

CS 497 Research Experience

Joseph Parker Fath

This semester, I started slow with research, but I ended up finding my way and making a solid contribution to my team's work. My work has been on the SALIVATE project, which began during the Fall 2005 semester. Last semester's team mostly worked on creating the concepts and specifications that would form the SALIVATE project, while this semester, we worked on creating a working prototype of the project. I contributed to this as well as the research of methods of BIOS protection, a more conceptual area that had not been addressed. I learned to think in terms of system security, considering all possibilities of attack, not just common ones. Overall, this semester's research has been a great learning experience for me.

This semester, I started on the SALIVATE project with a disadvantage. The rest of the team members, Michael Armbrust, Jay Gengelbach, and Greg Ose, had worked on the project during the previous semester and had been the founding members of the project. I had to learn about the project "from scratch." While the concepts were not too difficult, it took me a while to learn what specific files and programs were included in, for instance, SALIVATE's modified version of the GRUB software. Over the course of the semester, though, I became more familiar with the SALIVATE setup and more comfortable working with the team.

Once I found my way into the project, I was put to work on a few aspects of the project. I helped Greg Ose document many parts of the project that had been developed during the previous semester but not well-documented. We also worked on a practical, working prototype of the software's re-imaging distribution. When the system detects a compromised partition or file, it needs to re-image the partition (or file). After spending a few weeks running into problems with a FreeBSD CD-based distribution, we found a suitable Linux-based distribution (Slax, based on Slackware) to use. We then configured it to carry out the needed actions to connect to the administration server and securely (via SSH) download a trusted copy of the system's files. Finally, we documented our procedure on setting up and using our re-imaging distribution (SLAXIVATE) thoroughly. [1]

My next task was to research protection for the BIOS of SALIVATE-enabled systems. The system depends on each server booting from a CD-ROM in order to ensure the boot code remains unchanged. It is possible for an attacker to remotely overwrite the BIOS with either malicious code or just a copy of the same BIOS that is not set to boot from CD-ROM. Therefore, a solution is needed to defend against this kind of attack. While certain motherboard and hardware makers build BIOS protection into their hardware, part of the Poly² project is the use of commodity hardware. We cannot

require users to buy a specific motherboard or a motherboard with a specific BIOS. It was mentioned to me by my research advisors, Keith Watson and Eugene Spafford, that the Trusted Platform Module, a chip recently developed by a work group supported by many major computer companies, might solve this problem. After doing much research into this chip and running into many dead ends and obscure references to protecting the “validity” of the BIOS, I decided the chip probably does not do what we need it to do to protect the BIOS. Rather, it only makes sure the BIOS continues to communicate correctly with the TPM chip. I then looked into other possible solutions and wrote a paper about this.

[2]

This semester, I learned a few important things about computers and computer security. First, I had never needed to think with such scrutiny about every aspect of the security of a connection, server, or system. In this project, it was important to try to think of every possible situation where an attacker could compromise the system. We spent a lot of time saying “What if ... ?” and then answering ourselves by saying “Well, no, because an attacker could ...” Next, I learned a little bit about how system startup works in the UNIX world. As I've worked mostly with Windows systems my whole life, I could write my own autoexec.bat or config.sys file, but I was pretty vague on the idea of rc.local and similar files and settings. Finally, I learned that answers are not always easy to find and, in fact, do not always exist. The problem of BIOS protection is still one that needs to be solved if our process is to be considered completely secure. At this time, it appears there may not be a solution to BIOS protection (on commodity hardware) that completely fits our needs.

Related Work

Michael Armbrust, Parker Fath, Jay Gengelbach, and Greg Ose. “Salivate - secure architecture for loading initializing and verifying a trusted environment.” 2005.

[1] Greg Ose with Parker Fath. “Slaxivate, the salivate re-imaging distribution.” April 2005.

[2] Parker Fath. “The Applicability of the Trusted Platform Module to SALIVATE.” April 2005.