular and simple, the identification of the virus is described separately from the detection. The identification of the virus is imperative to ensure proper recovery techniques are used. Since the focus of the automated response is on ITW viruses, the virus can be identified through either known virus signatures or known behaviors. Again, separate autonomous agents are advisable to identify boot sector, file infector and multi-partite viruses. The duties of the identification agents also need to be separate for each of the detection techniques used. There should be agents that handle viruses detected by known virus signatures, viruses detected by integrity checking and viruses detected by suspicious behavior. Once the virus is identified, the identity is returned to the ACE for appropriate recovery techniques. In addition, the identification agents are equipped to notify the ACE when the detected virus cannot be identified and, again, the ACE initiates the appropriate action(s).

Capturing Samples

Once the virus is detected and potentially identified, the ACE launches the appropriate agent(s) to collect a sample of the virus. Each capturing agent is supplied with the name/location of the infected item. Again, there is a separate agent to handle capturing boot sector, file infector and multi-partite viruses, since the tasks associated with each sample are different. To capture a sample, the agent makes a copy of the infected item and places it in a designated, protected location. A pointer to that location is sent to the ACE and the appropriate recovery agent is launched. For an unknown virus, the capturing agent activity is the same; however, the response from the ACE does not include a removal process, rather, it initiates the reporting agent(s).

Removing Viruses

After the sample is taken for ITW viruses, the ACE launches the appropriate agent for removing the virus. The information provided to the agent includes the name/location of the infected item and the identity of the virus. The recovery agent then determines the appropriate recovery technique for the identified virus and performs the necessary actions. Once completed, the recovery agent determines if the removal was successful and notifies the ACE of the removal status. If it was not successful, the agent notifies the ACE for appropriate reporting to the administrator.

Reporting Incidents

Once a virus is removed or, at least, the sample is taken (in the case of an unknown virus or unsuccessful removal), the ACE launches the reporting agent. The reporting agent generates a report of the incident including the date of the incident, the type of virus, the name of the virus detected and identified (if known), the location of the infection, and the success of the removal process, and other relevant information determined throughout the response. The agent then sends the report and the sample retrieved from the designated location to the administrator. The agent also sends the report and location of the sample to the repository site for future report compilation. Once the report is sent to the administrator and integrated into the repository, the reporting agent returns a completion notice to the ACE.

Compiling Reports

After reports are received from the reporting agent(s), they are stored in a repository site. The compiling agent(s) are launched to compile and generate reports. The agent may generate statistics based upon learned preferences [9] of the administrator. The compiled reports act as summaries of virus incidents and can be based upon specific intervals (i.e. monthly), virus type, virus name, or total incidents.

Future Considerations

There are many advantages for using autonomous agent technology, such as the ability of the agents to be easily tailored and trained, the efficiency, extensibility, scalability and graceful degradation of the agents, and the overall system's resilience to subversion [1]. While the advantages are numerous, there are also other considerations which will influence the use of autonomous agent technology for automated virus response. These considerations include: reducing processing overhead for the system, preventing deliberate or unintentional misuse, maintaining the integrity of agents, identifying the appropriate viruses to be included in an automated response, and providing accurate and consistent virus identification and recovery information. These considerations will impact the future directions taken for research in this area.

Reducing the Processing Overhead

While the agents themselves can be optimized to have minimal impact on system processing, the total automated virus response can impose an overhead on the computing system. The automated response will consume both memory and central processing time detecting and recovering from virus incidents. The use of memory and processing time will need to be minimized as much as possible to ensure that the benefits for automating the virus response are practical and can be realized. If the overhead imposed by an automated response degrades the overall performance of the system, the user community will disable or not install the product. The goal is not to decrease productivity but to enhance it.

Preventing the Misuse of Agents

Since agents can be defined as "good viruses" and have the ability to be executed throughout a system without user interaction or notification, it is imperative to ensure that the agent cannot be used for deliberate or unintentional misuse. Mechanisms will be needed to control the functions available to the agents and the scope or extent to which an agent can travel or perform its tasks. For instance, if an automated response is developed for a networked or client/server environment, the agents must be prevented from exceeding the boundaries of that environment. In addition, the system functions available to agents must be limited to those which do not allow the modification of other programs [5]. This can prevent an agent from being used to propagate viruses throughout the system or from changing programs to include Trojan horses.

Maintaining the Agent Integrity

As with the prevention of misuse, the integrity of the agents must also be ensured. Agents can be corrupted through deliberate or unintentional means. The results from executing a corrupted agent whether by design or accident can have disastrous results, such as system failure and data loss. It is possible to protect the integrity of the agents and the coordination engine with various forms of authentication or encryption. A possible method to protect the agent operating environment is to provide for integrity controls, such as authentication, through the design and implementation of a security architecture [11]. The mechanisms needed to maintain the integrity of the agents and their environment requires careful consideration to prevent a single agent or system of agents from causing harm.

Identifying the Target Response

Given that a small percentage of the viruses that exist are seen in actual incidents or in the wild, the automation of the virus response needs to focus its efforts on the detection and removal of the ITW viruses. To ensure that the automated response addresses the ITW viruses, a consistent designation of those viruses must be maintained and used. The *Wildlist* [14], maintained by Joe Wells of the IBM's T. J. Watson Research Center, provides a list of the viruses reported in actual virus incidents throughout the world. This list is currently being used by NCSA to test and certify anti-virus products [4]. The difficulties with the *Wildlist* are that the viruses noted as being in the wild currently contain naming variations and not all viruses actually in the wild are identified. Work is being done to address these issues [15]. Once the *Wildlist* and virus naming conventions are standardized, the targets of an automated response can be more clearly delineated.

Providing the Identification Information and Recovery Techniques

To minimize the impact of any virus response, it is important to have timely and accurate information on the identification and recovery of the ITW viruses. Accurate identification of viruses is important, since it directly affects the recovery process. It is the identification of the virus that determines the type and extent of the automated recovery process used. It is also imperative that the recovery techniques used for the ITW viruses are accurate and successful. Without successful recovery, an automated response loses its effectiveness and actually impedes productivity and fosters a false sense of protection. The fewer times that an administrator is involved with the recovery process, the fewer interruptions will be experienced by the user. Again, as in the identification of the virus, the recovery response needs to be standardized and robust enough to handle the ITW viruses consistently and effectively. It is possible that the agents could be trained [1,9] to determine the most appropriate recovery process if there are multiple infections present at the same time. In addition, false alarms are costly. In one case study, the cost of a small incident involving one virus and nine computers exceeded $23,000 in labor charges for lost time and productivity [12]. In actuality, the costs experienced in this case study were not significantly different than the costs that would have been experienced had the incident been real.

Summary

It is evident that the issue of computer viruses will be not going away in the near future. The current mechanisms for detecting and recovering from the growing number and complexity of computer viruses are no longer practical, timely, or efficient in regard to user productivity. The costs of training users and lost productivity due to virus incidents continue to rise as the complexity of both the operating environments and computer viruses increase.

Fully automating the response for the prevalent set of viruses would provide a considerable cost savings by eliminating the need for extensive training on the use of anti-virus products for the user and by reducing or eliminating user productivity interruptions. The generation of an immune system for computers using autonomous agent technology to combat virus infections can provide the automated response for computer viruses. Such an immune system can prevent the infection at its source by detecting a virus before it infects the computer or network. While the use of an automated response can be realized for known viruses with known recovery techniques, it should be noted that a fully automated response cannot be used in all computer virus incidents. For those incidents dealing with previously unknown viruses, expert technical assistance will still be needed.

The value of combining autonomous agent technology and automated virus response as suggested in this paper will be determined by the successful implementation of a prototype and operational use of the resulting automated virus response system. While researching and developing this prototype, the lessons learned throughout will be noted and used in determining other considerations, future directions and later versions.

The potential harm caused by making autonomous agent technology available for automated virus response provides a point to ponder. Are we providing the virus writers with a streamlined vehicle for virus propagation? As with most innovative concepts, autonomous agent technology can be used for both good and “evil”. Arguably, autonomous agent technology can be readily seen as a threat, particularly in the virus arena. The challenge is to harness this advantageous but volatile technology to protect the computing environment from its most prevalent enemy, the computer virus [2].

Acknowledgments

I wish to thank the numerous people who provided me with support and encouragement to pursue this effort as well as providing useful suggestions and critical comments during the preparation of this paper. The initial idea for an autonomous, mobile immune system for computers grew out of a casual discussion with Bob Williamson, formerly of The MITRE Corporation, after a virus incident. I also wish to thank Mike Lambert, Frontier Corporation, whose encouragement and frank discussions about the methods and reasons for fully automating virus response brought this idea and resulting paper to fruition.

References

[1] Crosbie, Mark and Eugene H. Spafford, “Defending a Computer System using Autonomous Agents,” *Making Security Real - 18th National Information Systems Security Conference Proceedings*, Baltimore, MD, October 1995, pp. 549-558.

[2] Datapro Information Service Group, “Computer Security Issues: 1995 Survey,” McGraw-Hill, Incorporated, Delran, NJ, October 1995.

[3] D'haeseleer, Patrik, Stephanie Forrest, and Paul Helman, "An Immunological Approach to Change Detection: Algorithms, Analysis, and Implications," *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, Oakland, CA, May 1996, pp. 110-119.

[4] Ford, Richard, "Why Viruses *Are* and *Always* will be a Problem," *NCSA News*, April 1996, pp. 5-7.

[5] Harrison, Colin G., David M. Chess, and Aaron Kershenbaum, "Mobile Agents: Are They a Good Idea?" IBM T. J. Watson Research Center, Yorktown Heights, NY, March 28, 1995, http://www.research.ibm.com/massive/mobag/ps.ps

[6] Heilmann, Kathryn, Dan Kihanya, Alastair Light, Paul Musembwa, "Intelligent Agents: A Technology and Business Application Analysis," November 1995. URL: http://haas.berkeley.edu/~heilmann/agents/

[7] Kephart, Jeffrey O., "A Biologically Inspired Immune System for Computers," High Integrity Laboratory, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1994, gopher://index.almaden.ibm.com:70/0VIRUS/PAPERS/ALIFE.PS

[8] Lambert, Michael, "Fully Automated Response for In The Wild Viruses (FAR-ITW)", Rochester, NY, July 1995.

[9] Maes, Pattie, "Intelligent Software," *Scientific American*, volume 273, number 3, September 1995, pp. 84-86.

[10] National Computer Security Association (NCSA), "1996 Computer Virus Prevalence Survey," NCSA, Carlisle, PA, April 1996.

[11] Thirunavukkarasu, Chelliah, Tim Finin, and James Mayfield, "Secret Agents - A Security Architecture for the KQML Agent Communication Language," University of Maryland Baltimore County, Baltimore, MD, 1995, http://www.cs.umbc.edu/kqml/papers/secret.ps

[12] Trently, Christine M., "False Alarms: A Case Study" *InfoSecurity News*, volume 7, number 2, March/April 1996, p. 47.

[13] Wayner, Peter, *Agents Unleashed, A Public Domain Look at Agent Technology*, AP Professional, Chestnut Hill, MA, 1995.

[14] Wells, Joe, "PC Viruses in the Wild - February 10, 1996 (The *Wildlist*)," 1996.

[15] Wells, Joe, "Reality Check: Stalking the Wild Virus," NCSA's International Virus Prevention Conference Proceedings, April 1996, pp. Q1-Q13.

[16] White, Steve R., Jeffrey O. Kephart, and David M. Chess, "Computer Viruses: A Global Perspective," *Virus Bulletin Conference Proceedings*, Oxfordshire, England, September 1995, pp. 165-182.