Scam Booter

CPEN442

October 11

Zoy Huang (20026150), Ryan Koon (11062149), and Wendy Zhou (41378150)
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada

I. What is the problem that we addressed?

We addressed the issue of social engineering from technical support scams, which has caused an increasing trend of reported financial losses [1]. We proposed a solution that protects users with little knowledge about Windows' administrative tools. The asset at risk is the victim's bank account. The vulnerability is the victims lack of knowledge about Administrative tools on Windows. The threats are people pretending to be legitimate technicians and deceiving users about the state of their computer.

II. Why is this problem important?

According to Microsoft, there were 153,000 technical support scam reports worldwide in 2017, a 24% growth from the previous year [3, Fig 1]. In the same year, the Internet Crime Complaint Center (IC3) received approximately 11,000 technical support complaints that totals to a loss of almost $15 million. That was an "86% increase in losses from 2016" [1].

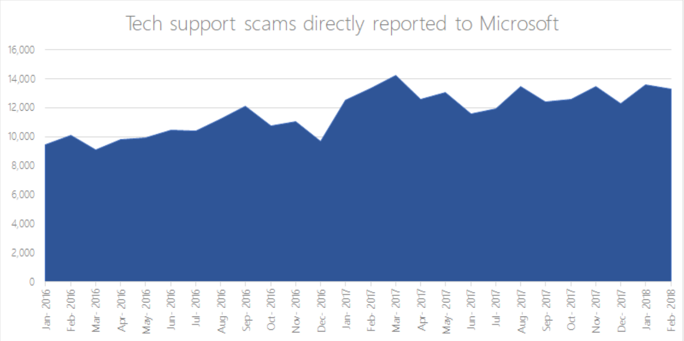

Fig. 1. Number of tech support scams reported to Microsoft

A study [2] conducted in 2014 discovered 1,688,412 unique visitors to scam sites and estimated a loss of at least $9.7 million. With this trend, it appears that technical support scams are not going away even with efforts to stop them [2, p. 8].

III. How is this problem currently addressed by others?

ROBOVIC, short for Robotic Victim, was a tool created in a study that collected data about technical support scams. It crawled the web to find websites, its visitors, and phone numbers used for these kinds of scams [2, Fig 2]. The investigators concluded that "the vast majority of AV users are likely not going to be protected against technical support scams" [2, p.7]. On average, 64% of 1624 malicious TLDs were only detected by 3.25 AV engines out of 68 engines [2, p.7]. Phone applications on Android detected "less than 1% of the 1,581 scammer-operated phone numbers" [2, p.8]. On average it took 44 days for a phone number to be reported as a part of a scam. Even worse, some mobile applications associated scam numbers with positive reviews and legitimate businesses such as Dell and McAfee. Microsoft has been making a wide range of improvements by "enhancing antivirus, email, URL blocking, and browser security solutions". TeamViewer has been displaying prompts to warn users about technical support scams. In terms of user education attempts such as public service announcements have been made. However, they were ineffective being on specific sites that were not known by the general population [2, p.13].

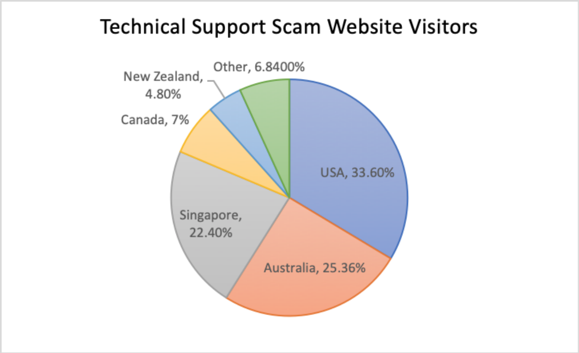

Fig. 2. Tech support scam website visitors by country

IV. What is the way we have proposed to address the problem?

Social engineering attacks thrive on human error based on an inadequate knowledge of the system being used. A novice user does not have the knowledge to recognize fallacies provided by the technical support scammer, nor do they have working knowledge of Windows tools used to manipulate them.

However, notably, scammers follow similar sequences of events and use similar techniques to convince their victims. The most common techniques are shown in [2, Fig 3].

Technique	% Calls
Stopped Services/Drivers	67
Event Viewer	52
Specific Virus Explained	50
System Information	47
Action Center	40
Fake CMD Scan	40
Netstat Scan	40
Installed/Running Programs	35
Browsing History/Settings	27
Downloaded Scanner	17
Reliability/Performance	15
Other (Temp, Registry)	13

Fig. 3. Techniques used by support scammers in order to convince their victims of a malware infection

We defined these events and techniques as "suspicious behavior". For example, the use of the administrative tool, "Event Viewer", qualifies as suspicious behavior. Our solution protects users that lack knowledge about the tools and techniques used in technical support scams by detecting these events on their behalf.

We addressed the problem by using the Windows API to build a Windows application that detects the suspicious behaviors. If a scam is detected, the application forcefully terminates any remote connections and informs the user of the event. We also explored options to prevent the application from being terminated. An option was to create a Windows service that manages the lifecycle of the application. There are other methods to make it even more difficult to shut down the application. However, those methods would likely be flagged by an antivirus solution. To ensure our application is compatible with antivirus programs, we uploaded our application binaries to VirusTotal to be analyzed by 68 antivirus scanners.

V. Why and in what respect is our way to address this problem better than those developed by others?

Existing solutions include informing the user through online information, blacklisting scammers' numbers and domain names, and antivirus solutions. Blacklisting scammer websites and phone numbers is not an effective method since the scammers can always register new domain names, create new websites, and use other phone numbers. Our solution focused on detecting a potential scam on the user's system and immediately brings them back to safety. It protects users who are experiencing potential phone scams and does not rely on numbers and IP addresses of scammers. The goal of our system was not only prevention but also detection and recovery. This can be considered a layer of defense inside a "fortress", which helps protect inexperienced users.

VI. How did we analyze the effectiveness of our solution?

We constructed a test suite that contains various forms of unique and documented technical support scams. For each scam, we made many similar attacks against a system protected by our solution. We assessed the effectiveness of our solution based on the number of detected attacks in the test suite.

VII. What was our plan for completing the project and submitting the report by the deadline?

A. *Research*

We researched scam detection heuristics to define suspicious behavior. The deliverables were documentation of specific applications that need to be opened in a specified time frame to be considered a scam in progress.

B. *Design*

We enforced design requirements throughout out project. The deliverables were UI mock-ups, use cases, and UML diagrams.

C. *Implementation*

We followed our design, used version control, and maintained best practices for software development. The deliverables were documentation on technologies and implementation.

D. *Testing*

We systematically verified and validated our implementation. The deliverables were documentation of scam and nonscam scenarios with the expected results, as well as test analysis statistics (e.g. code coverage).

E. *Report*

We continuously compiled the report and set milestones. The deliverables were a prefinal report on November 8 and a Final report on December 4.

F. *Gantt Chart*

We followed our plan with the following schedule.

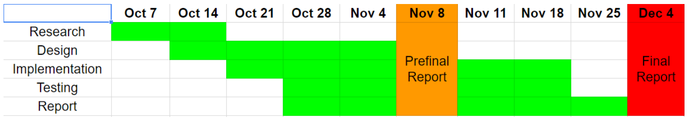

Fig. 4. Gantt Chart

References

[1] Federal Bureau of Investigation, "TECH SUPPORT FRAUD", https://www.ic3.gov/media/2018/180328, Mar. 28, 2018 [Oct. 6 2018]

[2] N. Miramirkhani, O. Starov, and N. Nikiforakis, "Dial One for Scam: A Large-Scale Analysis of Technical Support Scams," in Proceedings 2017 Network and Distributed System Security Symposium, 2017

[3] Erik Wahlstrom, "Teaming up in the war on tech support scams", https://cloudblogs.microsoft.com/microsoftsecure/2018/04/20/teaming-up-in-the-war-on-tech-support-scams/, April 20, 2018 [Oct. 6 2018]