Apr 11th, 11:00 AM - 2:00 PM

Malware for Macintosh

Nathan C. Shinabarger
Cedarville University, nathanshinabarger@cedarville.edu

Josiah E. Bills
Cedarville University, josiah@adoniram.net

Richard W. Lively
Cedarville University, rwlively@cedarville.edu

Noah S. Shinabarger
Cedarville University, noah.shinabarger@gmail.com

School of ENGINEERING and COMPUTER SCIENCE
CEDARVILLE UNIVERSITY

Dolos

Malware for Macintosh

Faculty Advisor: *Professor Dudenhofer*
Students: *Josiah Bills, Richard Lively*
Nathan Shinabarger, Noah Shinabarger

Research & Scholarship SYMPOSIUM

Key Terms

- **Trudy/Alice** This paradigm is seen all throughout our project. **Alice** represents the innocent end-user—the person we are trying to attack. **Trudy** (short for intruder) represents us—the people trying to break into the system.
- **Module**: A module is a framework that contains a certain exploit or code to be executed on Alice's side—it often sends output back to Trudy, such as a screenshot or query result.
- **Exploit**: An exploit is code that is intended for malicious purposes, including activities like system monitoring and stealing information
- **Obfuscation**: Obfuscation is the intentional disguise or removal of incriminating evidences about the inner-workings of a program.

How Dolos Works

The diagram on the bottom right demonstrates how the Dolos program works. The left side shows commands and data coming from Trudy, which Alice then responds to with output. While the network interactions are apparent from the diagram, the actual data is encrypted and invisible to Alice (and only the Dolos program can decode it for her).

0. First, we assume the installation of Dolos on a target computer. This was an allowable assumption established with the client.
1. Once installed, Alice 'dials out' her presence on the target to Trudy. Trudy was already listening and establishes a connection and now can tell Alice to do whatever he wants.
2. Now that Trudy has control over the system, he can add modules. Upon adding a module, it is moved first into memory (**2a**), and then encrypted and put on disk (**2b**) so that it will stay on Alice's computer, even if restarted.
3. Now that the module is loaded on the system, Trudy can execute commands interpreted by that module. Once a command for a particular module is received, that module is loaded into memory (so that the nature of the program is hidden from Alice's system) and then executed.
4. Now that the program is running, we don't know how long it will take to get the output back. We wait for output for that command on another thread, while Trudy can continue having other interactions with Alice. Once the output is ready, the associated string or file is sent to Trudy.
5. Any modules that have been sent to Alice, such as in Step 2, will persist on Alice, being encrypted on the hard drive. Each time Alice logs in, her computer will dial-out to Trudy.

Abstract

Technology is a cornerstone of modern society. Unfortunately, it seems that every new piece of technology is accompanied by five computer-security breaches elsewhere. Most people associate hacks with Windows computers. This is a problem because Apple computers, and other non-Windows systems, are also extremely vulnerable to attacks and risk being compromised. Dolos is a piece of malware we developed intended to exploit the macOS Sierra operating system. It provides a framework for running exploits and comes built in with certain control and data exfiltration capabilities. Dolos also helps destroy the misconception of "the impenetrable Macintosh computer" by showing that Apple computers are also subject to cyber-attack. The creation of malware like Dolos requires an in-depth knowledge of different security mechanisms and protocols to find the points at which they break. Being aware of those weaknesses allows for the creation of stronger and more resilient systems in the future. We created Dolos to evade anti-virus software by encrypting communication between the attacker and the victim as well as encrypting the malicious code on the victim machine. This allows us to understand the strategies malicious criminals might take so that we may counter their strategies. By designing malware, we are better able to provide security and we begin on the path towards a safer and better world through designing more secure computing systems.

In Defense of Hacking

"The best computer security experts have the hacker mindset. When I look to hire people, I look for someone who can't walk into a store without figuring out how to shoplift. I look for someone who can't test a computer security program without trying to get around it. I look for someone who, when told that things work in a particular way, immediately asks how things stop working if you do something else. We need these people in security, and we need them on our side. Criminals are always trying to figure out how to break security systems. […] But if we have hackers working for us, they'll figure it out first -- and then we can defend ourselves. It's our only hope for security in this fast-moving technological world of ours."

- Bruce Schneier, Harvard Fellow, Board Member EFF