IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

EFFECTIVE DETECTION OF MULTIMEDIA PROTOCOL BY EXPLOITING EMAIL TUNNELS THROUGH SWEET SERVING

[1]Veema Rao.Alluri
Pg Scholor
Narasaraopeta Institute of Technology, Narasaraopet

[2]Suresh.Munnangi
Assoc Professor
Narasaraopeta Institute of Technology, Narasaraopet

ABSTRACT: Open communications over the Internet pose serious threats to countries with repressive regimes, leading them to develop and deploy censorship mechanisms within their networks. Unfortunately, existing censorship circumvention systems do not provide high *availability* guarantees 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.

1.INTRODUCTION

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 openaccess 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) .With the use of censorship technologies, a number of different systems were developed to retain the openness of the Internet for the users living under repressive regimes. The earliest circumvention tools are HTTP proxies that simply intercept and manipulate a client's HTTP requests, defeating IP address blocking and DNS hijacking techniques. The use of more

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

advanced censorship technologies such as DPI, rendered the use of HTTP proxies ineffective for circumvention. This led to the advent of more advanced tools such as Ultrasurf and Psiphon, designed to evade content filtering. While these circumvention tools have helped, they face several challenges.

We believe that the biggest one is their lack of *availability*, meaning that a censor can disrupt their service frequently or even disable them completely. The common reason is that the network traffic made by these systems can be distinguished from regular Internet traffic by censors, i.e.,such systems are not *unobservable*. For example, the popular Tor network works by having users connect to an ensemble of nodes with public IP addresses, which proxy users' traffic to the requested, censored destinations. This public knowledge about Tor's IP addresses, which is required to make Tor usable by users globally, can be and is being used by censors to block their citizens from accessing Tor. To improve availability, recent proposals for circumvention aim to make their traffic unobservable to the censors by pre-sharing secrets with their clients. Others suggest to conceal circumvention by making infrastructure modifications to the Internet.

2.RELATED WORK

The second generation onion router by R. Dingledine, N. Mathewson. We present Tor, a circuit-based low-latency anonymous communication service. This second-generation Onion Routing system addresses limitations in the original design by adding perfect forward secrecy, congestion control, directory servers, integrity checking, configurable exit policies, and a practical design for location-hidden services via rendezvous points. Tor works on the real-world Internet, requires no special privileges or kernel modifications, requires little synchronization or coordination between nodes, and provides a reasonable tradeoff between anonymity, usability, and efficiency. We briefly describe our experiences with an international network of more than 30 nodes. We close with a list of open problems in anonymous communication.

Ignoring the great firewall of China by R. Clayton, S. J. Murdoch The so-called "Great Firewall of China" operates, in part, by inspecting TCP packets for keywords that are to be blocked. If the keyword is present, TCP reset packets (viz: with the RST flag set) are sent to both endpoints of the connection, which then close. However, because the original packets are passed through the firewall unscathed, if the endpoints completely ignore the firewall's resets, then the connection will proceed unhindered. Once one connection has been blocked, the firewall makes further easy-to-evade attempts to block further connections from the same machine. This latter behaviour can be leveraged into a denial-of-service attack on third-party machines.

3 EXISTING SYSTEM:

Tor network works by having users connect to an ensemble of nodes with public IP addresses, which proxy users' traffic to the requested, censored destinations. This public knowledge about Tor's IP addresses, which is required to make Tor usable by users globally, can be and is being used by censors to block their citizens from accessing Tor. To improve availability,

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

recent proposals for circumvention aim to make their traffic unobservable to the censors by pre-sharing secrets with their clients.Telex and Cirripede provide this unobservable communication without the need for some pre-shared secret information with the client, as the secret keys are also covertly communicated inside the network traffic. Cirripede uses an additional client registration stage that provides some advantages and limitations as compared to Telex and Decoy routing systems.

Lack of *availability*, meaning that a censor can disrupt their service frequently or even disable them completely.It has recently been shown that these systems' unobservability is breakable; this is because a comprehensive imitation of today's complex protocols is sophisticated and infeasible in many cases

4 PROPOSED SYSTEM:

In this paper, we design and implement SWEET, a censorship circumvention system that provides high availability by leveraging the openness of email communications.This paper makes the following main contributions: i) we propose a novel infrastructure for censorship circumvention, SWEET, which provides high availability, a feature missing in existing circumvention systems; ii) we develop two prototype implementations for SWEET (one using webmail and the other using email exchange protocols) that allow the use of nearly all email providers by SWEET clients; and, iii) we show the feasibility of SWEET for practical censorship circumvention by measuring the communication latency of SWEET for web browsing using our prototype implementation.

The SWEET server acts as an Internet proxy by proxying the encapsulated traffic to the requested blocked destinations.Our approach can be deployed through a small applet running at the user's end host, and a remote email-based proxy, simplifying deployment

5 IMPLEMENTATION

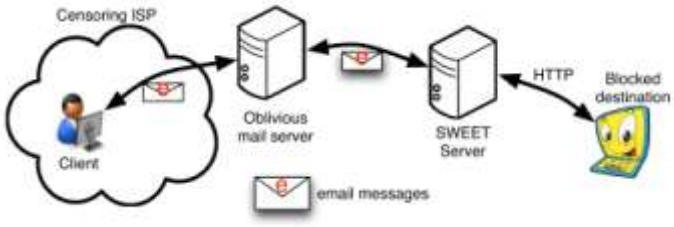


Fig 1 SYSTEM ARCHITECTURE

SWEET server:

In this module, the SWEET server is the part of SWEET running outsidethe censoring region. It helps SWEET clients to evadecensorship by proxying their traffic to blocked destinations.More specifically, a SWEET server communicates withcensored users by exchanging emails that carry tunnelednetwork packets. The main design of SWEET server, which is composed of the following elements:

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 component of the SWEET server, the converter and the registration agent. The email agent also sends emails to SWEET clients, which are generated by other components of SWEET server and contain tunneled network packets or client registration information.

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

Converter:The converter processes the emails passed by the email agent, and extracts the tunneled network packets. It then forwards the extracted data to another component, the proxy agent. Also, the converter receives network packets from the proxy agent and converts them into emails that are targeted to the email address of corresponding clients. The converter then passes these emails to the email agent for delivery to their intended recipients. As described later, the converter encrypts/decrypts the email attachments of a user using a secret key shared with that user.

Proxy agent:The proxy agent proxies the network packets of clients that are extracted by the converter, and sends them to the Internet destination requested by the clients. It also sends packets from the destination back to the converter.

Registration agent:This component is in charge of registering the email addresses 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 to prevent denial-of-service attacks on the server. Additionally, the registration agent shares a secret key with the client, which is used to encrypt the tunneled information between the client and the server.

Client registration:Before the very first use of the SWEET service, a client needs to register her email address with the system. This is automatically performed by the client's SWEET software. The objective of client registration is twofold: to prevent denial-of-service (DoS) attacks and to share a secret key between a client and the server.

SWEET Client:

In this module, a client needs to obtain a copy of SWEET'sclient software and install it on her machine. The client alsoneeds to create one or two email account (depending on if sheuses an AlienMail or a DomesticMail for her primary email).A client needs to configure the installed SWEET's softwarewith information about her email account. Prior to the firstuse of SWEET by a client, the client software registers theemail address of its user with the SWEET server and obtainsa shared secret key.

Web Browser:The client can use any web browser that supports proxying of connections, e.g., Google Chrome, Internet Explorer, or Mozilla Firefox. The client needs to configure her browser to use a local proxy server, e.g., by setting localhost:4444 as the HTTP/SOCKS proxy. The client can use two different browsers for browsing with and without SWEET to avoid the need for frequent re-configurations of the browser. Alternatively, some browsers (e.g., Chrome, and Mozilla Firefox) allow a user to have multiple browsing *profile*s, hence, a user can setup two profiles for browsing with and without SWEET.

Email Agent:It sends and receives SWEET emails thorough the client's email account. The client needs to configure it with the settings of the SMTP and IMAP/POP3 servers of her email account. The client also needs to provide it with the account login information.

Converter:It sits between the web browser and the email agent, and converts SWEET emails into network packets and vice versa.

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

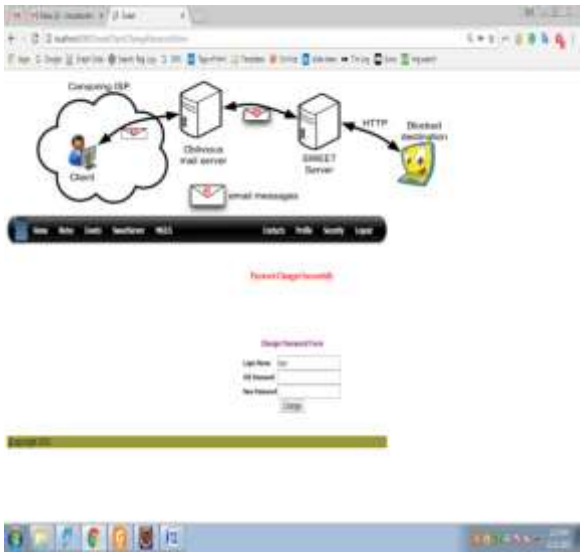
It uses the keys shared with SWEET, *kC,R*, to encrypt/decrypt email content.

The email agent uses its web browser to open a webmail interface with the client's email account, using the user's authentication credentials for logging in. Through this HTTP/HTTPS connection, the email agent communicates with the SWEET server by sending and receiving emails.

Proxy Protocol:

In this module, the SWEET server uses a proxy agent to receive the tunneled traffic of clients and to establish connections to the requested destinations. We consider the use of both SOCKS and HTTP proxies in the design, as each provides unique advantages. Our server's proxy agent runs a SOCKS proxy and an HTTP proxy in parallel, each on a different port. A user can choose to use the type of proxy by configuring her client to connect to the corresponding port. The use of the SOCKS proxy allows the client to make *any* IP connection through the SWEET system, including dynamic web communications, such as Javascript or AJAX, and instant messaging. In contrast, an HTTP proxy only allows access to HTTP destinations. However, an HTTP proxy may speed up connections by using HTTP-layer optimizations such as caching or pre-fetching of web objects.

6. RESULTS



7.CONCLUSION

In this paper, we presented SWEET, a deployable systemfor unobservable communication with Internet destinations.SWEET works by tunneling network traffic through widelyusedpublic email services such as Gmail, Yahoo Mail, andHotmail. Unlike recently-proposed schemes that require a collectionof ISPs to instrument router-level modifications in supportof covert communications, our approach can be deployedthrough a small applet running at the user's end host, and aremote email-based proxy, simplifying deployment. Throughan implementation and evaluation in a wide-area deployment,we find that while SWEET incurs some additional latency incommunications, these overheads are low enough to be usedfor interactive accesses to web services. We feel our work mayserve to accelerate deployment of censorship-resistant servicesin the wide area, guaranteeing high availability.

REFERENCES

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

[1] J. Zittrain and B. Edelman, "Internet filtering in China," *IEEE InternetComput.*, vol. 7, no. 2, pp. 70–77, Mar. 2003.
[2] (Nov. 2007). *Defeat Internet Censorship: Overview ofAdvanced Technologies and Products*. [Online]. Available: http://www.internetfreedom.org/archive/DefeatInternet Censorship WhitePaper.pd
[3] C. S. Leberknight, M. Chiang, H. V. Poor, and F. Wong.(2010).*A Taxonomy of Internet Censorship and Anti-Censorship*.[Online].Available: http://www.princeton.edu/ chiangm/anticensorship.pdf
[4] 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. 2002.
[5] *Ultrasurf*, accessed on Jan. 7, 2017. [Online]. Available:https://ultrasurf.us/
[6] J. Jia and P. Smith. (2004). *Psiphon: Analysis and Estimation*.[Online]. Available: http://www.cdf.toronto.edu/ csc494h/reports/2004-fall/psiphon_ae.html
[7] I. Cooper and J. Dilley, "Known HTTP proxy/caching problems," IETF,Fremont, CA, USA, Tech. Rep. Internet RFC 3143, Jun. 2001.
[8] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The secondgenerationonion router," in *Proc. USENIX Secur. Symp.*, 2004,pp. 21–37.
[9] J. Boyan, "The anonymizer: Protecting user privacy on the Web,"*Comput.-Mediated Commun. Mag.*, vol. 4, no. 9, pp. 1–6, Sep. 1997.
[10] *DynaWeb*, accessed on Jan. 7, 2017. [Online]. Available:http://www.dongtaiwang.com/home_en.php
[11] R. Clayton, S. J. Murdoch, and R. N. M. Watson, "Ignoring the greatfirewall of China," in *Proc. Int. Workshop Privacy Enhancing Technol.*,2006, pp. 20–35.
[12] Y. Sovran, A. Libonati, and J. Li, "Pass it on: Social networks stymiecensors," in *Proc. 7th Int. Conf. Peer-to-Peer Syst.*, Feb. 2008, p. 3.[Online]. Available: http://www.iptps.org/papers-2008/73.pdf
[13] D. McCoy, J. A. Morales, and K. Levchenko, "Proximax: A measurementbased system for proxies dissemination," *Financial Cryptogr. DataSecur.*, vol. 5, no. 9, pp. 1–10, 2011.
[14] N. Feamster, M. Balazinska, W. Wang, H. Balakrishnan, and D. Karger,"Thwarting Web censorship with untrusted messenger discovery," in *Int.Workshop Privacy Enhancing Technol.*, 2003, pp. 125–140.
[15] M. Mahdian, "Fighting censorship with algorithms," in *Proc. Int.Conf. Fun Algorithms*, 2010, pp. 296–306. [Online]. Available:http://dx.doi.org/10.1007/978-3-642-13122-6_29
[16] J. McLachlan and N. Hopper, "On the risks of serving whenever yousurf: 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
[17] P. Winter and S. Lindskog. Apr. 2012. "How China is blocking Tor."[Online]. Available: https://arxiv.org/abs/1204.0447
[18] (Sep. 2007). *Tor Partially Blocked in China*. [Online]. Available:https://blog.torproject.org/blog/tor-partially-blocked-china
[19] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, andD. Karger, "Infranet:

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

Circumventing Web censorship and surveillance,"in *Proc. 11th USENIX Secur. Symp.*, Aug. 2002, pp. 247–262.[Online]. Available: http://www.usenix.org/events/sec02/feamster.html

[20] S. Burnett, N. Feamster, and S. Vempala, "Chipping awayat censorship firewalls with user-generated content," in *Proc.USENIX Secur. Symp.*, 2010, pp. 463–468.