Creating a Secure Computer Virus Laboratory

John Aycock and Ken Barker
Department of Computer Science
University of Calgary

About the Authors

John Aycock is an Assistant Professor at the University of Calgary in the Department of Computer Science. He received a B.Sc. from the University of Calgary in 1993, and an M.Sc. and Ph.D. from the University of Victoria in 1998 and 2001, respectively. His research interests include compilers and programming language implementation, compiler tools, system software, operating systems, and all things low-level. He conceived and taught the University's infamous "Computer Viruses and Malware" course.

Ken Barker is a Professor at the University of Calgary and Head of the Department of Computer Science. He holds a B.Sc. and M.Sc. from the University of Calgary and a Ph.D. (1990) from the University of Alberta. His primary research area is in database systems but has published in several other areas including security, bioinformatics, distributed systems, and networks. He is also the Director of the Advanced Database Systems and Applications Laboratory in Calgary which consists of about 30 active researchers.

Mailing Address: Department of Computer Science, University of Calgary, 2500 University Drive N.W., Calgary, Alberta, Canada T2N 1N4; Phone: +1 403 210 9409; E-mail: aycock@cpsc.ucalgary.ca.

Reference to, or **Citation** of this paper should be made as follows:
Aycock, J. & Barker, K. (2004). Creating a Secure Computer Virus Laboratory (Case Study). In U.E. Gattiker (Ed.), EICAR 2004 Conference CD-rom: Best Paper Proceedings (ISBN: 87-987271-6-8) 13 pages. Copenhagen: EICAR e.V.

CASE STUDY

Creating a Secure Computer Virus Laboratory

Abstract

A secure environment in which to work with computer viruses and other malware is not created through technical means alone, yet published virus laboratory specifications very seldom venture into the nontechnical. Our secure computer virus laboratory has five categories of safeguards: legal, ethical, social, behavioral, and technical. These safeguards have been subject to an external, independent security review, and parts have been reviewed by anti-virus researchers. Commissioning, operation, and decommissioning of the virus laboratory is also discussed. Our goal is to establish a computer security standard for secure laboratories analogous to that of biohazard laboratories.

Descriptors

computer viruses, computer security, malware analysis, anti-virus research, virus laboratories, secure laboratories, computer security education

Introduction

It is not news to anyone in the anti-virus community that the University of Calgary is offering a course on computer viruses and malware. This course was controversial because of the teaching method we used, where students would – among other things – learn about viruses by creating one themselves. The issue of teaching methods is outside the scope of this paper, however.

Even our staunchest critics would agree that a secure environment is necessary to safely work with viruses and other forms of malware. But how is a secure environment established? Current published work in this area details the technical aspects of laboratory setup and configuration, but this is only one piece of the puzzle. Any technical safeguards can be circumvented by people, either external attackers or insiders. Consider, for example, a person who is supposed to work with computer viruses in a secure laboratory, but instead works outside that secure environment. No amount of technical wizardry in the laboratory can prevent this.

There is an entire class of social engineering attacks which exploit weaknesses in human nature (Granger, 2001, 2002; Mitnick & Simon, 2002), further proof of the limitations of technical safeguards. We are forced to conclude that any computer security solution which fails to address the human element is doomed to fail. Similarly:

ISBN: 87-987271-6-8

> Any "secure" computer laboratory which fails to address the human element cannot be considered secure.

This is the key distinguishing feature of our computer virus laboratory. We do not achieve security through a narrow focus on technical matters, but by employing a much broader range of safeguards, which we group into five categories:

1. Legal
2. Ethical
3. Social
4. Behavioral
5. Technical

These safeguards have been subject to an external, independent security review and risk assessment. In the remainder of this paper we discuss our safeguards in detail, followed by our procedures for laboratory commissioning, operation, and decommissioning. A discussion of desired future enhancements, related work, and our conclusions complete the paper. As our work has been in a university environment, we assume throughout that the users of the virus laboratory are graduate and senior undergraduate students.

Disclaimer

The information in this paper details our experience and the procedures we followed in creating a virus laboratory suitable for graduate and undergraduate instruction. Obviously we cannot offer any guarantees of security, even if our methods are reproduced, as each environment is different and our methods would need to be adapted appropriately.

Legal Safeguards

Our legal safeguards are twofold:

1. Teaching students the legal repercussions of malware creation and release, and
2. Imposing contractual legal obligations on the students.

Before students had access to the virus lab, we spent roughly the first 10% of the course on malware-related legal topics. This included discussions about relevant parts of Canadian and American law, the Council of Europe Convention on Cybercrime (Council of Europe, 2001), and the enforcement of laws across international boundaries. In addition, a practicing lawyer was brought in for a guest lecture to provide authoritative, practical legal information.

Contractual obligations were established by a legal agreement drawn up by the University's lawyers. Students were required to sign this agreement to take the virus course and have access

ISBN: 87-987271-6-8

to the virus laboratory. It should be noted that students were apprised of this requirement prior to the start of the course, were given ample time to review the agreement (with their own legal counsel if desired), and had the opportunity to transfer to any other Computer Science course without prejudice if they chose not to sign.

The legal agreement includes, among other things, adherence to the laboratory protocol we established (described below), and proper usage and handling of course material. With respect to the virus laboratory, "course material" not only covers code and information placed in the lab by the instructor, but also code the students develop and printouts they make in the laboratory. Furthermore, *de rigueur* for a litigious society, the agreement spells out liability and indemnity.

Ethical Safeguards

Following legal topics, and also before students had access to the virus lab, we spent approximately another 10% of the course on ethics. Compared to some other disciplines, many computer science programs do not spend a lot of time on ethics. We therefore went through a lot of foundational material: general ethical theories, moral development, and ethical decision-making processes. More specialized areas were then considered, such as codes of ethics for computer professionals and the AVIEN and EICAR codes of conduct (AVIEN, 2001; EICAR, 2003). As we did for legal topics, an expert guest speaker was brought in, who talked about professional ethics.

In a university environment, discussing ethics in relation to malware is a balancing act. Our job at a university is to remain objective, so students cannot be indoctrinated with specific values; we can, however, impress upon students the expectations that society has of computer professionals, particularly those handling dangerous things such as malware.

We "test ethics," insofar as is possible, by working through examples in lectures and with a written ethics assignment.

Social Safeguards

Socially, we exploit peer pressure to add an additional set of safeguards. Students work on programming assignments in the laboratory in groups of two, and the lab machines are configured in such a way that a machine cannot be used unless *all* group members in a particular group have logged in. We stress to the students, and entrench in the laboratory protocol, that they are jointly responsible for actions taken in the laboratory with these accounts.

This arrangement creates social pressure to conform to safety guidelines, but has two other side effects. First, it adds an extra level of security in the laboratory, because any illegal or unethical action would require collusion between users. Second, it guards against inadvertent bypassing of safeguards, by establishing an environment conducive to pair programming, a technique whose

ISBN: 87-987271-6-8

proponents argue increases software quality (Beck, 2000). This latter point has been observed more generally by social scientists, who noted improvements in product quality in small face-to-face groups compared to individual efforts (Deutsch, 1949; Hammond & Goldman, 1971).

Behavioral and Technical Safeguards

Behavioral safeguards regulate conduct within the laboratory. This is laid out in a formal laboratory protocol, which is tightly coupled with the technical safeguards. Technical safeguards, in turn, are based on the laboratory specification and physical setup. The laboratory specification and protocol described in this section were vetted by our technical staff and anti-virus researchers, in addition to a separate, independent, external review.

Laboratory Protocol

Our laboratory protocol to regulate behavior in the laboratory was initially based on biohazard protocols (Health Canada, 2001); biologists and chemists have had decades of experience working with dangerous substances, and it is only prudent to build on their experience. Obviously, the analogy breaks down after a certain point, but there were a number of things to be learned about laboratory access, operation, and personnel training.

Since the contagions of concern in the computer virus lab are electronic, we had to add a number of provisions with respect to media handling, and any means of electronic transmission, both wired and wireless. Our initial thought was to let students bring media *into* the lab, so long as it was not brought out again, to allow material researched on the Internet to be brought in, but after negative reviewer feedback we scrapped this idea. Printouts were also contentious, in two ways: first, that we were allowing them to be made at all; second, how they were to be handled by students. We eventually clarified the protocol to specify how printouts should be handled, but still allowed them to be made – at the very least, printouts can be useful for debugging purposes.

The final lab protocol is given in the Appendix. The provision dealing with laboratory entry, where people have to enter individually, is taken from our observations of airport security. Disallowing ‘tailgating’ gives us an electronic card key entry record for each person, in case a later audit is necessary.

The lab protocol includes mention of technical staff and the course instructor, and this is deliberate. While the laboratory was active, even these people were subject to the laboratory protocol. The behavioral safeguards established by the laboratory protocol are meaningless unless *everyone* adheres to them. Indeed, the first author (as the course instructor) religiously removed his cell phone upon entry to the laboratory and was seen by the students to be doing so. This type of action is vital to underscore the seriousness of the protocol.

ISBN: 87-987271-6-8

Specification and Physical Setup

Our virus laboratory is physically located inside two card key access areas. All computer science students and staff have access to the outer area, but the card key lock for the virus laboratory itself is only accessible to authorized users. The laboratory is a brick-walled room with one entrance, whose door is equipped with a door closer and an alarm which rings if the door is open too long; this alarm notifies Campus Security. In case of a power failure, the card key lock on the laboratory defaults to the locked state (exiting the lab is always possible for safety reasons). All network ports in the room are both disabled and physically disconnected.

Two cameras are ceiling-mounted in the room. They are motion-triggered, and send their images to a computer located outside the laboratory; these two video lines are the only cables leaving the room. We used ZoneMinder to manage the camera images, whose fields of view are shown in Figure 1. The goal was to have one camera watching the door, the other the locked cabinet where the laboratory's server and networking equipment was installed. The blind spots in the fields of view were not critical, because the captured images provided a record of who was in the laboratory, and when – sufficient information for an investigation should the need arise.

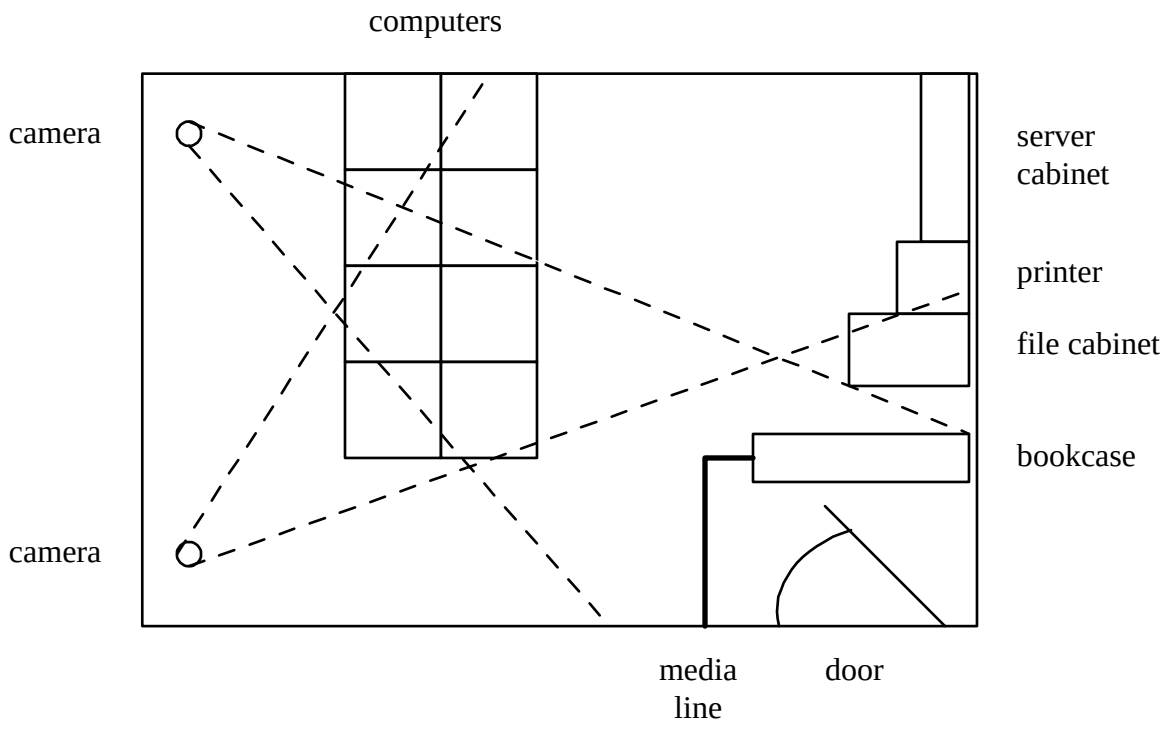

Figure 1. Floor plan of our virus laboratory. Dashed lines indicate the cameras' field of view.

The server we used was a Sun Blade 100 (which has a SPARC architecture) with RAID disks running Solaris. It had a stripped-down OS configuration, basically only running Samba and `rsync` servers. The server was connected to the other lab machines via a network switch, whose ports were locked to the MAC addresses of the lab machines.

The lab machines that students used were x86-based (specifically, 1.4GHz P4 Xeons) with at least 256M RAM and at least a 10G hard drive. All ran Red Hat Linux 9, with VMware Workstation on each, and FreeBSD 4.9 ran inside VMware. The Linux environment was set so that only VMware could be run upon login.

Lab machine I/O was limited to a PS/2 keyboard and mouse, video monitor, network, and a read-only CD-ROM drive. Everything else was disconnected physically if possible, but modern machines have a number of I/O ports directly mounted to the motherboard, which we disabled in the machines' BIOS. Other BIOS settings were locked down, too, including enabling the BIOS setup password and chassis intrusion detection. Physically, chassis intrusion was discouraged by padlocking the lab machines to their associated tables.

To clearly demarcate where outside electronic devices and media could and could not be, we placed a "media line" in the lab by the door, literally a line marked on the floor by brightly-colored tape. An area inside the laboratory, but not crossing the media line, was equipped with a bookcase to give laboratory users a secure place to leave restricted items.

Printouts were made using a networked PostScript printer. Secure disposal of printouts consisted of a locked filing cabinet with a slot cut in the top for paper to be inserted.

Technical Safeguards

From the technical standpoint, we not only had to ensure physical security of the laboratory and its machines, but also electronic security. Clearly, restricting I/O, BIOS settings, and the login environment go a long way towards electronic security. At a higher level, we wanted a virtual barrier to malware escape, a goal met through use of the virtual x86 machine that VMware provides.

We also wanted to avoid a software or hardware monoculture, to reduce the risk of malware propagation in the unlikely event that some bit of malicious code escaped its VMware prison. The degree of heterogeneity in operating systems on the lab machines, and in architectures between the lab machines and the server, is intentional.

Commissioning, Operation, and Decommissioning

Safeguards and specifications are not the only concerns involved in creating a virus laboratory. *Commissioning* the laboratory includes steps taken to verify security and liaise with external

ISBN: 87-987271-6-8

organizations before the lab goes live; *operation* is the set of normal maintenance procedures; *decommissioning* describes how the laboratory is shut down and decontaminated.

Commissioning

After the technical staff had set up the laboratory hardware and software, the lab was checked for both adherence to the original specification as well as ways to misuse the computers, especially misuses that could potentially result in safeguards being bypassed. Kernighan is quoted as saying 'Each new user of a new system uncovers a new class of bugs' (Bentley, 1988, p. 60). Keeping this in mind, the first author[1] – not the technical staff who set the machines up – performed the checking. Computer operation was examined from initial booting through running VMware and logging out, all BIOS settings on all the machines were checked, and two machines were chosen randomly and physically taken apart to inspect the internal connections. The few minor problems noted were corrected by the technical staff and rechecked before the laboratory became operational.

We also had a "no clean" order issued for the laboratory by Campus Infrastructure, to avoid any eager caretaking staff from trying to get into the lab, and met with Campus Security to clarify their role should they be called to the laboratory. Essentially, security personnel would be used to secure the laboratory site, leaving everything in a state suitable for computer forensics (NHTCU, 2003).

Finally, student access to the laboratory's key card lock was enabled, and all appropriate people were notified that the lab protocol was in full effect.

Operation

Laboratory operation was uncomplicated. We stopped in several times a week to inspect the premises, and periodically emptied the filing cabinet of printouts and shredded them. The handful of minor maintenance issues that arose (none of which were security-related) were handled by the technical staff.

Decommissioning

The room our virus laboratory occupies is being used for a different course this upcoming semester, so we were able to experience the full decommissioning process too. Student key card access was removed immediately following the due date for their final programming assignment, and all paper materials were removed from the room.

The technical staff sanitized the lab machines' hard drives to RCMP TSSIT specifications (1997, Appendix OPS-II). The software used for cleaning does not run on the SPARC architecture the

[1]This author has a background in system administration.

ISBN: 87-987271-6-8

laboratory server uses, but the server's drives were connected to one of the lab machines and sanitized there. As an added precaution, the PostScript printer's NVRAM was cleared too.

Camera images taken during laboratory operation were written to DVD and are being retained, along with the students' signed legal agreements, for a reasonable period in the department's safe. As with commissioning, the final step was to notify appropriate people that the laboratory was no longer in use, and that the lab protocol had been lifted.

Future Enhancements

One area of concern for us is that our contingency plan for the escape of malware from the laboratory is limited to computer forensics only. Of course, we would attempt to notify anti-virus companies of any breach, but at present we cannot be assured that we could contact the right people in a timely fashion. Emergency communication channels need to be established from laboratories such as ours to industry. From a safety point of view, it would also be useful to try and measure the effectiveness of the nontechnical safeguards.

Actively preventing use of wireless electronic equipment in the laboratory might be a useful addition. Currently we address the issue by disallowing wireless equipment in the laboratory protocol, securing the machines physically, and disabling noncritical ports on the machines. Jamming the wireless signals would be a more direct method, but the protection it provides would need to be examined.

We anticipate doing some fine-tuning on the laboratory environment based on our experience and student feedback. In particular, we want to allow VMware to boot from a set of system images, so that multiple operating systems may be used for laboratory work, and permit students to customize their working environment in persistent ways. We have also had some technical glitches with both Samba and FreeBSD, and we will be looking at alternatives to them in future instantiations of the virus laboratory.

Related Work

Other laboratory setups have been described in the literature, and fall into two groups: those designed specifically for handling of malware, and those intended for more general computer security instruction. Interestingly, neither group seems to reference the others' work.

Malware-specific setups span quite a range. At the one extreme is Ludwig, who simply gives an admonishment to be careful (1998, p. 20), which is not a very satisfactory solution. Cohen briefly describes the technical specifications and operation of his "Virus Research Network," used to experiment with viruses on networks (1994, pp. 218-221). More recently, Schšldstršm (2002) described a portable laboratory, intended for educational virus demonstrations; Skoudis and Zeltser (2004) give several configurations for a "Malware Analysis Laboratory," one of

ISBN: 87-987271-6-8

which is also portable. None of these malware-specific laboratory setups take nontechnical issues into account.

Work on general computer security laboratories can be further divided. One body of work uses physically isolated networks for experiments (Bishop & Heberlein, 1996; Hill, Carver, Humphries, & Pooch, 2001; Mateti, 2003), while the other relies on nonphysical mechanisms to keep traffic from the laboratory from leaking out and posing a threat (Padman, Memon, Frankl, & Naumovich, 2003). We classify Mayo and Kearns (1999) in this latter group, although they are focused more on systems development than security.

Looking at nontechnical aspects, Bishop and Heberlein (1996) mention "procedural" security mechanisms in passing, which we would equate to our laboratory protocol. Mateti (2003) notes that approximately 20% of their security course is devoted to ethical and legal issues, but this training is a goal of their course rather than an integral part of their laboratory's security. More generally, there have been many calls for computer ethics training (Currie Little et al., 1999; Martin & Holz, 1992; Martin & Weltz, 1999), even as early as childhood (Gordon, 1995).

Conclusion

Only by taking nontechnical, human aspects into account along with technical aspects can we arrive at a truly safe virus laboratory. There are simply too many ways for humans to defeat technical-only safeguards to be ignored. We have addressed this problem by employing five categories of safeguards in our virus laboratory: legal, ethical, social, behavioral, and technical. We hope that other institutions constructing similar laboratories will find our experience useful.

Acknowledgments

Many people helped put together and maintain the lab. We want to thank our technical staff, especially Tim Bliek, Darcy Grant, Shaun Laing, Brian Scowcroft, Jennifer Walker, and Erik Williamson. A number of people reviewed our lab specification and lab protocol to help ensure safety; Randy Abrams, Brad Arlt, Fred Cohen, Sarah Gordon, and Darcy Grant all provided valuable feedback. Thanks also to Gerry Bliss and Sid Tolchinsky from SPIE for their independent review and risk assessment. Stephen Jenuth and Ken Chapman gave the guest lectures on law and ethics, respectively. Shannon Jaeger proofread a draft of this paper. Our apologies to anyone whose contribution we have inadvertently omitted here.

ISBN: 87-987271-6-8

References

AVIEN. (2001). Anti-virus information exchange network code of conduct. Available: http://www.avi-ews.org/codeconduct.html.

Beck, K. (2000). Extreme programming explained. Boston, MA: Addison-Wesley.

Bentley, J. (1988). More programming pearls. Reading, MA: Addison-Wesley.

Bishop, M., & Heberlein, L. T. (1996). An isolated network for research. In Proceedings of the 19th NIST-NCSC national information systems security conference (pp. 349-360).

Cohen, F. B. (1994). A short course on computer viruses (2nd ed.). New York: Wiley.

Council of Europe. (2001). Convention on cybercrime (ETS No. 185). Available: http://conventions.coe.int.

Currie Little, J., Granger, M. J., Boyle, R., Gerhardt-Powals, J., Impagliazzo, J., Janik, C., Kubilus, N. J., Lippert, S. K., McCracken, W. M., Paliwoda, G., & Soja, P. (1999). Integrating professionalism and workplace issues into the computing and information technology curriculum. ITiCSE '99 Working Group Reports, 31(4): 106-120.

Deutsch, M. (1949). An experimental study of the effects of co-operation and competition upon group process. *Human Relations, 2*(3): 199-231.

EICAR. (2003). EICAR code of conduct. Available: http://www.eicar.org/code_of_conduct.htm.

Gordon, S. (1995). Ethical Computing. Available: http://www.commandsoftware.com/virus/ethics.html.

Granger, S. (2001). Social Engineering Fundamentals, Part I: Hacker Tactics. Security Focus. Available: http://www.securityfocus.com/infocus/1527.

Granger, S. (2002). Social Engineering Fundamentals, Part II: Combat Strategies. Security Focus. Available: http://www.securityfocus.com/infocus/1533.

Hammond, L. K., & Goldman, M. (1971). Competition and non-competition and its relationship to individual and group productivity. In B. L. Hinton & H. J. Reitz (Eds.), *Groups and organizations: integrated readings in the analysis of social behavior* (pp. 339-348). Belmont, CA: Wadsworth.

ISBN: 87-987271-6-8

Health Canada. (2001). The laboratory biosafety guidelines (3rd ed., draft). M. Best & M. Heisz, eds. Available: http://www.hc-sc.gc.ca/pphb-dgspsp/ols-bsl/lbg-ldmbl/index.html.

Hill, J. M. D., Carver Jr., C. A., Humphries, J. W., & Pooch, U. W. (2001). Using an isolated network laboratory to teach advanced networks and security. In Proceedings of the 32nd SIGCSE technical symposium on computer science education (pp. 36-40).

Ludwig, M. (1998). The giant black book of computer viruses. Show Low, AZ: American Eagle Publications.

Martin, C. D., & Holz, H. J. (1992). Integrating social impact and ethics issues across the computer science curriculum. In Education and Society: Information Processing 92 (Vol. II, pp. 137-143).

Martin, C. D., & Weltz, E. Y. (1999). From awareness to action: Integrating ethics and social responsibility into the computer science curriculum. ACM SIGCAS Computers and Society, 29(2): 6-14.

Mateti, P. (2003). A laboratory-based course on Internet security. In Proceedings of the 34th SIGCSE technical symposium on computer science education (pp. 252-256).

Mayo, J., & Kearns, P. (1999). A secure unrestricted advanced systems laboratory. In Proceedings of the 30th SIGCSE technical symposium on computer science education (pp. 165-169).

Mitnick, K. D., & Simon, W. L. (2002). The art of deception: Controlling the human element of security. Indianapolis, IN: Wiley.

NHTCU: National Hi-Tech Crime Unit. (2003). Good practice guide for computer based electronic evidence (2nd ed.).

Padman, V., Memon, N., Frankl, P., & Naumovich, G. (2003). Design and implementation of an information security laboratory. Journal of Information Warfare, 2(3).

Royal Canadian Mounted Police. (1997). Technical security standard for information technology (TSSIT).

Schšldstršm, K. (2002). How to use live viruses as an education tool. In Proceedings of the twelfth Virus Bulletin international conference (pp. 251-261).

Skoudis, E., & Zeltser, L. (2004). Malware: Fighting malicious code. Upper Saddle River, NJ: Prentice Hall.

ISBN: 87-987271-6-8

Appendix

CPSC 599.48/601.92: Computer Viruses and Malware

Lab Protocol (Fall 2003)

Working with malicious software carries a great responsibility with it, unlike any other area of computer science. The following lab protocol **MUST** be strictly adhered to to ensure safety.

- Any accidents or lab protocol violations must be reported in writing to the course instructor and department head as soon as possible.
- People must enter the lab individually, using their own card key.
- Lab access is restricted to authorized people only: students in CPSC 599.48/601.92, computer science technical support staff, and the course instructor. If any unauthorized people are in the lab, Campus Security should be notified (220-5333).
- The lab door is equipped with a door closer. The door may not be held open in any way.
- There must be no electronic connection between inside the lab and outside the lab. The wired in-lab network may not be changed or tampered with.
- In the lab, it is forbidden to use any electronic equipment apart from the designated lab equipment. This includes, but is not limited to, wireless network devices, IR, PDAs, digital cameras, laptops, and cell phones.
- Upon entry to the lab, there is a "media line." No electronic devices may cross this line, even if powered off. No passive electronic media, such as floppy disks, CD-ROMs, and USB drives may cross this line either.
- Assignment code may **under no circumstances** be worked on outside the lab or stored electronically outside the lab machines.
- There is a printer in the lab. Any assignment-related printouts must be made with this printer. The printouts should be kept in a secure fashion, i.e., they should not leave the lab. If you discover that a printout has been brought out accidentally, you should store it safely and return it to the lab for secure disposal as soon as possible.
- Unwanted printouts must be placed into the locked box in the lab for secure disposal.
- On the lab machines, code may only be worked with in the specified computing environment.
- All malicious software, or self-propagating software, created in the lab must contain one or more safety mechanisms that prevent its operation if used outside the lab machines, or past December 2003.
- The passwords of all members in a group must be entered into a lab machine to log in; all group members are responsible for the work done or actions performed on that login session.
- There are unmonitored cameras in the lab recording periodic images; they may not be tampered with in any way.
- Transporting electronic equipment and electronic media across the media line may only be done by authorized computer science technical support staff and/or the course instructor. They must ensure that proper labeling and decontamination procedures are observed.

ISBN: 87-987271-6-8