Sweet Serving the Web by Exploiting Email Tunnels

Nali Raghavulu[1], M.Naresh[2],
[1]*PursuingM.Tech (software engineering), Newton's Institute of Engineering College, Alugurajupally, macherla, Guntur dist, AP, India*
[2]*Associate Professor, Newton's Institute of Engineering College, Alugurajupally, macherla, Guntur dist, AP, India*

Abstract- **Open communication over the Internet poses a serious threat to countries with repressive regimes, leading them to develop and de- ploy censorship mechanisms within their networks. Unfortunately, ex- isting 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 censor- ship technologies. In this paper we propose SWEET, a highly available censorship-resistant infrastructure. SWEET works by encapsulating a censored user's traffic inside email messages that are carried over by typ- ical email service providers, 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 all email communications in order to disrupt SWEET, which is unlikely as email constitutes an important part of to- day's Internet. Through experiments with a prototype of our system we find that SWEET's performance is sufficient for web traffic. In particular, regular websites are downloaded within couple of second.**

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. As a result, repressive regimes extensively monitor their citizens' access to the Internet and restrict open access to public networks [37] by using different tech- nologies, ranging from simple IP address blocking and DNS hijacking to the more complicated and resource-intensive Deep Packet Inspection (DPI) [3, 22]. 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 repres- sive regimes [2,5,9,10,12,19]. 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 [14,24,26,27,31]. 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. To improve availability, recent proposals for circumvention aim to make their traffic unobservable to the censors by pre-sharing secrets with their clients [6, 11, 13]. Others [16, 18, 21, 36] suggest to conceal circumvention by making infrastructure modifications to the Internet. Nevertheless, deploying and scaling these systems is a challenging problem. A more recent approach in designing unobservable circumvention systems is to imitate popular applications like Skype and HTTP, as suggested by Skype- Morph [28], CensorSpoofer [34], and StegoTorus [35]. However, it has recently been shown [15] that these systems' unobserved ability is breakable; this is because a comprehensive imitation of today's complex protocols is sophisticated and infeasible in many cases. A promising alternative suggested [15, 17] is to not mimic protocols, but run the actual protocols and find clever ways to tunnel the hidden content into their genuine traffic; this is the main motivation of the approach taken in this paper. In this paper, we design and implement SWEET, a censorship circumvention system that provides high availability by leveraging the openness of email communications. Figure 1 shows the main architecture.

II. LITERATURE SURVEY

2.1 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, 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.

DISADVANTAGES OF EXISTING SYSTEM:

- 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.

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.

ADVANTAGES OF PROPOSED SYSTEM:

- 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.

SYSTEM ARCHITECTURE:
SYSTEM REQUIREMENTS:
HARDWARE REQUIREMENTS:

- System : Pentium Dual Core.
- Hard Disk : 120 GB.
- Monitor : 15'' LED
- Input Devices : Keyboard, Mouse
- Ram : 1 GB

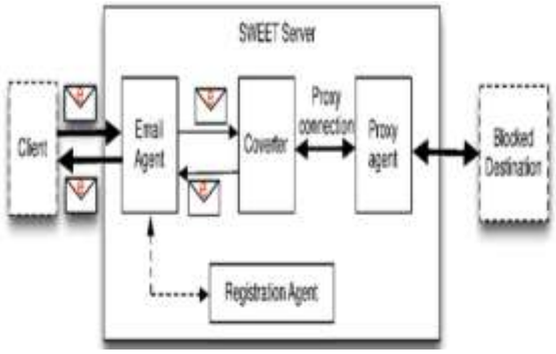
SOFTWARE REQUIREMENTS:

- Operating system : Windows 7.
- Coding Language : JAVA/J2EE
- Tool : ECLIPSE
- Database : MYSQL

III. DESIGN OF SWEET

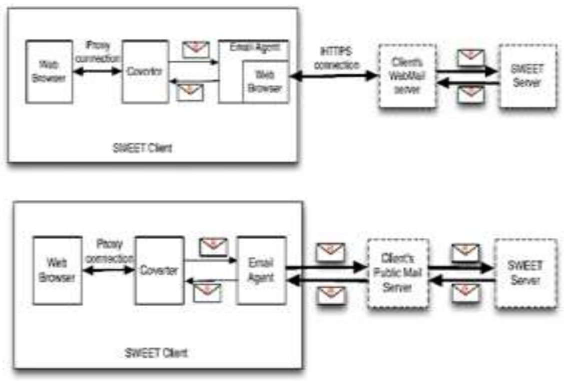
In this section, we describe the design of SWEET. Figure 1 shows the overall architecture. SWEET tunnels network connections between a client and a server inside email communications. Upon receiving the tunneled network packets, the SWEET server acts as a transparent proxy between the client and the network destinations requested by the client. A client's choices of email servicesi) AlienMail An Alien Mail is a mail provider whose mail servers reside out-side the censoring ISP, e.g., Gmail for the Chinese clients. We only consider AlienMails that provide email encryption, e.g., Gmail and Hushmail. A SWEET client who uses an AlienMail does not need to apply any additional encryp-tion/steganography to her encapsulated contents. She simply sends her emails to5the publicly advertised email address of SWEET server5.1 SWEET SERVER

The SWEET server is running outside the censoring region. It helps SWEET clients to evade censorship by proxying their traffic to blocked destinations. Figure 3 shows the design, composed of four elements: Email agent, Converter, Proxy agent, and Registration agent. Here the Email agent is an IMAP and SMTP server.



5.2 SWEET CLIENT

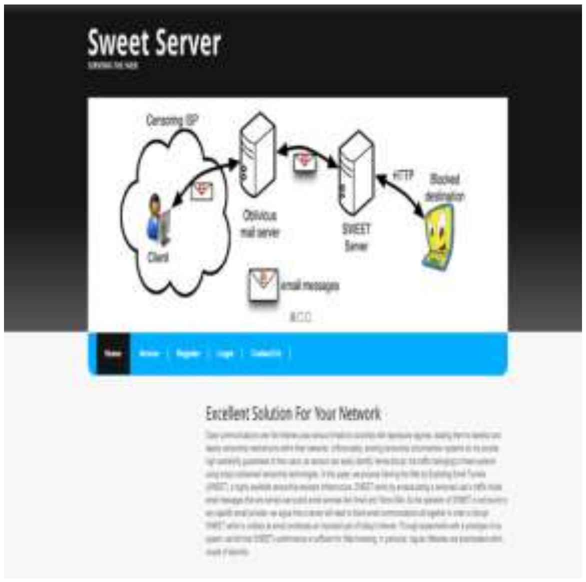
To use SWEET, a client needs to obtain a copy of SWEET's client software and install it on her machine. The client also needs to create one email account, and to configure the SWEET's software with information of her email account. Prior to the first use, the client software registers the email address of its user with the SWEET server and obtains a shared secret key kC,R. We propose two designs for SWEET client: a protocol-based design, which uses standard email protocols to exchange email with client's email provider, and a webmail-based design, which uses the webmail interface of the email provider.We describe these two designs in the following.



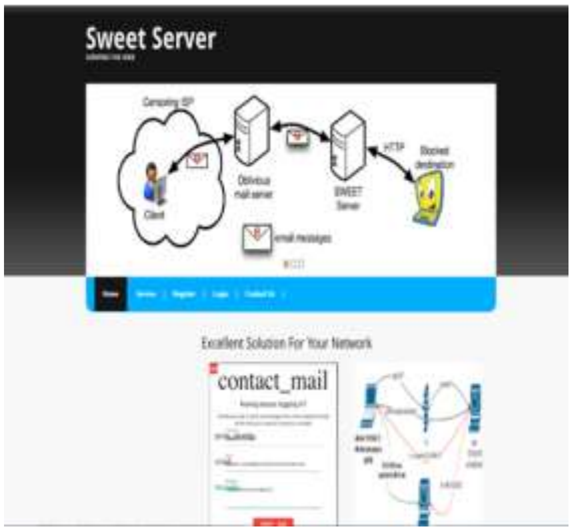
IV. OUTPUT SCREEN

Front-end snapshot of the source node

Home page:

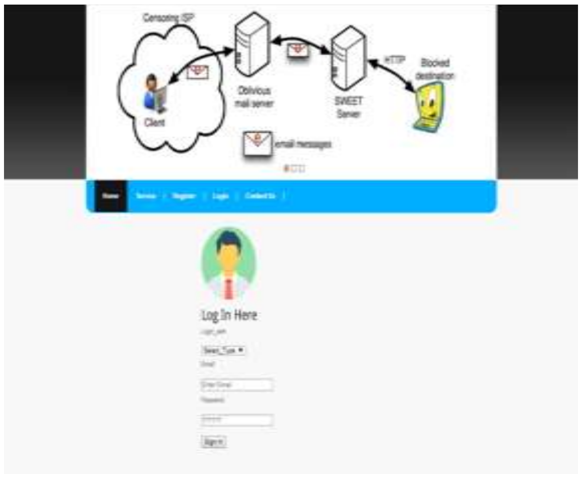


Services:

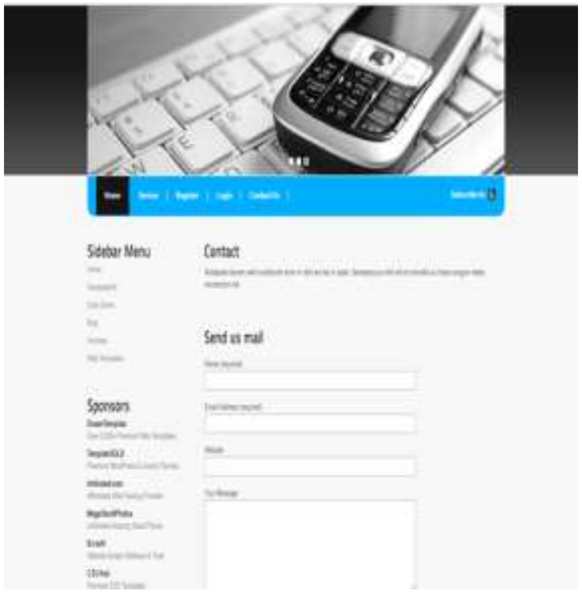


Register:

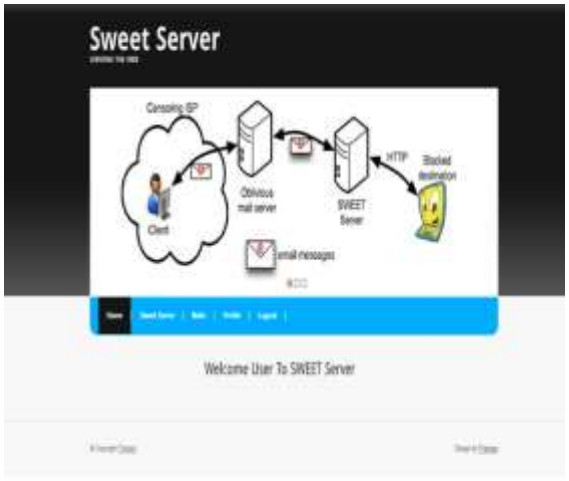
Login:



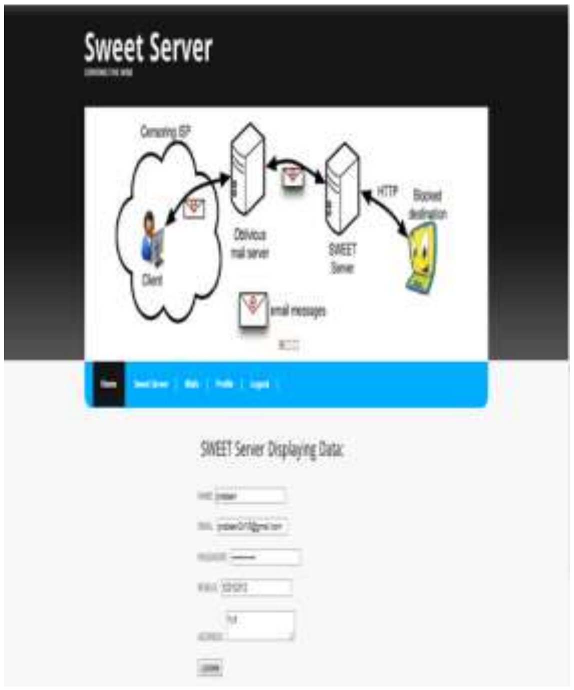
Contact:



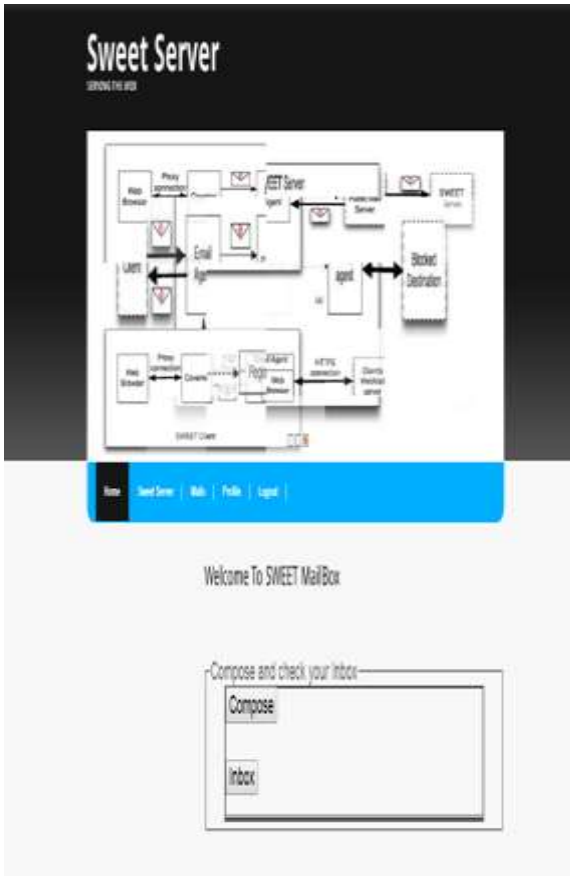
User home:



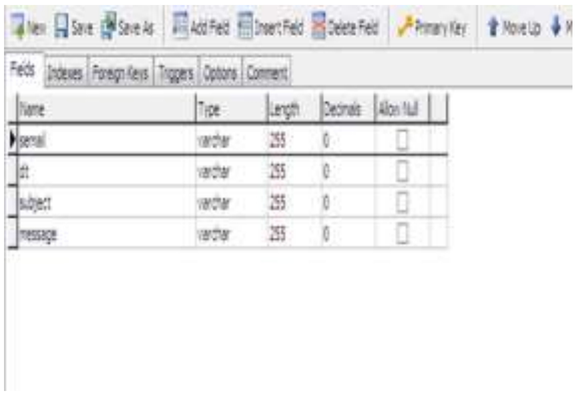
Profile:



Mails

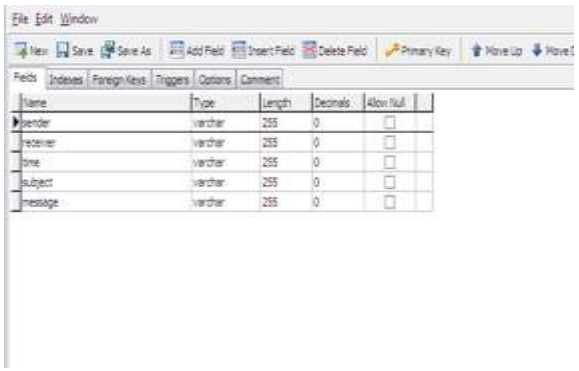


Proxy server:



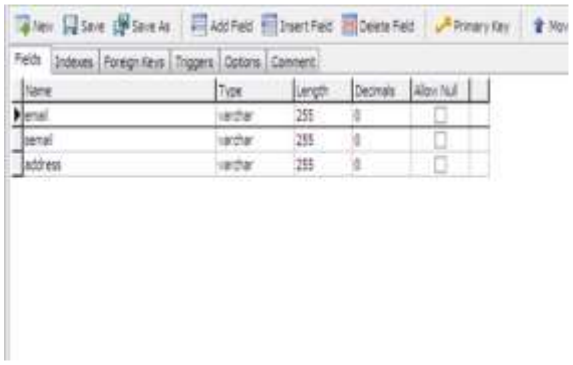
Name	Type	Length	Decimals	Allow Null
semail	varchar	255	0	☐
dt	varchar	255	0	☐
subject	varchar	255	0	☐
message	varchar	255	0	☐

Send mail:



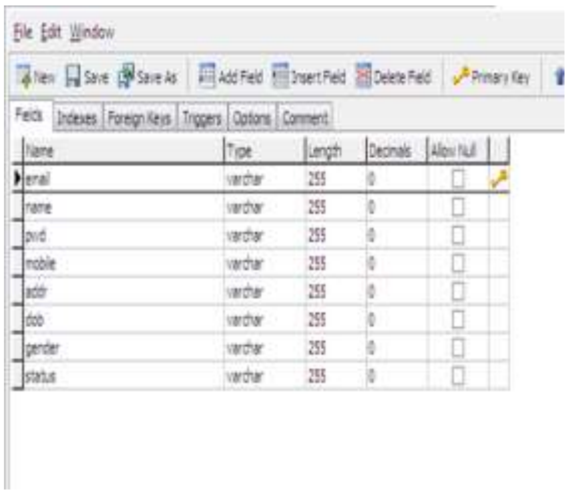
Name	Type	Length	Decimals	Allow Null
sender	varchar	255	0	☐
receiver	varchar	255	0	☐
time	varchar	255	0	☐
subject	varchar	255	0	☐
message	varchar	255	0	☐

Server1



Name	Type	Length	Decimals	Allow Null
email	varchar	255	0	☐
semail	varchar	255	0	☐
address	varchar	255	0	☐

User details



Name	Type	Length	Decimals	Allow Null	
email	varchar	255	0	☐	🔑
name	varchar	255	0	☐	
pwd	varchar	255	0	☐	
mobile	varchar	255	0	☐	
addr	varchar	255	0	☐	
dob	varchar	255	0	☐	
gender	varchar	255	0	☐	
status	varchar	255	0	☐	

V. CONCLUSION

In this paper, we presented 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 ISPsto 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.