International Journal on Applications in Information and Communication Engineering
Volume 5 : Issue 2: November 2019, pp 1 – 5 www.aetsjournal.com ISSN (Online) : 2394 - 6237

EFFECTIVE AND EFFICIENT TRAFFIC SCRUTINY IN SWEET SERVER WITH DATA PRIVACY

Dr. E. PUNARSELVAM , S. GOPI

Abstract— Data communications over the Internet face serious threats to countries with repressive regimes, leading them to develop and deploy censorship mechanisms within their networks. Unfortunately, existing censorship systems do not provide high availability guarantee to their users, as censors can easily identify, hence disrupt, the traffic belonging to these systems using today's advanced censorship technologies. In this paper, we propose serving the Web by Exploiting Email Tunnels (SWEET), a highly available censorship-resistant infrastructure. SWEET works by encapsulating a censored user's traffic inside email messages that are carried over public email services like Gmail and Yahoo Mail. As the operation of SWEET is not bound to any specific email provider, we argue that a censor will need to block email communications all together in order to disrupt SWEET, which is unlikely as email constitutes an important part of today's Internet. Through experiments with a prototype of our system, we find that SWEET's performance is sufficient for Web browsing. In particular, regular Websites are downloaded within couple of seconds.

Index terms: Censorship circumvention, email communications, traffic encapsulation.

I.INTRODUCTION

The Internet provides users from around the world with an environment to freely communicate, exchange ideas and information. However, free communication continues to threaten repressive regimes, as the open circulation of information and speech among their citizens can pose serious threats to their existence. Recent unrest in the middle east demonstrates that the Internet can be widely used by citizens under these regimes as a very powerful tool to spread censored news and information, inspire dissent, and organize events and protests. As a result, repressive regimes extensively monitor their citizens' access to the Internet and restrict open access to public networks by using different technologies, ranging from simple IP address blocking and DNS hijacking to the more complicated and resource-intensive Deep Packet Inspection (DPI).

Dr. E. Punarselvam , Professor , Department of Information Technology, Muthayammal Engineering College.

S. Gopi , Assistant Professor , Department of Information Technology, Muthayammal Engineering College.

II. RELATED WORKS

1) A Taxonomy of Internet Censorship and Anti-Censorship

Internet is supposed to be born free, yet it is censored almost everywhere, and severely censored in a few countries. The tug-of-war on the Internet between censors and anti-censors is intensifying. This survey presents taxonomy on the principles, techniques, and technologies of Internet censorship and anti-censorship. It highlights the challenges and opportunities in anti-censorship research, and outlines a historical account via the lenses of news coverage in the past decade. The main two objectives of this research are to provide an overview of online censorship including the interplay between technology and policy, and present a taxonomy and set of key design factors for anti - censorship technologies. With respect to the first objective, a historical review of information control including the evolution into online censorship and the corresponding commercial applications demonstrates how censorship has been practiced and employed throughout the world as a social and political tool to control free thought and expression.

2) Protecting free expression online with freenet

Freenet is a distributed information storage system designed to address information privacy and survivability concerns. Freenet operates as a self-organizing P2P network that pools unused disk space across potentially hundreds of thousands of desktop computers to create a collaborative virtual file system. Freenet employs a completely decentralized architecture. Given that the P2P environment is inherently untrustworthy and

unreliable, we must assume that participants could operate maliciously or fail without warning at any time. Therefore, Freenet implements strategies to protect data integrity and prevent privacy leaks in the former instance, and provide for graceful degradation and redundant data availability in the latter. The system is also designed to adapt to usage patterns, automatically replicating and deleting files to make the most effective use of available storage in response to demand.

3) Infranet: Circumventing Web censorship and surveillance

An increasing number of countries and companies routinely block or monitor access to parts of the Internet. To counteract these measures, we propose Infranet, a system that enables clients to surreptitiously retrieve sensitive content via cooperating Web servers distributed across the global Internet. These Infranet servers provide clients access to censored sites while continuing to host normal uncensored content. Infranet uses a tunnel protocol that provides a covert communication channel between its clients and servers, modulated over standard HTTP transactions that resemble innocuous Web browsing. In the upstream direction, Infranet clients send covert messages to Infranet servers by associating meaning to the sequence of HTTP requests being made. In the downstream direction, Infranet servers return content by hiding censored data in uncensored images using steganographic techniques. We describe the design, a prototype implementation, security properties, and performance of Infranet. Our security analysis shows that Infranet can successfully circumvent several sophisticated censoring techniques.

III.PROPOSED SYSTEM

SWEET works by encapsulating a censored user's traffic inside email messages that are carried over public email services like Gmail and Yahoo Mail. As the operation of SWEET is not bound to any specific email provider, we argue that a censor will need to block email communications all together in order to disrupt SWEET, which is unlikely as email constitutes an important part of today's Internet.

SWEET client, confined by a censoring ISP, tunnels its network traffic inside a series of email messages that are exchanged between herself and an email server operated by SWEET's server. The SWEET server acts as an Internet proxy by proxying the encapsulated Propose Serving the Web by Exploiting Email Tunnels (SWEET), a highly available censorship-resistant infrastructure. Traffic to the requested blocked destinations. The SWEET client uses an oblivious, public mail provider (e.g., Gmail, Hotmail, etc.) to exchange the encapsulating emails, rendering standard email filtering mechanisms ineffective in identifying/blocking SWEET-related emails. More specifically, to use SWEET for circumvention a client needs to create an email account with some public email provider; she also needs to obtain SWEET's client software from an out-of-bound channel (similar to other circumvention systems).

Before being able to request data from Internet destinations, a user needs to be registered by the SWEET server. A powerful censor can perform traffic analysis to detect the use of SWEET, e.g., by comparing a user's email communications with that of a typical email user. As a result, a SWEET user who is concerned about unobservability needs to ensure that her SWEET email communications mimic that of a normal user (a user who does not fear reprisal from her government might opt to have lower unobservability in order to gain a higher communication bandwidth).

The censors may also try to detect SWEET users by analyzing the inter-arrival times of the email messages exchanged between SWEET users and their mail service providers. More specifically, a SWEET client may send and receive multiple emails in a shorter time interval compared to regular email clients. We, however, argue that this is not a serious vulnerability for SWEET. As discussed earlier, it require SWEET clients to use mail service providers that encrypt email exchanges (otherwise, the censors can simply filter SWEET emails by searching for the email addresses of SWEET servers). The use of encryption prevents the censors from identifying the number of emails exchanged between a SWEET user and her mail service provider.

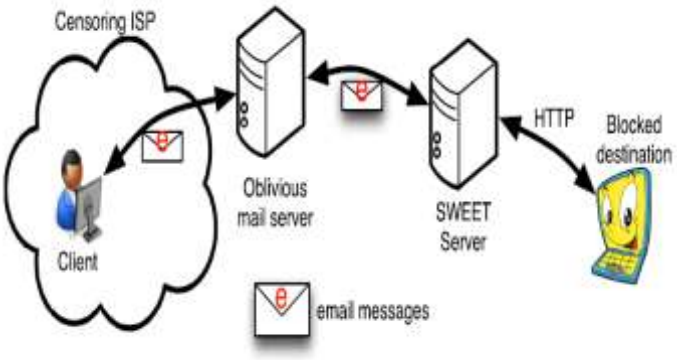

Figure. 1 Overall System Architecture

1) SWEET's UNOBSERVABILITY:

We claim that a censor is not easily able to distinguish between SWEET's email messages and benign email messages. A SWEET client has two options in choosing her email account:

1) AlienMail a non-domestic email that encrypts emails (e.g., Gmail for users in China)

2) DomesticMail a domestic email account with no need for encryption (e.g., 163.com for users in China).

2) SWEET server

The SWEET server is running outside the censoring region. It helps SWEET clients to evade censorship by proxying their traffic to blocked destination. Figure 2 shows the design, composed of four elements.

- Email Agent
- Convertor
- Proxy Agent
- Registration Agent

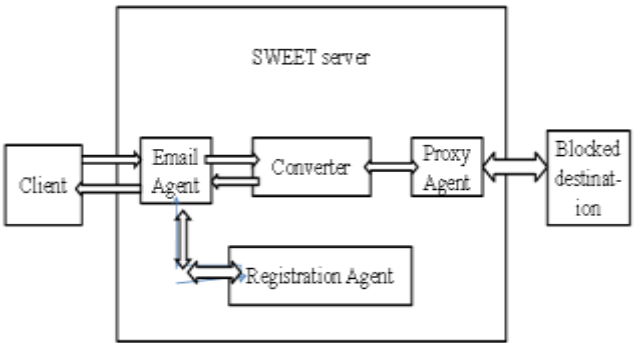

Figure.2 SWEET Server Architecture

A. Email Agent

The email agent is an IMAP and SMTP server that receives emails that contain the Tunneled internet traffic, sent by SWEET clients to SWEET's email address The email Agent passes the received emails to another components of the SWEET server, the convertor and registration agent.

B.Convertor

The Convertor process the email passed by the email agent, and extracts the tunnele Packets. It then, forwards the extracted data to another component, proxy agent. Also the convertor receives the packets from the proxy agent. Convertor encrypts/decrypts the email attachment using a secret key shared with that user.

C. Proxy Agent

The Proxy agent proxies the network packets of client that are extracted by the convertor, and sends them to the internet destination requested by the clients. It also sends packet from the destination back to the convertor.

D. Registration Agent

This component is in charge of registering the email address of the SWEET clients, prior to their use of SWEET. The information about the registered clients can be used to ensure quality of service and prevent denial-of service attacks on the server. It shares the secret key with the client which is used for encrypt the tunneled information between the client and server.

3) SWEET Client

To use SWEET, a client needs to obtain a copy of SWEET's client software and Install it on machine. The client also needs to create one email account, and to configure the SWEET's software with information of email account. The client software register the email address of its user with SWEET server and get a shared secret key.

4) PROTOCOL BASED DESIGN

Figure 3 shows four elements. They are

- Web Browser
- Email Agent
- Convertor

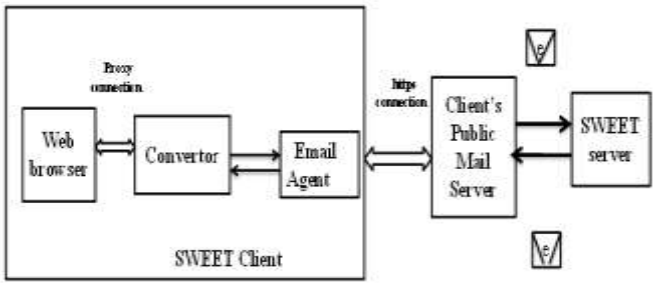

Figure.3 Protocol Based Design

A. Web Browser

The client can use ant web browser that supports proxying of connections,e.g., Google Chrome, Internet Explorer. The client needs to configure the browser to use a local proxy server.

B. Email Agent

It sends and receives SWEET emails through the clients email account. It is configured with setting of the SMTP and IMAP/POP3 servers of the user's email account, as well as the login information.

C. Convertor

It sits between the web browser and email agent and converts, SWEET emails into network packet and vice versa. It uses the keys shared with SWEET.

IV. ALGORITHM

ABE Encryption

It is a traditional public key encryption. It is a powerfull mechanism to ensure confidentiality of data storage and transmission. It can work effectively in the scenario that user who encrypts the sensitive data knows exactly, the identity and public key of the recipient. ABE is realistic for the real environment.

It is a type of public-key encryption in which the secret key of a user and the ciphertext are dependent upon attributes(e.g. the country in which he lives, or the kind of subscription he has). In such a system, the decryption ciphertext is possible only if the set of attributes of the user key matches the attributes of the ciphertext.

A crucial security aspect of attribute based encryption is collusion-resistance.. An adversary that holds multiple keys should only be able to access data if at least one individual key grant access.

There are mainly two types of attribute based encryption schemes,

1.Key-policy attribute based encryption(KP-ABE)

2.Ciphertext-policy attribute based encryption(CP-ABE)

ABE can be used for log encryption. Instead of encrypting each part of log with the keys of all recipients, it is possible to encrypt the log only with attributes which match recipient's attributes.

1) KEY POLICY-ATTRIBUTE BASED ENCRYPTION

In KP-ABE, the owner of the data creates a master key,using master key, the owner encrypts the data such that a cipher text is labeled with a set of attributes. The private key(decryption key) given to others to decrypt the data is associated with a tree-based access structure which specifies which ciphertext the key can decrypt. The tree-based access structure contains leaves which are associated with attributes. A user is able to decrypt a cipher text if the attributes associated with the cipher text satisfies the user's key access structure. In KP-BE each ciphertext is associated with a set of attributes, and each user secret key is associated with access policy on attributes. Decryption is enabled if and ony if the ciphertext attribute set satisfies the access structure of the user secret key.

2) CIPHERTEXT POLICY-ATTRIBUTE BASED ENCRYPLTION

We use CP-ABE Algorithm. It uses access trees to encrypt data and user's secret keys are generated over a set of attributes. Attributes are associated with user secret keys and access structures with ciphertexts.

In CP-ABE system, a users's private key is associated with a set of attributes. When a party encrypts a message in this system, they specify an access structure over attributes and associate his access structure with the cipher text. A user will only be able to decrypt a cipher text if that user's attributes pass through the cipher text's access structure.

i. Setup

ii. Key generation

iii. Encrypt
iv. Decrypt

1) SETUP

Setup(1k)→PK,MK. This algorithm takes the security parameters k as input and generates a public key PK and master key MK.

2) KEY GENERATION

Key Generation(MK,L)→SK. This algorithm takes the master key MK and a set of attributes L as input and outputs the user's private key SK associated with L. In other words, the user's attribute set L describes his /her private key SK and determines his/her right for decryption.

3) ENCRYPTION

Encrypt(PK,M,W)→CT. The user uses public key PK, and ciphertext access policy W to encrypt message M, and outputs the ciphertext CT.Only the user whose attributes meet W can decrypt CT correctly.

4) DECRYPTION

Decrypt(PK,CT,SK)→M. In this algorithm, the user uses pubic key PK, his/her private key SK,and his/her own attributes set L to decrypt CT. He/She can decrypt CT correctly when and only when his/her attributes set L meets the access policy contained in CT.

V. CONCLUSION

The SWEET, a deployable system for unobservable communication with Internet destinations. SWEET works by tunneling network traffic through widely used public email services such as Gmail, Yahoo Mail, and Hotmail. Unlike recently-proposed schemes that require a collection of ISPs to instrument router-level modifications in support of covert communications, our approach can be deployed through a small applet running at the user's end host, and a remote email-based proxy, simplifying deployment. Through an implementation and evaluation in a wide-area deployment, we find that while SWEET incurs some additional latency in communications, these overheads are low enough to be used for interactive accesses to web services. We feel our work may serve to accelerate deployment of censorship-resistant services in the wide area, guaranteeing high availability.

References

[1] D. McCoy, J. A. Morales, and K. Levchenko, "Proximax: A measurement based system for proxies dissemination," Financial Cryptography Data Security., vol. 5, no. 9, pp. 1–10, 2016.

[2] J. McLachlan and N. Hopper, "On the risks of serving whenever you surf: Vulnerabilities in Tor's blocking resistance design," in Proc. 8th ACM Workshop Privacy Electron. Soc., Nov. 2009, pp. 31–40. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1655188.1655193

[3] Y. Sovran, A. Libonati, and J. Li, "Pass it on: Social networks stymie censors," in Proc. 7th Int. Conf. Peer-to-Peer Syst., Feb. 2012, p. 3. [Online]. Available: http://www.iptps.org/papers-2012/73.pdf

[4] (Nov. 2007). Defeat Internet Censorship: Overview of Advanced Technologies and Products. [Online]. Available: http:// www.internetfreedom.org/archive/DefeatInternet Censorship White Paper.pd

[5] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second generation onion router," in Proc. USENIX Secur. Symp., 2018, pp. 21–37.

[6] N. Feamster, M. Balazinska, W. Wang, H. Balakrishnan, and D. Karger, "Thwarting Web censorship with untrusted messenger discovery," in Int. Workshop Privacy Enhancing Technol., 2018, pp. 125–140.

[7] I. Clarke, S. G. Miller, T. W. Hong, O. Sandberg, and B. Wiley, "Protecting free expression online with freenet," IEEE Internet Comput., vol. 6, no. 1, pp. 40–49, Jan. 2015.