Breaching & Protecting
an Anonymizing Network System

Jason W. Clark, and Angelos Stavrou

Abstract—The protection of personally identifiable information (PII) is paramount for certain organizations. As a result, these organizations are currently employing both commercial products and open source Anonymizing Network Systems (ANS) as part of their standard network defenses to anonymize their outbound traffic. In this paper, we analyze the benefits and caveats of a typical deployment of an ANS. We call this the Case-Study Anonymizing Network System (CSANS) and we show that while network-level anonymity systems are better at protecting end-user privacy than having no ANS in place, they are unable to thwart de-anonymization attacks aimed at applications and private data of end-users. Indeed, the anonymity protection that ANS users enjoy can be bypassed by targeted attacks against both applications and users. To further explain our claims and quantify the different types of attacks and intruder capabilities, we introduce two threat models. We demonstrate and substantiate our claims using a targeted experiment against actual operational scenarios of users who are relying on the CSANS in a real organization to complete their research tasks. To that end, we set up an experiment to guide users (often referred to as researchers) to a website containing both non-malicious and malicious files. The downloading and opening of these files can enable an intruder to expose the true identities of the users even though the users were using a protected, isolated ANS connected machine. The main takeaway from this study was that both the administrators and CSANS users were not educated about quite prevalent attack vectors for compromising client systems and violating user privacy. Finally, we discuss the danger of using location-based services while on an ANS and we show that network-level anonymizers are insufficient. We conclude with recommendations for defending against anonymity attacks that aim for the application-level.

Index Terms—Anonymizing Network Systems, Security, Privacy, User and Application Attacks, User Training and Awareness, Traffic Analysis, Defenses and Countermeasures.

I. INTRODUCTION

THE Internet offers a low-cost, low-risk, and high-value of return intelligence gathering and archival system. With the advent of social networking and information aggregation, the Internet has become an information highway storing copious amounts of information that is typically unintended

Jason W. Clark and Angelos Stavrou are with the Computer Science Department in the School of Information Technology and Engineering at George Mason University, Fairfax, VA 22030 USA (e-mails: {jclarks, astavrou}@gmu.edu).

for the intruder, but is freely available and relatively easy to discover for an analyst.

Often users of computer systems perform operations on the Internet, such as searching for information, which they would like to keep anonymous. One popular way to help protect a user's identity against a possible intruder is to implement an anonymizing network system (ANS). The field of anonymous communications and specifically, ANS started in 1981 with David Chaum's Mix-net [3]. In general, an ANS is a system or set of systems that is designed to protect a user's identity while they are using a computer system that can access the network. Therefore, an ANS is a tool that attempts to make activity on the Internet untraceable.

As described by Pang, et al., the first goal of an ANS is aimed at preventing the true identities of specific hosts from being leaked such that an audit trail of user activity cannot be formed. The second goal is to prevent the true identities of internal hosts from being leaked such that a map of supported services can be constructed. The third goal is to prevent the leakage of specific security practices within the publishing organizations network [28].

When implemented correctly, an ANS will access the Internet on the user's behalf and in theory protect personally identifiable information (PII) by hiding the source computers identifying information along the way. The goals of any ANS is to first hide the structural information about the network on which traces are collected and second to prevent the assembly of behavioral profiles for users on that network, such as the sites that they browse [1]. The ultimate goal of Internet anonymization is to allow a host to communicate with a non-participating server in such a manner that nobody can determine the user's identity.

Simply implementing an ANS will not necessarily protect a user's identity. However, it is safe to say that ANS are better than no system at all. As we will describe in a later section, a user will have to be careful and cognizant of what sites they visit and what actions they perform even when "protected" by an ANS. Often users of ANS gain a false sense of security as they are under the impression that they are completely protected and anonymous. As we will show, this assumption could not be further from the truth.

There exists a plethora of different ANS including the

popular Anonymizer [1] and Tor [2] that are available on the market today. Other ANS solutions exist and they include but are not limited to: Browzar [20], JAP [21], Privoxy [33], and SafeSurf [22]. As a result, these ANS are predominantly built with the goal of protecting the network layer of the OSI model in mind. While this does provide a certain level of anonymity and identity protection compared to no system at all, it does leave itself vulnerable to a variety of different styles of attacks.

The focus of this paper is the analysis of the actual anonymity that an ANS provides. We assume that this system is going to be used to protect the identity of a user or an organization in a real-world setting using operational scenarios. The actual setup employed a widely used variation of the ANS offered by Anonymizer, Inc. To gather experimental evidence, we developed a case-study based on a typical ANS deployment in a large US-based organization. We call this version the case-study anonymizing network system or CSANS. The CSANS represents a typical implementation of an ANS that is available today. The inner-workings of the CSANS is described in detail in section III B.

The purpose of this study is to introduce the shortcomings of the CSANS by conducting attacks that focus on the actions of the users. We were approached by the management team at the case-study agency about the security and privacy of the CSANS. Specifically, the CSANS management was interested in determining whether the users were performing actions that could lead to their identity being discerned. The goal of this study is to allow defenses to be developed that can help reduce the likelihood that attacks against ANS are successful.

The main argument that we make is that organizations and users can't solely rely on an ANS to protect their identity. Rather we warn users to be cognizant of their actions when on an ANS and provide users recommendations and defenses to help protect against the intruder attack vectors we outline.

This study is significant because it shows that users behind an ANS are not as protected as they need to be. By completing this study, we show which attacks are the most successful and more importantly why they are successful.

The concept of researching ANS and their ability to protect a client's identity is something that has been studied for the last decade. Also, the concept of basic attacks by analyzing web traffic and attempting to perform application-layer attacks against systems and users is not original either. However, the merging of the two concepts is a new idea.

This paper focuses on explaining how the security of ANS such as Anonymizer, Tor, and CSANS can be "broken" by performing a variety of different and often unsophisticated attacks. It is this new outlook that makes the research original and worthwhile in doing.

Furthermore, our research is original in the sense that it is a hard look at all the essential ingredients of a specific attack used against a real-world ANS. We will conclude the research

by offering recommendations to users of ANSs as to best defend against the attacks that we outline.

While it is true that we are predominantly focusing on CSANS, our research will be able to scale to many other ANSs. The reason for this is because they all have the potential to suffer from the same kinds of attacks that we describe.

The main contributions of this paper are we:

1) Introduce the concept of ANS and the current research literature in the field of ANS attacks
2) Examine how real users behave while on a real-world implementation of an ANS
3) Show that unsophisticated attacks can be quite useful for an intruder
4) Prove that users and a lack of education remain the weakest link when trying to keep privacy intact

II. SETUP

This paper considers three main methods that could be used by an intruder to possibly discern the true identity of the user. The first method of potentially de-anonymizing a user is by a user clicking on a file that has logging capabilities enabled. This would allow the intruder to perform traffic analysis and find a correlation between the anonymized IP traffic and the user's real IP traffic.

Next, the intruder could create a malicious file that, when opened with a vulnerable application on a vulnerable operating system, could exploit the user giving the intruder the ability to take control of the system.

The last method we consider is attributable searching while on a non-attributable ANS. We define attributable searching to be the action of performing search queries that could yield information that might potentially de-anonymize the user. The intruder could gather attributable information based on search queries including, but not limited to, local weather, sports, and restaurants. Figure 1 shows the different de-anonymization methodologies discussed in this paper.
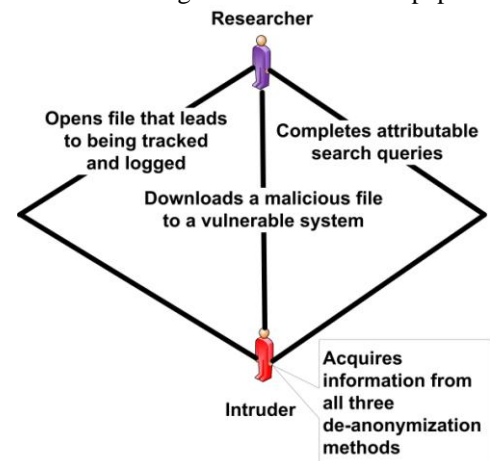

Fig. 1. Possible de-anonymization methods.

The experiments conducted required the research manager of the case study organization to design a fictitious research task. The research task was designed entirely by the research manager with no assistance from the authors of this paper. Therefore, we conducted the experiment on a real operational environment and gathered the reported results from field experiments using real human subjects.

Figure 2 shows the relationships between the key players and the entities associated with the experiment.
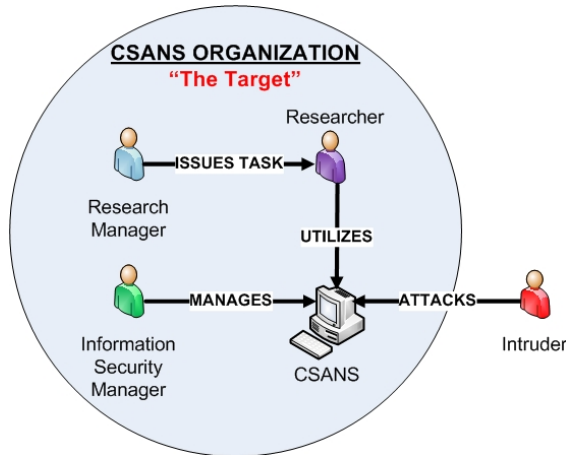

Fig. 2. Relationship between key players in the experiment.

As depicted in Figure 2, the research manager issues a specific task to the researcher which will require the use of the CSANS to accomplish. The information security manager is solely responsible for developing, managing, monitoring, and securing the CSANS. The intruder is assumed to be someone outside the organization who is interested in breaking the anonymization of the CSANS organization and its users in an effort to gain the researcher's PII. While it is possible that the intruder is an inside threat, we assume that the intruder is not associated with the CSANS organization. Figure 3 shows the interactions between the key players and the entities associated with the experiment.
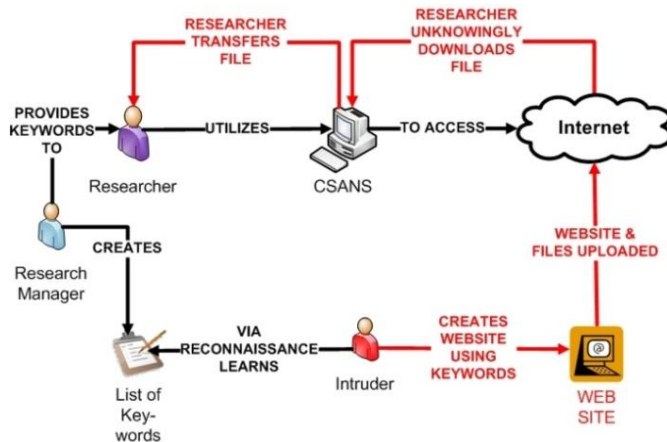

Fig. 3. Interactions between key players and entities in the experiment.

A. Ethics and Institutional Review Board for Human Subjects

Given the sensitivity and potential ethical issues associated with this type of experiment, some organizations and its users may feel that purposely trying to compromise systems and discover PII is a violation. However, we were authorized to conduct this experiment against the CSANS users with the support of the internal review board and permission of upper management.

To address any ethical issues that could stem from a study that involves humans, we requested the permission of the organization's review board, who subsequently authorized this experiment to ensure that the CSANS and its users were following best practices when it came to protecting the anonymity of the organization.

One of the major difficulties we encountered was to prevent people not associated with the experiment from accessing our website and downloading malicious files on to their system. We considered putting up a firewall to block access to our website except from a certain range of IP addresses. This proved to be difficult as the CSANS IP address changes randomly on a daily basis as part of the anonymization process.

Next, we considered password protecting our website, but were concerned about negatively impacting the experiment. This is because it would not have made sense for the researchers to know the password associated with our experimental website. However, we felt that this was the best option and worth preventing the risk of adversely infecting unsuspecting users.

Specifically, a password was used to protect the malicious files listed in Table III and the password was given to the research manager. The research manager then created a fictitious story as to how he learned the password and passed it on to his researchers as part of the task. While not a perfect solution, we did attempt to prevent people who accidentally visited our site from infecting their machines. We leave it to future experiments and research to find a more elegant solution to this problem.

B. Roles and Responsibilities

In Table I, we describe the roles and responsibilities of several key people who participated in the experiment.

TABLE I
ROLES AND RESPONSIBILITIES

Role	Responsibilities
Research Manager	Authorizes experiment Creates and provides list of keywords to researchers Manages researchers Authorized to stop experiment
Information Security Manager	Develops, secures & monitors CSANS Assists with security-related issues
Paper Authors	Act as intruder Develop experiment Conduct experiment Create website and corresponding files Capture and analyzes data Document and report on findings of experiment
CSANS Users (Researchers)	Do not have knowledge of experiment Complete research task independently with no collaboration
CSANS Organization	Management has knowledge of research task or experiment whereas user community does not

III. BACKGROUND INFORMATION

In this section, we briefly discuss anonymity on the Internet, inner-workings of the CSANS, targeting and profiling, current literature, Google location, and an introduction to our intruder threat models.

A. Anonymity on the Internet

Depending on the individual reason for accessing the Internet, anonymity may or may not be an important issue. For example, anonymity requirements may not be a concern for someone who is using the Internet to access a news site. Similarly, there is a growing trend for people willing to disclose their identity by posting personal information about themselves on a social networking site.

Conversely, users in defense and intelligence-related fields almost always have a requirement to remain anonymous when researching and compiling data from other (hostile) countries; if their identity is known, it makes completing the research far more difficult and possibly even dangerous. For example, the groups and countries that they are researching could potentially retaliate in response to the research being conducted without their consent. For the purposes of this paper, the CSANS organization consisted of real users who needed to remain anonymous in order to complete the research tasks that they were given.

B. Inner-workings of the CSANS

The CSANS is a variation of Anonymizer offered by Anonymizer, Inc [1]. The CSANS has been designed to prevent an open Internet connection from being exploited. All traffic is routed through dedicated hardware housed in a secure facility. Only authorized administrators are given access to this facility. The CSANS maintains a large pool of IP addresses and is able to easily rotate and replace IP addresses that have possibly been compromised. The users of CSANS should expect to see their IP address change about once every 24 hours. This concept is known as the IP rotator scheme.

In this scheme, anonymized users are mixed with "regular" worldwide consumers. The behaviors of both types of users (anonymized and regular) are mixed together in an effort to prevent behavioral patterns from being noticed. The regular users are diverting attention away from the anonymized users by virtue of the search queries and traffic that they are generating.

The hardware associated with CSANS is an enhanced version of the Juniper Virtual Private Network (VPN) box. The CSANS includes two interfaces: 1) the front-end and 2) the back-end. The front-end interface has a specific network IP address, and the back-end has a completely different network IP address. For security purposes, the IP addresses are visible internally, but not externally. The firewall routing for CSANS is designed to protect users from accidentally visiting sites associated with the specific organization that implemented CSANS. The routing rules are set to essentially mask the source and destination as to provide a second layer of anonymity.

The CSANS is hosted on a Windows server virtual machine, and users of the CSANS have to remote desktop into the system to get access to CSANS as seen in Figure 4.
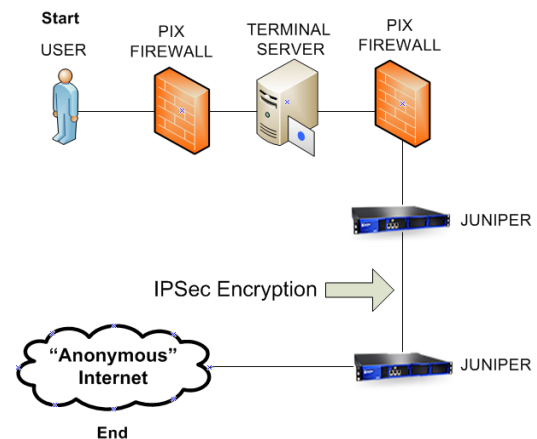

Fig. 4. CSANS Architecture.

The CSANS environment is locked down so that only certain users have administrative privileges. Since this particular implementation is a shared environment, all user requests including software installations need to be authorized. This follows best practices for the security aspects of the server. Specifically, in an effort to keep the CSANS clean from malicious code, the system administrators incorporate the use of a virtual machine snapshot. Once a week Windows updates, anti-virus updates, and software installations are performed resulting in a new snapshot being taken. If the

current snapshot becomes corrupted or infected with malware, CSANS would be able to revert back to a known good snapshot.

One of the key features of CSANS is the ability to transfer files from CSANS back to the client OS. While this is extremely helpful to the end-users in terms of usability and functionality, it is also a possible security concern. The rationale for implementing the transfer file component is because the CSANS users have a requirement to share files and to create and analyze reports using software from their computers.

The developers of the CSANS implemented a simple file sharing system. The file sharing system utilizes a FTP client on their machines along with a FTP server that could run from the CSANS environment. The CSANS users simply save their files to a "Home" directory and using FTP moved the files to their client machine.

C. Targeting and Profiling

In this section, we discuss targeting a specific organization and creating a document to be used in the attack. The goal of the intruder is to get the targeted users to download the malicious document. To do this, the intruder needs to profile the organization so the intruder can create documents that are appealing. Ideally, the documents that are uploaded on the website will be related to a topic that is of interest to the user as shown in Table III. In addition, most users feel comfortable and safe opening Adobe PDF and Microsoft Word files.

Therefore, if the intruder can customize the document to make it look like it was an organizational newsletter, job posting, company form, etc., it is more likely that the document will be downloaded [5]. If intruders can build up a good data set, it would only be a matter of time before a user comes to the intruder's website and downloads something malicious.

The attack that we will demonstrate is based on the idea that the intruder can use "common and public" knowledge about a particular organization and its users. If, for instance, the intruder can infer and attract attention about the target, it will be more likely that the attack will succeed. Through profiling, an intruder could potentially identify the CSANS organization and their research tasks and create documents that are related to these same areas of interest.

D. Survey of Current Research on ANS Attacks

Almost all ANSs, including the one used for this case study, are vulnerable to a variety of different types of attacks. In this section, we describe current research that is being conducted to attack ANSs. The literature review considers traffic analysis, application, and network-based ANS attacks. In addition to what was previously discussed, there is another significant body of work in the field of traffic generated by real-time ANSs.

Schear and Nicol [6] discuss the idea that network encryption, both at the packet and session layer, is used widely for securing private data. They use simulation and an analytical model to examine the impact on user experience via a scheme that masks the behavior of real traffic by embedding it in synthetic and encrypted cover traffic.

Coull, et al. [4] attempt to solve the problem of publishing data that can potentially leak sensitive information about the publishing organization. The article introduces the problem by suggesting that it is imperative that trace and log data be made publicly available for verification and comparison of results. The authors attempt to conduct an analysis and create techniques to infer sensitive information from these network traces. The study, as performed by the authors, demonstrates that there are more substantial forms of information leakage that inherently compromise current anonymization methodologies.

Wang, Chen, and Jajodia [9] describe the concept of watermarking network traffic in order to break the ANS. The watermarking aspect in this case can potentially allow an intruder to influence the traffic thus allowing them to discern the identity of the user. The authors described how they were able to successfully penetrate the Anonymizer, Inc. "Total Net Shield", the "ultimate solution in online identity protection" [9], which is almost the same ANS that is used in our case study.

Coull et al. [10] improve on previous research done on reconstructing web browsing activities from anonymized packet-level traces. This is accomplished by accounting for real-world challenges such as browser caching and session parsing. This research evaluated the effectiveness of the author's techniques by identifying the front pages of the fifty most popular websites on the Internet.

Hopper, Vasserman and Chan-Tin [11] present two attacks on low-latency ANS schemes. The first attack allows a pair of colluding websites to predict, based on local timing information and with no additional resources, whether two connections from the same Tor exit node are using the same circuit with high confidence. The second attack requires more resources, but allows a malicious website to gain several bits of information about a client each time a user visits the site. The authors evaluate both of their attacks against the Tor network and the MultiProxy proxy aggregator service.

Casado and Freedman [12] state that online services often use IP addresses as client identifiers when enforcing access control decisions. The idea of the paper is to determine the impact that edge technologies such as NAT, proxies, and DHCP have on the utility of using IP addresses as client identifiers. Specifically, whether the IP address of an incoming client is a useful identifier of access control decisions and to explore the extent to which edge opacity obscures a server's view of the client.

Coull et al.'s [13] primary concern is to evaluate the efficacy of network data anonymization techniques with respect to the privacy that they afford. Specifically, the authors are trying to make the network flow uniquely

identifiable even after it has gone through the ANS. They are also considering techniques for evaluating the anonymity of network data and to simulate behavior of an intruder whose goal is to de-anonymize the objects (host or web pages).

Jung, et al. [14] described the design and implementation of a system called Privacy Oracle that is capable of finding application leaks. The authors attempt to solve the problem of application leaks by using black box differential testing. The creators of Privacy Oracle are most interested in what information was leaked, when it is exposed, and who can receive it.

Scott and Sharp [15] use new tools and techniques which address the problem of application web security. The authors describe their solution to address the problem of application layer web security by describing a scalable structuring mechanism facilitating the abstraction of security polices from web applications developed in heterogeneous multi-platform environment. The second aspect to their solution is to represent a tool which assists programmers developing secure applications which are resilient to a wide range of common attacks. Finally, the third aspect is to report results and experiences arising from the implementation of these techniques.

Huber, Mulazzani, and Weippl [29] completed a full-analysis of HTTP usage and information leakage for the Tor network. The authors performed an analysis of the web browsing behaviors of Tor users to show which websites were of interest to them. Next, they determined an upper bound on how vulnerable Tor users were to sophisticated de-anonymization attacks.

Kadloor, et al. [30] were able to show that a remote intruder can obtain significant traffic timing and volume information about a particular user simply by observing the round-trip time of the pings.

Wandracek, et al. [31] introduce a novel de-anonymization attack that exploits group membership information that is available on social networking sites. This paper illustrates the tactic of using a malicious website to launch a de-anonymization attack to learn the identity of the visitors. The advantages of this type of attack are that it has a low cost and has the ability to affect a plethora of users.

Chaabane, Manils, and Kaafar [32] analyze Tor in the wild by setting several exit notes and distributing them worldwide. This is done to show that Tor is actually being misused as most observed traffic belongs to P2P applications.

Abbott, et al. [34] completely detail and describe a new attack on the anonymity of web browsing with Tor. The attack focuses on the user's web browser and works by sending a distinctive signal over the Tor network that can be detected using traffic analysis. Similar to the attacks we demonstrate, both the attack and the traffic analysis can be done by an intruder with limited resources.

Based on a review of current literature, we found that many security researchers are studying attacks that focus on traffic analysis, network-layer based attacks, and some application-layer based attacks. Furthermore, the focus of this research seems to be with Tor and not Anonymizer.

We introduced several different anonymization schemes and concepts including ANS, network trace anonymization, and to a lesser extent, proxy systems. While each of these systems have unique properties and disjoint security goals, we felt compelled to provide an overview of the current state of attacks against ANS. We introduced de-anonymization attacks that are complex and detailed in nature when compared to the attack that we are demonstrating. What we are suggesting is users and certain applications, if successfully compromised, can expose the organization, its users, and its tasks even if an ANS is implemented.

In fact, ANS are not designed to prevent basic attacks against applications and the users. The attacks we describe are quite effective in terms of how they can be used to produce user and computer information such as IP addresses, email addresses, source location, timestamps, and other pertinent system information.

.Location-based Exposure (Google Location)

While not an attack by definition, the Google Location feature has a major impact on the security and privacy associated with the users of Google. Specifically, the Google Location technology in certain cases identifies the user's true location. Google has massive amounts of technology and tools to be able to analyze traffic and find correlations between the user's search criteria and the user's likely location(s).

We define the concept of location-based exposure to be when users who pass through an ANS search for local: businesses, politics, weather, sports, etc. Essentially, users forget or are unaware that they are performing an attributable search that should not be done while on an ANS. As a result of this location-based exposure, Google is often able to correlate these searches and show the user's real location regardless of whether they are behind an ANS or not.

For example, suppose that the anonymized IP address of a given user is 123.45.67.89 and it shows that the location is anonymized to be in Ghana. However, multiple ANS users begin to search for Denver, CO weather conditions and the score from last night's Denver Nuggets basketball game. Google is able to identify this behavior and, given enough searches and traffic, will assume with a high likelihood that this IP address is not associated with Ghana, but instead with that of Denver, CO. The user's behavior is being used by Google to find the most likely location from which this user and corresponding IP address is coming. Given enough of this data, the Google Locator actually becomes quite accurate.

This problem is not necessarily specific to Google; although they are by far the biggest actor in developing this technology. Note that Google is not attacking in an effort to be malicious or uncover the true identity of any organization. Instead, they are simply trying to be more precise with their

search results and advertisements. From Google's perspective, they want to be both complete and accurate with their search results.

This particular attack vector would almost certainly be useful in determining whether or not the user was behind an ANS. By the very nature of location-based exposure, the browsing history would be known as illustrated in the above Denver search query example. It might be difficult for an intruder to determine with certainty with what organization the user belongs. However, they may be able to narrow it down by geographic location, thus potentially leading to the CSANS user's physical location.

Intruder Threat Model

Threat modeling is a method of assessing and documenting the security risks associated with an application. The threat model for CSANS is based on the process outlined in Swiderski and Snyder [7].

First, we examine what PII the intruder seeks. The intruder is most interested in the true identity of a user behind the CSANS. Such information could include:

- User's real name
- User's place of employment (organization name)
- Real IP address (non-anonymized IP address)
- Browser configuration and browser history
- Sensitive documents that reside on users system
- Software and programs currently installed on CSANS
- Configuration and implementation of CSANS

One method for identifying and categorizing threats is known as STRIDE. This method is a classification of the effects of realizing a threat and it stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege [7]. The STRIDE classification for various vulnerabilities is listed in Table II. Next, we consider internal and external threats. The following are a few threats associated with the CSANS.

- The CSANS users have been known to post sensitive documentation to fake email accounts for the purposes of storing data for future use.
- The CSANS administrator has full control over all the user accounts and passwords.
- The CSANS system has directories of sensitive documents that are stored on the server itself (before it is transferred). By logging into the server, an intruder could view these documents.
- The CSANS has a built-in transfer system that is designed for convenience rather than security.
- The anonymization is being done by a third-party.
- Users have been known to perform attributable search queries while on the CSANS.

In addition, we included a DREAD rating for each of the CSANS vulnerabilities that we identify in Table II. The DREAD rating is a method for characterizing the risk associated with vulnerabilities. The DREAD rating comprises Damage potential, Reproducibility, Exploitability, Affected users, and Discoverability [7].

The DREAD rating for the vulnerabilities associated with CSANS uses a three-point rating scale. A rating of one indicates that the impact is minor, the number of systems impacted is a small percentage, and the likelihood that the attack is identified is low.

In general, the lower score equates to minor damage, less reproducibility, less exploitability, fewer numbers of affected users, and less likely to be discovered. Conversely, a rating of three equates to major damage, more reproducibility, more exploitability, larger numbers of affected users, and greater likelihood for the attack to be discovered. The final DREAD rating score is simply the average of the scores for each of the five aforementioned categories.

Table II displays a non-exhaustive list of several vulnerabilities that the CSANS intruder could take advantage of along with the corresponding DREAD ratings.

IV. METHODOLOGY

The following sections describe our experimental methodology. Specifically, we discuss the experimental goals and objectives, experimental setup, and the creation of a fictitious research task and website.

A. Experimental Goals and Objectives

The primary goal of the experiment was to determine the extent to which we could use application and user-based attacks to discern the true identities of the CSANS users. Also, we wanted to show that network-level anonymization is not enough and that there needs to be better education to promote better anonymization.

TABLE II
VULNERABILITIES ASSOCIATED WITH THE CSANS

Vulnerability 1: Intruder able to determine browsing patterns	
Description	An intruder inspects the traffic as it goes through the server and onto the anonymized network
STRIDE Classification	Tampering Information Disclosure
DREAD Rating	Damage Potential = 3 Reproducibility = 1 Exploitability = 3 Affected Users = 3 Discoverability = 2 DREAD Rating = 2.4

Vulnerability 2: Intruder attacks the anonymization process	
Description	The intruder attacks the de-anonymization servers and/or the processes associated with the endpoint
STRIDE Classification	Tampering Information Disclosure
DREAD Rating	Damage Potential = 3 Reproducibility = 1 Exploitability = 2 Affected Users = 1 Discoverability = 1 DREAD Rating = 1.6

Vulnerability 3: Intruder performs any number of the attacks outlined in the literature review	
Description	By performing any number of the attacks outlined in the literature review
STRIDE Classification	Tampering Information Disclosure Denial-of-Service
DREAD Rating	Damage Potential = 2 Reproducibility = 2 Exploitability = 2 Affected Users = 2 Discoverability = 2 DREAD Rating = 2.0

B. Experimental Setup

To accomplish this goal we created documents that were used to both track and infect the CSANS users. We also examined the impact of users performing attributable search queries while on CSANS. We relied on analysis of traffic that we captured to find key points of interest.

The experiment for our case study lasted for two months. There were twenty CSANS users that were included in the experiment and they were asked to conduct all research within the confines of the CSANS.

Prior to the start of the experiment, we created and hosted an inconspicuous website and ensured that both the non-malicious and malicious files were tested before being uploaded as was depicted in Figure 1. Furthermore, we ensured that file logging and tracking was enabled to allow us to capture statistics from the web server. The website utilized various sources for content and was approved by the research manager as being enticing enough to attract the researchers.

The next step was to ensure that the website could actually be found via a simple search. By using Google search engine optimization and similar search ranking tools, we found that our website could in fact be located by virtue of simple search logic. A major aspect of the proposed study design was to determine the attack vector. For the purposes of this experiment, we focused exclusively on Adobe PDF.

The design of our study focuses on three major aspects, namely impact, stealth, and depth. The impact aspect measures how many CSANS users actually accessed the website and clicked on the files. The stealth aspect indicates how many users observed and notified management about the website and/or malicious file. Lastly, the depth aspect quantifies the quality of the malicious file as it relates to the amount and value of PII that a potential intruder could discern.

C. Fictitious Research Task

The CSANS users were specifically asked to complete a research task that focuses on the commercial airline industry. This fictitious research task is like any other task that the CSANS users would typically perform.

To begin the experiment, the research manager provided the CSANS users with a list of keywords to search on as part of the task. To protect the anonymity of the research tasks being done at the case-study agency, the actual and specific keywords have been sanitized and modified to appear more generic. However, to give the reader a frame of reference, we mostly considered keywords associated with the airline industry including: "Airports," "Flight Times," "Pricing," "Mileage Rewards," "Safety and Security," and "Jobs."

The assumption here is that the keywords are general enough to be known by a potential intruder as part of their targeting and profiling. Therefore, simply protecting the keywords is not a viable security measure.

D. Website

The necessary pre-requisite for this experiment was to design and build the aforementioned website. We researched each of the keywords independently and extracted content from other related websites to build into our own custom website.

Since the goal of the experiment is not a demonstration in website design, we developed a basic HTML website and hosted it on a third-party hosting site. The website corresponded to the different topics associated with the keyword list, and using simple logging techniques, the website can log the actions of the users.

The files were uploaded to the "documents" section of the website. The files included documents associated with the keywords. Each file was randomly selected to fall into one of three categories.

The first category was that the files were original and

completely clean. The second category was that the files were embedded with a JavaScript (app.launch) function that would automatically open a new webpage in the user's browser. The file would open a webpage called www.website.com/3490 where 3490 is a random number that can be easily identified in the data logs.

The third category was that the file itself was embedded with malicious code via the Metasploit Framework (see Section IV E). If the malicious code is opened in a vulnerable version of Adobe Reader on a vulnerable Windows OS, the user's machine could possibly become compromised. Once compromised, it would be trivial for the intruder to de-anonymize the user. Table III shows a sample summary of the document name, the corresponding category, and a description of the exploit where applicable.

TABLE III
SAMPLE SUMMARY OF DOCUMENT AND ACTION

Document (.pdf)	Category	Action
Airports	#1 Clean	Downloads and opens
Flight Times	#2 Redirection	JavaScript app. launch to random page on website
Pricing	#2 Redirection	JavaScript app. launch to random page on website
Mileage Rewards	#3 Malicious	Redirects to webpage being listened to by Metasploit: multi/handler listener
Safety and Security	#3 Malicious	Infected w/ "Aurora" exploit
Jobs	#3 Malicious	Infected w/ "Adobe (Collect_Info)"

E. Metasploit

In order to create the website's malicious files, we relied on the Metasploit framework [8]. The Metasploit framework was built to provide useful information and tools for penetration testers, security users, and intrusion detection system (IDS) signature developers. The Metasploit framework was created to provide information on exploit techniques and to create a functional knowledge base for exploit developers and security professionals.

The intruder would utilize the Metasploit framework to create attacks that target items such as specific applications, operating systems, and web browsers. The exact steps of finding the exploit, setting and executing the payload, and listening are outside the scope of this paper.

F. Capturing Traffic

We also consider a second threat model associated with the intruder's ability to capture traffic which is shown in Table IV.

In the ISP case, the anonymized traffic blends with the rest of the regular traffic from subscribers to that particular ISP. Assume that the ISP is Comcast or Cox Communication. There are tens of thousands of users that are accessing the Internet by virtue of these ISPs. In addition, there are several search queries on any number of topics that are being performed, and

TABLE IV
THREAT MODEL: CAPTURING TRAFFIC

Source of Traffic (Example)	Success / Complexity	Attack Level Success Rate
ISP (Comcast, Cox)	Low / High	Low
Service (Facebook, Gmail, Amazon)	Low / High	Medium
Custom Website	Low / High	High

this traffic will blend together.

As it stands, the attack level is low because it would be difficult for an intruder to determine if the search queries and the traffic were from a truly anonymized user or a "regular" user who coincidentally was searching for the same information. Also, it is more difficult to obtain access to a big ISP and monitor all communications. The complexity and cost for doing so is high when compared to poisoning Google or posting an advertisement to CNN.

For example, consider the CSANS user who is researching "Airplane Safety" and finds a website on the "Pan AM Flight 100 disaster" and finds it pertinent to the research task. Now suppose a student who has Cox as their ISP finds the same website for a high school research paper that they are doing. From an intruders point of view, it would be difficult to determine whether either or both the CSANS users and the high school student were attempting to hide their identity given only the traffic obtained from the ISP.

In the external website (service) case, the anonymized traffic blends less when compared to that of the ISP traffic. When someone goes to an external website such as Facebook, Google Mail (Gmail), or Amazon, they give away some of their identity by virtue of logging into a site, accessing personal items, and cookies (e.g. Amazon book preferences). If this traffic is obtained by an intruder, the success rate of determining the identity is increased when compared to simply obtaining the ISP traffic.

For example, consider a CSANS user who accesses their personal Facebook account. The CSANS user must log into Facebook, perhaps views photos, sends a message via Facebook, and posts a few messages. There is something inherently personal about this example. An intruder who is able to acquire the traffic, can often determine the true identity of the user as we will show in the results section of this paper.

In the final case, as was described earlier, we created a custom website that is meant to entice the CSANS users. The website content is associated with a list of keywords that a

typical CSANS researcher would use as their search query. Included in this website are a series of files that serve different purposes. The files have the ability to track the users and to also infect the users. The files, if downloaded (and transferred) to their client machine, will potentially give the intruder the ability to discern the user's true identity. It is for this reason that the success rate of the intruder is the highest. We will show the success rate of this novel idea in the results section. The complexity of obtaining traffic from a custom website is also low because there is no dependence on the ISP or popular services.

To substantiate our claims, we also attempted to gather and analyze the traffic that was generated by the researchers while on CSANS. We ran the Wireshark [23] packet analyzer to accomplish this. The generated Wireshark .pcap was then run against a tool called tcpxtract and used to carve headers and footers from the traffic [24]. This extraction yielded several web images, cookies, files, and portions of web activity.

Next, by using a custom-developed Perl script, we were able to take the destination IP address that was captured, and in most cases, identify the hostname, internet registries (e.g. APNIC, ARIN, RIPE), autonomous system (AS) number, the country where the servers are located, the routing information, and the date the website was registered.

We then took all of this information and created a simple Microsoft Access database so that we could perform queries on the data that we found. The database included over 10,000 records as a result of running the experiment and capturing the traffic. The following section describes our findings.

V. EXPERIMENTAL RESULTS

Based on an analysis of the traffic that was generated by the CSANS researchers, we were able to identify a variety of data points that could have been used to identify the CSANS and its users. First, we saw several connections being made from the real (unanonymized) IP address of the CSANS. The source of the IP address was likely a result of the researchers utilizing remote desktop (RDP) from their client (unanonymized) system to access CSANS.

Furthermore, we identified almost daily requests to Facebook. This was a concern because of the obvious attributable nature of this social networking website. We then began to analyze traffic, cookies, and ultimately uncovered two distinct email addresses via the Firefox profiles. Note: It was necessary for us to download SQLlite Database Browser [25] in order to extract this information from Firefox. Going one step further, we logged on to Facebook and did a "Friend Finder" search, based on the two emails that we uncovered. The results of our search yielded the names of two different people who were later shown to have been part of the research group at CSANS.

Next, we discuss the concept of Facebook profile ID. For example, when you login to Facebook (assuming you have not replaced ID with username) you will see something similar to the following whereby the ## correlate to real numbers:

http://www.facebook.com/profile.php?id=604###

During our traffic capture, we were able to identify two distinct Facebook profile ID's. However, when we accessed the profiles, we noticed that it did not relate to any CSANS users. In fact, they were the profiles of two different men living in a foreign country who were not associated with the CSANS organization whatsoever. We still need to perform future research to understand our findings with respect to profile ID. However, it seems to indicate that the CSANS researchers were searching for these two individuals as part of their research task. This has not yet been confirmed or denied. For more information on Facebook ID, the reader is urged to view [26].

The traffic analysis also uncovered the researchers accessing Amazon, Continental Airlines, and several other U.S. based companies. Furthermore, there were other attributable websites such as LinkedIn and Twitter that were being regularly accessed by the CSANS researchers.

By using the tcpxtract tool, we were able to uncover portions of email that were created from Google Mail (Gmail) and Thunderbird. While it was difficult to view the email in its entirety, it appeared to have a subject associated with the Transportation Security Administration (TSA).

By capturing the traffic for such a prolonged amount of time, we were able to find various patterns associated with the time of day users log in to CSANS, the types of sites (both attributable and non-attributable), and the routing of the traffic as it makes its way to these sites. We then compared this traffic to that of non-anonymized traffic acquired from George Mason University (GMU) and found that it was quite easy to tell which traffic patterns were associated with what entity (GMU or CSANS).

In addition to capturing the traffic and utilizing tcpxtract, we also made us of a tool called network miner [27]. This tool allowed us to import a Wireshark .pcap file and mine the traffic. The tool quickly places the traffic into categories such as hosts, frames, files, images, messages, credentials, sessions, DNS, parameters, keywords, clear text, and anomalies. The main results that were acquired from this tool were the search criteria and queries used by the researchers. For example, we found the following search criteria used by the researchers "petrochemical production process," which is an acceptable search query that relates to the given research task. However, we also came across a search query for "Fairfax County School Closings" that is clearly more attributable to the CSANS location.

In the actual fictitious website experiment, we saw the anonymized IP address 207.195.x.x that was identified as Global TAC, LLC. This log entry showed up several times throughout the experiment on any given day. We identified that several of the files from the website were downloaded by this IP address. Shortly thereafter, we could see that our

random page was identified in the log files, but this time with the NAT IP address that was confirmed to be that of the target organization. This was repeatable as we saw new IP addresses being related to the CSANS IP address. Also, most of this activity appeared during 8:00 a.m. EST and 5:00 p.m. EST, the timeframe most U.S. based organizations are open for business.

In our logs, we were able to clearly see the web browser and the operating system of the user who opened a certain page in our website. This information could be extended to other de-anonymization projects such as [17]. The intruder could also use this information to customize the Metasploit exploits.

In addition to the logs supplied by our web server, we also utilized tracking statistics from TraceMyIP [19]. This allowed us to have a separate tracking mechanism in place, and we were able to identify on several occasions that a user was in fact logged into our website. We also noticed that both the CSANS anonymized IP address and the case study agency were logged in at the same time. Since the traffic to our website is light, it became quite clear that there was some relationship between the two sets of IP addresses that are simultaneously logged in. This was further supported by the fact that both users logged out of our website at nearly identical times.

Within the logs, we were able to determine whether or not we have seen this person before and in some cases determine the first time they visited and the last time they visited. In addition, we were able to identify the system hardware used and the browser, browser language, operating system, and screen resolution. Next, we could uncover the ISP and NAT IP address of the CSANS organization. Another piece of data that we saw in the logs was the hostname of the CSANS. This hostname appeared to be fictitious as it was the name of a children's game. This information was then looked up via nslookup and came back with an IP address that is tied to the CSANS web proxy. Finally, we identified the connecting city, state, country, time zone, and latitude/longitude.

One particular user was paying close enough attention and identified a suspicious "tracking code icon" that was intentionally displayed on the site. However, as far as we can tell, this was the only time that anyone from the CSANS identified the website or the experiment as suspicious.

Also, we were able to see that on certain days (due to what the IP address was on that given day) that Google Location was able to identify the location of the CSANS organization within about 20 miles. However, on other days, the Google Locator was not able to identify the CSANS location. These results show that at least one CSANS IP address was compromised. This was likely due to attributable search queries being performed. It is not immediately clear whether or not it was a CSANS user who was responsible for the attributable search queries.

Lastly, and most damaging, one user downloaded a malicious file, transferred it to their vulnerable Windows XP computer, and opened it in a vulnerable version of Adobe Reader. By utilizing the Metasploit multi/handler listener, we were able to insert the Metasploit payload and open a shell to the CSANS users system. At this point, we could easily have dropped a keylogger, taken a screen capture of the page, run system commands, and quite easily harvested information that would result in determining the user's true identity.

VI. DISCUSSION AND PROPOSED DEFENSES

The results of our traffic analysis and experiment were conclusive in determining a few unique CSANS users. In general, the results of the experiment showed that the anonymized IP address could be directly correlated to the CSANS organization. This was shown by looking at the logs and cross referencing a series of random IP addresses with the CSANS organization's NAT address. This information was most likely due to the user downloading the file, transferring it to their computer, opening the file, and allowing JavaScript to run thus causing a random page within the intruder's webpage to launch. In the log files, it was easy to see that a given page of our website was accessed.

Specifically, we were able to conclude that users who transferred the malicious document to their client operating system were at greater risk. This is because the client operating systems and applications were not nearly as secure as the CSANS. In some cases, the client operating system was unpatched and utilizing an obsolete version of Adobe Reader.

Furthermore, we were able to conclude to some degree that attributable searching on CSANS did play a role in determining the location as shown by Google Location. It is not known at this time what the specific search criteria used was, nor is it known whether or not it was a CSANS user who performed the attributable search. We were able to show that given a clean CSANS virtual machine snapshot, certain IP addresses used in the CSANS IP rotation scheme were identifiable as coming from a location in close proximity.

Overall, we were able to show that we can break the CSANS anonymization. Specifically we relied on targeted attacks that focused on the weaknesses of the users and applications. Based on the results and the conclusions, we recommend several possible defenses that the organization, the CSANS, and the users could have implemented to prevent the attacks.

One possible defense would be to reconsider the use of the transfer file system. Even though the transfer file system is a convenient way to move files, it is one of the weakest links in the CSANS security architecture. At a minimum, it is recommended that the users create new and clean PDF files from the ones that they download. If the CSANS organization decided to keep the transfer drive, it is highly recommended that the files be run through multiple anti-virus engines and/or a sandboxed system that could execute the file, trigger on illegal system operations, and therefore potentially block

the exploited file from connecting back to the intruder's listener server.

The transfer drive system can be extended with a secure and dedicated analysis platform. An improvement to the system architecture might allow the transfer of files to another virtual machine that would allow for execution but prevent information flows to untrusted hosts. This scheme, similar to what has been done with honeypots, would address the vulnerabilities in the current CSANS, while at the same time, allowing for richer analyses to be performed.

One simpler defense is to patch and harden the client machines. However, we believe that the intruders will continue to create exploits and eventually will be able to infect the client machines. This defense will reduce but not eliminate the risk.

The users should be cautious with allowing JavaScript to run on their computers. Options such as noscript from Firefox [16] will automatically prevent JavaScript from running. The users should have immediately contacted the organization's information security group when their browser started to open new browser windows.

To defend against attributable searching while on the CSANS, the users need to be cognizant that logging into personal accounts such as email, social networking, and banking sites could provide information that could discern their identity. Next, the users are advised not to perform any searches associated with their true geographic location. For example, if the user is located in Denver, CO, they should refrain from searching for items such as Denver sports teams, weather, traffic, jobs, and restaurants. This information can be combined to yield the users identity.

As we demonstrated in section III, Google is already proving this by virtue of their Google service location feature that is tied to a specific IP address regardless of whether that IP address is anonymized [18]. For this reason, we have recommended that the CSANS users complete all of their searches using www.google.com.gh whereby the "GH" is the country code for Ghana. We chose Ghana primarily because it is an English speaking country that is often attributed to the IP anonymization scheme that is used at the CSANS.

Still another possible defense would be to set the Google Location to be a random place in the country. This concept is commonly referred to as "artificial pinning" and is just another small measure that could potentially help to prevent the identity of that particular IP address from being known. At the time of this writing, it was not possible to change the location from outside of the United States assuming that you were using www.google.com (.en) as your main search engine. Also, in some cases, it is not even possible to change this setting and/or it may not appear for a plethora of reasons that are outside the scope of this paper.

Similarly, the users should not provide any personal information to a website, toolbar, or widget about their true location when logged on to CSANS. For example, users often set weather monitoring tools to be their local zip code. The CSANS developers should create visual guidance when a user is using the non-attributable CSANS to search for information related to their true location. Perhaps a flashing notification or a confirmation screen before a given search is executed would be useful.

The overall limitations outlined for privacy-preserving systems are well known, to the point that there are tools such as Privoxy that are designed to supplement the protection offered by ANS. Specifically, Privoxy [33] is a non-caching web proxy with advanced filtering capabilities for enhancing privacy, modifying web page data HTTP, and controlling access. Since it has an application for both stand-alone systems and multi-user networks, one suggestion would be to integrate Privoxy on-top of CSANS.

Another potential solution is to allow regular, non-anonymized users to share the IP address space of the anonymized users. This will create a large quantity of search results that could allow the anonymized users to potentially hide behind an overwhelming amount of search data. For example, if instead of using the CSANS, the organization tunneled a portion of their traffic to universities and colleges in the same area it might actually improve the anonymization. This is because a potential intruder would find it difficult to determine the source of the traffic and whether it was anonymous. Since the research being performed by the CSANS organization might be similar to that of a university professor, it would be nearly impossible to determine which one is anonymous.

VII. CONCLUSIONS & FUTURE WORK

We conclude this paper by looking at future research. This experiment was done on a small scope with little funding available to the authors. Future research could extend this experiment and consider different anonymization schemes and different applications. We leave it to future research to determine if attacks against other ANS users would be successful. Also, this experiment was narrow with a focus on the commercial airline industry. It would be interesting to determine if a similar attack would work in other organizations such as healthcare or banking. Future research could provide methods to improve the defenses offered in this paper and more importantly help educate the users on how to best defend themselves from the three main attack methodologies.

While not a revelation on its own, we once more show that users and their activities remain the weakest link in any non-trivial security scheme. The implication, however, is the researchers were careless or ignorant of basic security principles. The fact that they betrayed personal information so easily does not reflect well on the level of security education provided to the researchers.

The main takeaway from this study was that both the administrators and CSANS users were not educated about

quite prevalent attack vectors for compromising client systems and violating user privacy.

To address this problem, we briefed the CSANS researchers on the results of our findings. We also explained to them the dangers of accessing websites and giving away PII while on the CSANS. The training that we performed was over a two-day span and in it we discussed our findings along with our recommendations as to how to resolve the problem. We are planning on conducting a similar experiment in the upcoming months in order to measure the effectiveness of our training with regards to combating the attacks mentioned in this paper.

ACKNOWLEDGMENT

REFERENCES

[1] Anonymizer [Online]. Available: http://www.anonymizer.com
[2] Tor [Online]. Available: http://www.torproject.org
[3] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudo-nyms," *Communications of the ACM*, vol. 4, no. 2, February 1981.
[4] S. Coull, C. Wright, F. Monrose, M. Collins, and M. K.Reiter. "Playing devil's advocate: Inferring sensitive information from anonymized network traces," In *Proceedings of the Network and Distributed System Security Symposium (NDSS),* San Diego, CA, 2007.
[5] H. D. Moore and V. Smith, "Tactical Exploitation" [Course Slides], *Black Hat USA 2010*, Las Vegas, NV, July 2010.
[6] N. Schear and D. M. Nicol, "Performance analysis of real traffic carried with encrypted cover flows," In *Proceedings of the 22nd Workshop on Principles of Advanced and Distributed Simulation (PADS)*, Washington, DC, USA: IEEE Computer Society, 2008, pp. 80–87.
[7] F. Swiderski, and W. Snyder, *Threat Modeling*. Redmond, WA: Microsoft Press, 2004.
[8] Metasploit [Online]. Available: http://www.metasploit.com
[9] X. Wang, S. Chen, and S. Jajodia, "Network flow watermarking attack on low-latency anonymous communication systems," in *Proceedings of the IEEE Symposium on Security & Privacy (S&P)*, Washington, DC, USA: IEEE Computer Society, May 2007, pp. 116-130.
[10] S. Coull, M.P. Collins, C.V. Wright, F. Monrose, and M. Reiter, "On web browsing privacy in anonymized netflows," In *Proceedings of the 16th USENIX Security Symposium*, Boston, MA, August 2007, pp. 339-352.
[11] N. Hopper, E. Y. Vasserman, and E. Chan-Tin, "How much anonymity does network latency leak?" in *Proceedings of ACM CCS*, Alexandria, VA, USA, 2007, pp. 13:1 – 13:28.
[12] M. Casado, and M. J. Freedman, "Peering through the shroud: The effect of edge opacity on IP-based client identification. In Proceedings of the 4th USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI '07), Cambridge, MA, USA, 2007, pp. 173-186.
[13] S. Coull, C. Wright, A. Keromytis, F. Monrose, and M. Reiter, "Taming the devil: Techniques for evaluating anonymized network data," In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS '08)*, San Diego, CA, USA, February 2008.
[14] J. Jung, A. Sheth, B. Greenstein, D. Wetherall, G. Maganis, and T. Kohno, "Privacy Oracle: A System for Finding Application Leaks with Black Box Differential Testing," In *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS '08)*, 2008, pp. 279-288.
[15] D. Scott and R. Sharp, "Abstracting application layer web security," In *Proceedings of the Electrical International World Wide Web Conference (WWW 2004)*, Honolulu, HI, May 2002.
[16] NoScript [Online]. Available:https://addons.mozilla.org/en-US/firefox/addon/722/
[17] Panopticlick [Online]. Available: https://panopticlick.eff.org/
[18] Google Location [Online]. Available: http://www.google.com/support/websearch/bin/answer.py?answer=179386&hl=en
[19] Trace My IP [Online]. Available: http://www.tracemyip.org
[20] Browzar [Online]. Available: http://www.browzar.com
[21] JAP [Online]. Available: http://anon.inf.tu-dresden.de/index_en.html
[22] Safesurf [Online]. Available: http://www.safersurf.com
[23] Wireshark [Online]. Available: http://www.wireshark.org
[24] Tcpxtract [Online]. Available: http://tcpxtract.sourceforge.net
[25] SQLite [Online]. Available: sqlitebrowser.sourceforge.net
[26] Facebook Profile [Online]. Available: http://www.ehow.com/how_5753004_facebook-id.html
[27] Network Miner [Online]. Available: http://networkminer.sourceforge.net
[28] R. Pang, M. Allman, V. Paxson, and J. Lee, "The devil and packet trace anonymization," *ACM Computer Communication Review,* vol. 36, no. 1, pp. 29–38, January 2006.
[29] M. Huber, M. Mulazzani, and E. Weippl, "Tor HTTP usage and information leakage," Springer Berlin Heidelberg: *Communications and Multimedia Security*, vol. 6109, pp. 245-255.
[30] S. Kadloor, X. Gong, N. Kiyavash, T. Tezcan, and N.Borizov, "A low-cost side channel traffic analysis attack in packet networks. In *Proceedings of the 2010 IEEE International Conference on Communications (ICC)*, pp.1-5, May 2010.
[31] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, "A practical attack to de-anonymize social network users," In *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, pp. 223-238, May 2010.
[32] A. Chaabane, P. Manils, and M. Kaafar, "Digging into anonymous traffic: A deep analysis of the tor anonymizing network," In Proceedings 4th International Conference on Network and System Security (NSS), Melbourne, VIC, pp. 167 –174, 2010.
[33] Privoxy [Online]. Available: http://www.privoxy.org
[34] T.G. Abbott, K.J. Lai, M.R. Lieberman, and E.C. Price, "Browser-based attacks on Tor," *Lecture Notes in Computer Science*, vol. 4776/2007, pp. 184-199, 2007.