Scam Booter

CPEN442

November 9

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

***Abstract*—Technical support scams are a growing issue as more financial losses have been reported year after year. Software companies like Microsoft and Google have been improving their software security and processes to block malicious websites as well as working with law enforcement to tackle the issue. There is no perfect solution and user education is the most effective method provided that the information reaches those that will benefit from it. Therefore, to provide an extra line of defense, we created ScamBooter to detect remote connections and analyze interactions with administrative applications on Windows. If malicious behavior is detected, we attempt to terminate the user's connection with the scammer, and provide details about the scam so that the user is better equipped to avoid falling for a technical support scam in the future. This further promotes user education regarding the issue.**

We built and tested our solution based on scenarios and statistics provided by a large-scale analysis conducted on technical support scams. Due to the rapidly changing environment of cybercrime, we can only deem our solution as a part of a multi layer defense system. Ideally, our heuristics in detecting a scam and the implementation recover from from a scam would be used to enhance an existing security software solution or built into an operating system. Alternatively, operating systems can restrict access to administrative applications not often used by general users. In addition, they should provide descriptions in plain English on what the applications can and cannot do before allowing the user to enable them. However, we cannot expect a user to read the description and make the expected choices under the direction of a scammer. Therefore, automated solutions like ScamBooter can still provide value in protecting users.

I. Introduction

A. 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.

B. 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]. In a recent study (September 2018) released by Microsoft, financial losses have become more consistent worldwide as some countries have experienced less losses while others have experienced an increase in losses [7, p.12]. Not only is financial loss an issue but 52% of surveyed users spent time finding out what was wrong with their computer and "three-in-four consumers who continued with a scam experienced moderate to severe stress" [7, Fig 3].

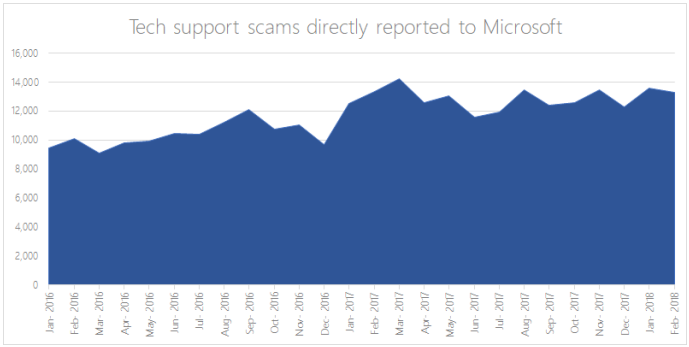

Fig. 1. Number of tech support scams reported to Microsoft

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].

C. Summary of the Designed System

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 4].

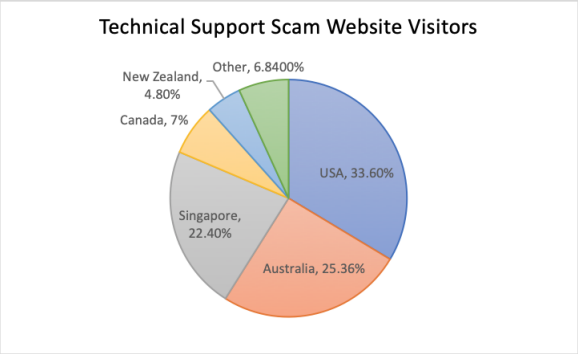

Fig. 2. Tech support scam website visitors by country

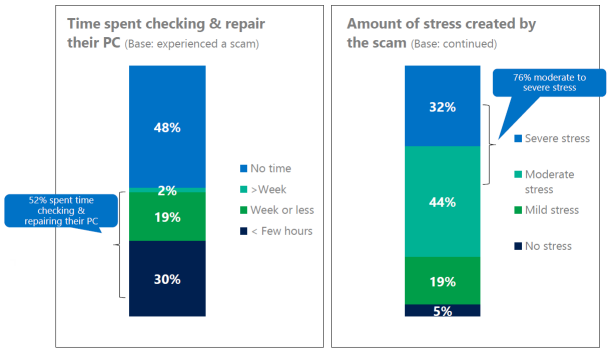

Fig. 3. Time spent fixing computer and stress levels

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, called ScamBooter, 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 commonly used remote access applications in this type of scam and informs the user of the event. We also prevented the application from being terminated by creating a Windows service that manages the life cycle of the application. In certain scenarios, we analyzed keystrokes to more accurately determine whether a scam is in progress. To ensure our application is compatible with consumer security solutions, we tested our programs in environments installed with security software from major security software vendors such as Norton Security and Sophos.

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

As an overarching goal, we aimed to protect users from malicious remote access to their computer and to educate them in a timely manner on technical support scams. This was also an exploration into how we can provide security via behavioral analysis without severely impacting the user experience of Windows.

D. Summary of Related Works

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 [antivirus] users are likely not going to be protected against technical support scams" [2, p.7]. On average, 64% of 1624 malicious TLDs (Top Level Domains) were only detected by 3.25 antivirus 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]. Having their customers as being a major target for technical support scams, Microsoft has been dedicating more resources into the issue and working with law enforcement to track down the scammers.

E. Summary of the Methodology

To test our detection rate, we compiled a suite of known remote connection programs and windows administrative tools. To test our key logging accuracy, we constructed a test suite that contains various forms of malicious strings and harmless strings. We assessed the effectiveness of our solution based on the total fraud and insult rate of the tests.

F. List of Contributions

Our source code has been made public on GitHub so anyone can build upon, improve, or use parts of our solution. The repository can accessed here: https://github.com/ryankoon/ScamBooter

II. Related Work

Besides the large scale analysis conducted by researchers from Stony Brook University, there has not been any new academic studies in this area [2]. However, there has been widespread media coverage of call center raids [6]. In fact, the increased usage of pop-up ad-blockers potentially reduced the likelihood of preventing exposure to scam websites [7]. Microsoft's Digital Crimes Unit has continued to conduct investigations, collect evidence, and work with law enforcement to combat technical support scams. This year, Google has started restricting advertisments that claim to provide technical support services by starting a verification program to ensure that only legitimate third-party technical support providers are using their advertising platform [8]. Scammers are developing new ways trick people in making a payment and certainly more effort has been put into blocking websites and ads. However, analyzing user behavior as the last line of defense is not seen in products for the mass market. This could be due to resource limitations of most consumer computers to be able to provide detailed and accurate analysis, as well as privacy concerns. The best solution is providing user education on this issue and there are plenty resources on such scams and many others [10]. The problem is being able to reach out to those that will benefit from the information.

III. Adversary Model

We developed an adversary model by analyzing patterns of technical support scams. This model facilitated our understanding of the problem and helped us to design our system.

A. Objectives

The adversaries are scammers who profit off of inexperienced Windows users. The objective of the adversaries is to convince the victims that their computer is infected by virus. After gaining the victims' trust, the adversaries force them to pay to fix non-existent problems.

B. Initial capabilities

The adversaries can call users, pretending to be from well-known tech companies. The adversaries can also set up malicious websites or advertisements to trick the user into calling fradulent technical support numbers.

C. Capabilities during the attack

The adversaries ask the user to download remote access software and gain access to their computer. They can open administrative tools to show system "errors", which are actually logs of normal operation. They can install malware on the user's computer, causing the system to crash.

IV. System Design

A. Detecting a remote connection

Detecting an active remote desktop connection is not as trivial as checking a flag. For Windows, that is only possible if the application is using the Microsoft Remote Desktop Protocol, which operates on specific ports. However, the remote desktop applications used by technical support scammers typically establish connections on port 80 and 443, the standard ports used by web clients to communicate with a machine. In addition, these ports are only used to establish a connection. Therefore, the resulting port is not the same for each session as it is assigned by the machine serving the desktop. Other non-standard ports are used by remote desktop applications, but they might differ between software, are undocumented, or the ports that they use may change depending on network conditions. Such a moving target will result in a high rate of false-positives. Instead of analyzing network traffic, our application analyses raw input data using the Windows Desktop API [4]. Every time a mouse click occurs, regardless of the application in focus, our application will check whether a handler to the device is provided in the raw input header. If so, it is a local device that is attached to the computer. If no handler is provided, it is an input made from an active remote connection to the computer. Using this method, we can distinguish between clicks from a locally connected mouse and clicks from a remote desktop application.

B. Detect the use of administrative tools

As illustrated in [2, Fig 2], scammers use specific Windows programs during attacks. 67% of scammers use services.exe to show the users a list of stopped services, and a large portion of scammers use Event Viewer, Netstat, Command Prompt, or a combination thereof. Since the calls to these tools are closely related to scams in progress, we decided to monitor the use of these administrative tools. In order to accomplish this, our application registers a WMI (Windows Management Instrumentation) event handler to receive notifications from the start of a process [9]. The application checks if the launched processes include services.exe, eventvmr.exe, mmc.exe, cmd.exe or netstat.exe.

C. Analysing keystrokes

To improve the accuracy of detecting a scam in progress, we log keystrokes when the Windows Command Prompt is in focus. Scammers frequently execute certain commands to trick the user into believing that there is malware on their computer. 40% of scammers employ a "Fake CMD scan" by running a command to list the tree of files, folders, and sub-folders at a certain path [2, p.7]. Our application will capture keystrokes that occur while the command prompt is outputting information. We will look for common words and phrases typed in such as "Virus detected". We employ this strategy because users are often tricked into believing that the system produced the output not knowing that user inputs are displayed after a command is done executing. To capture keystrokes we use a global hook open source library that will allow our application to learn about keystrokes regardless of the application in focus [5]. However, to protect the privacy of the user, we only keep keystrokes in memory and analyse them when it is destined for the Windows Command Prompt.

D. Terminating remote access applications

Stopping all network traffic would be a fail-safe strategy to terminate the connection between the user and the scammer. However, this will affect all other applications that require network connectivity which arguably, makes our application appear to be malware. To prevent such degradation to the user experience, we terminate common remote desktop applications used by technical support scammers instead [2, Fig 5]. Our application runs with elevated privileges to allow us to terminate remote access applications with elevated privileges and to counter process renaming done via a file rename. We access the file metadata to determine the original name of the application rather than relying solely on the process name.

Remote Administration Tool	Websites	Scammer abuse
LogMeIn Rescue	www.support.me www.lmi1.com www.logmein123.com	60%
CITRIX GoToAssist	www.fastsupport.com	21%
TeamViewer	www.teamviewer.com *Scammer-controlled*	12%
Other	www.anydesk.com www.gethelp.us www.supremocontrol.com	7%

Fig. 5. Remote access tools used by technical support scammers.

E. Principles of design

We applied the following principles to our system design.

- *Psychological Acceptability.* It is easy for users to install and start our system. Our system shows messages that inform users about potential scam activities.
- *Defense in Depth.* By implementing a Windows service program that launches on startup and restarts our application program, we prevented our application from being terminated by scammers. Our solution by itself is an extra layer of defence for the system in stopping scammers from communicating with users.
- *Open Design.* We made our source code public on GitHub so that it is available for anyone to evaluate and make more robust. With the rapidly changing threat environment, this will allow us to be more informed about changes to be made increase our system's effectiveness.

V. System Prototype

Our prototype consists primarily of a Windows application that detects suspicious activity and a complementary Windows service that manages the life cycle of the application. We developed the two programs using C# and .NET framework.

A. Windows application

The application is responsible for detecting and terminating potential technical scam activities. It monitors launched processes and detects remote connections using raw mouse input and keystrokes. Based on these two factors, the application predicts if there is possibly a scam and notifies the user.
In extreme cases, our application terminates remote connections by disconnecting the computer from the Internet for a short period of time. It then displays a message to explain the reason for disabling the Internet and re-directs users to a reputable page about techical support scams.

B. Windows service

The role of the service is to prevent the application from being terminated. It checks the list of running processes at a set interval and restarts the application program if the application has stopped running.

VI. Evaluation

A. Evaluation Methodology

We determined a list of success criteria for our program: fraud rate, insult rate, and psychological acceptability.

To assess the fraud rate of detecting programs used in a scam, we run the entire list of Windows administrative tools and popular remote connection software (seen in Fig. 4) against our program. The target fraud rate is 0 where our program detects each item on the list.

Based on our threat model, users who can trigger our detection system on their own have enough technical knowledge to be safe from technical support scams. Therefore, the insult rate for programs will not be a significant metric in our evaluation.

However, our program also performs key logging to detect intrusion. To assess the fraud rate of the key logger, we will test it against strings of input known to be part of a scam. To assess the insult rate for our key logger, we will test it against harmless common text found in online databases such as books, blogs, as well as random generated text.

A limitation of our design is that our detection works based on the a signature of program names. The program can only correctly identify remote access software that is known to be used for software scams. Therefore, we will not evaluate unknown programs, and expect signature updates to patch breaches using unknown programs.

One other important metric is psychological acceptability. Users should not be aware of our program running in the background. To assess this criteria, we ran the program, and check these places for traces of the program running: Task Bar, application switch window, timeline, action center, and program icons. Eliminating traces from these menus render our program invisible, and only seen as a background process. It also makes it more difficult for scammers to disable.

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] E. Wahlstrom, "Teaming up in the war on tech support scams," Windows Server Blog, 06-Jun-2018. [Online]. Available: https://cloudblogs.microsoft.com/microsoftsecure/2018/04/20/teaming-up-in-the-war-on-tech-support-scams/. [Accessed: 06-Oct-2018].

[4] "C# Get Mouse handle (GetRawInputDeviceInfo)", Stack Overflow, 2018. [Online]. Available: https://stackoverflow.com/questions/14584280/c-sharp-get-mouse-handle-getrawinputdeviceinfo. [Accessed: 09- Nov- 2018]

[5] ”rvknth043/Global-Low-Level-Key-Board-And-Mouse-Hook”, GitHub, 2018. [Online]. Available: https://github.com/rvknth043/Global-Low-Level-Key-Board-And-Mouse-Hook. [Accessed: 09- Nov- 2018]

[6] S. Chabba, “Indian Fake Call Centers Scam Americans Of Millions Of Dollars, 70 Suspects Arrested,” International Business Times, 06-Oct-2016. [Online]. Available: https://www.ibtimes.com/indian-fake-call-centers-scam-americans-millions-dollars-70-suspects-arrested-2427340. [Accessed: 10-Nov-2018]

[7] “Global Tech Support Scam Research - Global Summary, September 2018.” [Online]. Available: https://news.microsoft.com/uploads/prod/sites/358/2018/10/Global-Results-Tech-Support-Scam-Research-2018.pdf.

[8] D. Graff, “Restricting ads in third-party tech support services,” Google, 31-Aug-2018. [Online]. Available: https://www.blog.google/products/ads/restricting-ads-third-party-tech-support-services. [Accessed: 10-Nov-2018].

[9] ”Receiving a WMI Event”, Microsoft, 2018. [Online]. Available: https://docs.microsoft.com/en-us/windows/desktop/WmiSdk/receiving-a-wmi-event#event-consumers. [Accessed: 09- Nov- 2018]

[10] “Protect yourself from tech support scams,” Microsoft Support. [Online]. Available: https://support.microsoft.com/en-ca/help/4013405/windows-protect-from-tech-support-scams. [Accessed: 10-Nov-2018]