EE/CprE/SE 491 Weekly Report 7

Report Coverage: 10/15/2018 - 10/19/2018
Project Title: Security Orchestration Platform
Client: “The Company”
Advisor: Doug Jacobson
Team Members:

- Adam Crosser (Implant and EDR Testing Developer)
- Logan Kinneer (Implant and EDR Testing Developer)
- Daniel Limanowski (Frontend Lead)
- Vijay Uniyal (Frontend Developer)
- Justin Roepsch (Frontend Developer)
- Paul Chihak (Implant and EDR Testing Lead)

Weekly Summary

This week our team continued to conduct research and started writing test code to learn various development capabilities. We reviewed our design plan with the client and updated it from their feedback.

Past Week Accomplishments

Group Accomplishments

- Group phone meeting with client

Individual Accomplishments

- **Adam Crosser:** Researched on using finite automata to generate malware C2 traffic to match regular expressions. An attacker could then define a structured protocol (e.g. SMTP) using regular expression or another grammar/language and the malware would then generate traffic which would conform to that protocol. This would be similar to the existing “malleable C2” profile used by Cobalt Strike (https://www.cobaltstrike.com/help-malleable-c2). Installed cobalt strike and configured it to do domain fronting to frontable domain. Wrote a custom malleable C2 and malleable PE profile for cobalt strike. Experiment with using a structured language provided by cobalt strike to control the forensic indicators created by the malware both at the network level and in memory. Wrote code for using the Microsoft.Workflow.Compiler.exe utility to load arbitrary C# code into a microsoft signed process and researched process doppleganger attack and other methods injecting code into a process to evade endpoint detection solutions. Since security software implicitly trusts microsoft signed processes more than others on the system this is a valuable technique. It was originally developed and discovered by industry researchers at a boutique consulting firm known as SpecterOps (https://posts.specterops.io/arbitrary-unsigned-code-execution-vector-in-microsoft-workflow-compiler-exe-3d9294bc5efb)

- **Daniel Limanowski:** Research state management in ReactJS as it is a complicated, overwhelming mess but is essential for complex, dynamic web applications. Frameworks like Flux and Reflux have been built to manage this mess and I began playing with Reflux to understand what exactly it does and why it is necessary as our application grows.
- **Vijay Uniyal:** Continued research and learned/tested hot module replacement along with ES6 arrow functions in React. Able to better modify src/App.js to develop a more flexible React application. Succeeded in creating the basics of an internal state of my App component.
- **Logan Kinneer:** Researched methods of static analysis on malware and other software. Researched Cuckoo's reporting configuration setting as well as methods of displaying cuckoo's reporting stats within our own webapp and methods of submitting malware automatically to cuckoo.
- **Paul Chihak:** Created a basic test case where a user could create a packet of their choosing and then have the C2 conform it's communication using that packet. This is trivial to do for http however in order to support other common protocols would require each protocol type to be implemented. Therefore could not switch on the fly to less commonly used protocols or to conform C2 communication to look the same as other packets leaving the network.
- **Justin Roepsch:** Setup local test database for use in practicing sending logs of user actions to better evaluate what is needed to generate and preserve logs. In doing this, I also continued to practice using React.

Individual Contributions

Brief summary of individual team contributions given below.

Name	Individual Contributions	Hours this week	Hours cumulative
Adam Crosser	Wrote test code to understand tools and frameworks.	12	44

	Reviewed published academic and industry research in problem domain (please see previous section for technical details)		
Daniel Limanowski	Set up and used Reflux to learn about state management with ReactJS.	8	41
Vijay Uniyal	Hot module replacement testing finished along with research into utilizing ES6 arrow functions.	5	36
Logan Kinneer	Researched integration of Cuckoo with our own webapp.	7	37
Paul Chihak	Wrote test code to create a POC malleable C2.	6	37
Justin Roepsch	Test database for logs, more use of React	7	38

Plan for the Upcoming Week

- **Adam Crosser:** Continue to write test code and review published research and conference presentations. Continue to communicate with client and other industry professionals to further understanding of problem domain.
- **Daniel Limanowski:** Further my knowledge in state management with ReactJS. I want to begin researching how to implement a builder in our web application.
- **Vijay Uniyal:** Continue research into react moving on to ES6 Object initialization and Internal Component State. Assuming that ES6 Object initializing doesn't prove too difficult I will also move on to testing unidirectional Data Flow to manipulate the local state of the application.
- **Logan Kinneer:** Continue to look for the best way to integrate Cuckoo and Malice into our project and develop a good understanding of architectural patterns that may help in creation of the project.

- **Paul Chihak:** Continue writing test code to attempt to find other more effective ways to implement malleable C2. Furthermore I will keep researching how other malware C2 channels are implemented and review conference presentations.
- **Justin Roepsch:** I will practice rapid development of React that results in functioning applications, so implementation times will be smaller in the future.