2009 WASE International Conference on Information Engineering

SkyF2F: Censorship Resistant via Skype Overlay Network

Shoufeng CAO*, Longtao HE‡, Zhongxian LI*† and Yixian YANG*
*Information Security Center, State Key Laboratory of Networking and Switching Technology
Key Laboratory of network and information attack & defence technology of MOE
Beijing University of Posts and Telecommunications
Email: caoshoufeng@gmail.com
†National Cybernet Security Ltd, Beijing, 100088
‡ Research Center of Information Security, Institute of Computing Technology
Chinese Academy of Sciences, Beijing, China

***Abstract*—Internet censorship is on the rise as an increasing number of countries and companies block or monitor access to parts of the Internet. Many censorship resistant systems have been proposed, which rely on deploying *many access points* to the censored domain. Therefore they face the problem of discovering available nodes and deploying a large number of nodes. Opposite to *many access point* approach, we present a system building on existing overlay network, a low-cost solution for circumventing Internet, called SkyF2F. SkyF2F is a plug-in for Skype client that allows users to establish a covert communication tunnel across Skype overlay network. We describe the design, a prototype implementation and security analysis of SkyF2F. Our security analysis shows that SkyF2F can successfully circumvent several sophisticated censoring techniques.**

***Index Terms*—censorship resistant, covert tunnel, friend-to-friend, Skype**

I. Introduction

Nowadays, Internet becomes a prime facilitator for many people share information freely all over the world. Many countries, political regimes and corporations have attempted to monitor and often restrict access to parts of the Internet by clients who use networks they control. Many of these attempts have been successful, and the use of the Web as a free-flowing medium for information exchange is being severely compromised.

This paper focuses on the challenging technical problems of circumventing Internet censorship and largely ignores the many related political, legal and policy issues. A well known idea is *many access points* which assumes that no user is able to get whole information about available access points. Many systems based on this idea have been proposed, such as Anonymizer [1], Zero-Knowledge [2], Infranet [3], Tor-Bridges [4]. These systems require a client in the censored domain to discover and communicate with an available node outside of the domain. A system with more nodes makes it more difficult for a censor to get all of them. This results in an arms race between the provider of available nodes and the censor that try to detect them. However, these systems face the following problems: deploying a number of nodes or need a number of volunteers, distributing available nodes. Another idea is *all or nothing* which assumes that it is hard to decide based on observing if certain communication data belongs to censored content or not. For example, if all emails are encrypted around the world, then a censor could not scan them by certain words. We investigate how to utilize this idea to design a low-cost system for circumventing Internet censorship. We are aware of the fact that building on top of existing overlays will make the job easier. For example, we could use the Skype overlay network as a messaging transport, a popular plug-in application for Skype client could spread from user to user through the network.

In this paper, we propose a system for circumventing censorship and surveillance of Internet traffic, called SkyF2F. Our system is a plug-in for Skype Client, allowing users to establish covert communication tunnel across Skype overlay network. The tunnel could circumvent several sophisticated censoring techniques and guarantee a certain amount of anonymity, forcing the censor to block *all or nothing* predicament. To assess the feasibility of our design, we implemented a SkyF2F prototype using the Skype “ap2ap” API [5] and conducted a series of tests using client-side Web traces to evaluate the performance of our system. Our experimental evaluation shows that SkyF2F provides acceptable bandwidth for covert Web browsing.

The rest of this paper is structured as follows. Section II reviews other relevant research in censorship resistant systems. Section III gives an overview of Skype overlay network. In Section IV, after describing our threat model and assumptions, we represent the design of our system. Then analyze its security in Section V, as well as limitations. Finally we conclude in Section VI.

II. Related Work

Many existing systems try to achieve censorship resistant by using the “many access points” approach. Anonymizer [1] is one of the oldest such systems. SafeWeb [6], and Zero-Knowledge System [2] use an SSL-encrypted channel to communicate requests to proxies outside of the censored domain, which then return the censored content over this encrypted channel. A more sophisticated system is Infranet [3] which uses steganographic techniques to establish a hidden channel

978-0-7695-3679-8/09 $25.00 © 2009 IEEE
DOI 10.1109/ICIE.2009.134

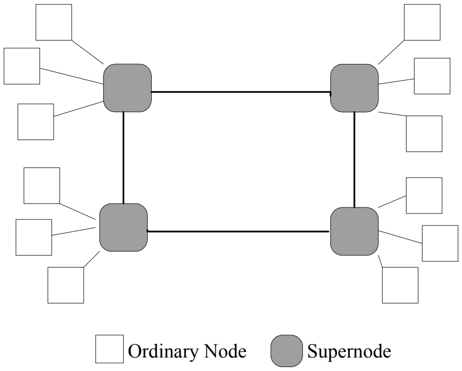

Fig. 1. Model of Skype network.

between a user and a forwarder node. Such a node acts also as normal web server and the hidden channel is embedded into allowed HTTP traffic (for upstream communication the information is embedded into the requested URLs and for downstream the information is embedded into downloaded images). Blocking resistant designs [7], [4] for existing anonymity systems that relies on volunteers operating a large number of access points to the core anonymity system, has been described and implemented by JAP and Tor. Because the censor are actively discovering and blocking the nodes, all these systems face the problem how to distribute available nodes. Keyspace Hopping [8] is a technique which tries to solve the problem of distributing the information about available forwarders. The goal is that no one gets the whole information about all forwarders.

Many other systems have attempted to protect anonymity for users who publish and retrieve censored content. Anonymity systems like Crowds [9], Tor [10], MorphMix [11], Tarzan [12] focus on user privacy, making it difficult to associate requests with the originating user. Freenet [13], Publius [14] and Tangler [15] focus on protecting the anonymity of publishers of censored content and the content itself. Freenet provides anonymous content storage and retrieval. Nevertheless these systems are not really designed to guarantee blocking resistance. A system either has some centralized parts (nodes) which could be blocked or information about all participating nodes could easily be collected and used to block access to the system.

III. Skype Overview

Skype is a peer-to-peer VoIP client that allows its users to place voice calls, send text messages and transfer files to other users of Skype clients, which is very similar to the MSN and Yahoo IM applications in essence. Despite its popularity, little is known about Skype's encrypted protocols and proprietary network. Garfinkel [16], concludes that Skype is related to KaZaA. Network packet level analysis of KaZaA [17] and Skype [18] supports this claim.

As Fig. 1 shows, Skype uses a supernode based peer-to-peer overlay network. There are two types of nodes in the overlay network, ordinary hosts and super nodes. Supernodes maintain an overlay network among themselves, while ordinary nodes pick one (or a small number of) supernodes to associate with; supernodes also function as ordinary nodes and are elected from amongst them based on some criteria. Ordinary nodes issue queries through the supernode(s) they are associated with. Skype implements a number of techniques to circumvent NAT and firewall limitations, and all communications are encrypted to ensure privacy.

Several properties of Skype have made it an attractive candidate for building a censorship resistant system on top of:

- It is free and widely used. It is being actively used by millions of people all over the world.
- All the Skype traffic is encrypted from end to end.
- Skype can automatically traverse most NAT and firewalls with the help of intermediate peers.
- Skype intelligently and dynamically routes the encrypted calls through different peers to achieve low latency. This means that the route and the intermediate peer could be changed during a call.
- Skype provides Developers API [5] that allows users develop their own Skype applications.

IV. System Description

A. Adversary Model and Assumptions

Our discussion on censorship resistant assumes an adversarial model, especially about the capabilities of the censor, following [7]. The attacker (or censor) has the following properties:

- *Controls all (network) links and nodes (routers, proxies etc.) to the outside world.* He can read and analyze all traffic; can delete, change and insert data. He himself, has also free access to the Internet.
- *Does not control (large parts of) the "outside" Internet.*
- *Knows everything about the design of the censorship resistant system* including how the system works and the reason for every design decision.
- *Owns huge amounts of resources*, including money, computing power and human resources.

We assume that the attacker maybe able to use political and economic resources to secure the corporations and entities. For example, the censor can threaten the service providers to remove some troublesome blogs.

We assume that the attacker allows partial access to the Internet. The attacker would like to restrict the flow of certain kinds of information rather than complete blocking. For example, some popular instant messenger (IM) softwares have free access to Internet (or maybe censored by certain keywords).

B. Design Goals

Like other censorship resistant designs, our system seeks to circumvent censorship and surveillance of Internet traffic. Within this main goal, we want to meet a number of goals list below:

- *Low resources cost*: Our system should be easily deployed and used in the real world. Unlike other censorship resistant design that need to deploy a number of nodes or need volunteers, our system is built on existing overlay network, users just install a plug-in for Skype client.

- *Easy to use*: As the Skype clients are widespread over the Internet, user can use our system easily. If a user with restricted internet access (call her Alice) to a censored server, she contacts a user (maybe his friend, call him Bob) with free internet access, Bob serves as a router for Alice and transparently forwards all communication data between Alice and the censored server.
- *Censorship resistant service*: Our system allows user to setup a censorship resistant service, and publish to his friends or the public over the Skype overlay network.
- *Performance*: we seek to achieve acceptable bandwidth for web browsing experiences.

C. System Description

1) Overview: Our System is built on top of Skype overlay network. Skype can be viewed as an overlay network of machines(peers), each with a unique identifier. The only thing required to send a message to a peer is its id. Furthermore, Skype allows us to abstract from routing, clients leaving and joining the network, and other low level issues, and also provides encrypted conversation, ability to cross most NAT and firewall boundaries, high availability.

Our system is a plug-in application Skype Client. As Skype clients are widespread over the Internet, low deployment costs become available and user can use the system easily (without installing any additional software). This is a big advantage compared to other systems like Zero-Knowledge or Infranet. These systems are stand-alone software which need to deploy a number of nodes or a number of volunteers to run their software.

Fig. 2 shows the architecture of SkyF2F system. A Skype user installs the plug-in which can be configured to run as a client, a server or both. A client establishes a tunnel with a SkyF2F server across the Skype Network. The client handles connections from user applications and forwards all communication data across the tunnel. The SkyF2F Server on the other side of the tunnel serves as router, connects to a service and relays data. A Skype application is identified by *userid:appid* pair, like *ipaddress:port* pair in Internet TCP/IP protocols. If a user (call her Alice) with restricted internet access wants to connect to a censored or blocked server, he can contacts a user (call him Bob) with free internet access to the server in his Skype buddy-list. Both of them running the Skype Client with the SkyF2F plug-in (If only one of them has the plug-in, they can share it easily). To begin creating a tunnel, they negotiate the appid (by using Skype VoIP or instant messages) used by Bob. Alice establishes a tunnel with Bob, Bob forwards all communication data between Alice and the server. Bob himself can setup a censorship resistant service in the Skype network, such as a webserver. He can advertise the service appid to his friends or public. Using Skype network also allows Bob to respond to some request and ignore others (for example, if Bob's service tends to get attacked by network adversaries).

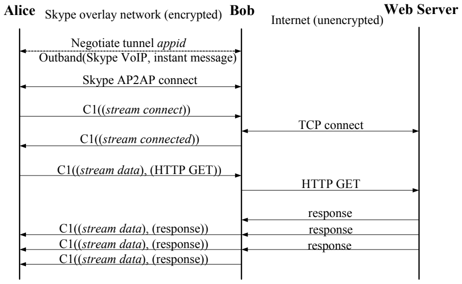
Alice
Skype overlay network (encrypted)
Bob
Internet (unencrypted)
Web Server
Negotiate tunnel *appid*
Outband(Skype VoIP, instant message)
Skype AP2AP connect
C1((*stream connect*))
TCP connect
C1((*stream connected*))
C1((*stream data*), (HTTP GET))
HTTP GET
response
C1((*stream data*), (response))
response
C1((*stream data*), (response))
response
C1((*stream data*), (response))

Fig. 3. Alice establishes a communication tunnel with Bob for web browsing.

2) Establishing a tunnel: Fig 3 shows the messages involved in establishing a tunnel for web browsing. Alice and Bob establish a tunnel across the Skype overlay network using the Skype "ap2ap" API. The "ap2ap" mechanism is relatively low level, and is suitable for transmitting text payloads. Because Skype system provides user authentication and encrypted conversation, currently, for simple and practical reasons, we did not redesign user authentication and session encryption protocol, and we leave this for future work if needed.

Once the tunnel has been established, Alice and Bob can send one another stream cells over the tunnel. The tunnel can be shared by many TCP streams. Traffic passes along the tunnel in fixed-size cells. Each cell is 512 bytes, and consists of a header and a payload. The header contains a streamID (stream identifier: many streams can be multiplexed over a tunnel); the length of the payload; and a stream command. The entire contents of cell (including the header and the payload) are encoded and decoded together as the stream cell moves along the tunnel. The stream commands are: *stream begin*, *stream connected*, *stream data*, *stream end*, *stream teardown*. More detail is given in the next section.

3) Opening and closing streams: When the SkyF2F client accepts a new connection from Alice's application, it opens the stream by sending a *stream begin* cell to the SkyF2F server, using a new random streamID. Once the SkyF2F server connects to the destination host, it responds with a *stream connected* cell. Upon receipt, the SkyF2F client now accepts data from the application's TCP stream, packaging it into *stream data* cells and sending those cells through the tunnel to the destination server.

Closing a stream is analogous to closing a TCP stream: it uses a two-step handshake for normal operation, or on-step handshake for errors. If the stream close abnormally, on side simply sends a *stream teardown* cell; if the stream closes normally, one side sends a *stream end cell*, and the other side responds with its own stream end cell.

V. System Analysis

In this section, we discuss SkyF2F's capability to achieve censorship resistant against a determined adversarial censor and some limitations.

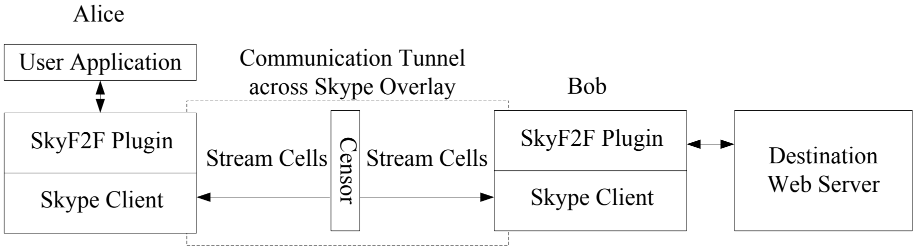

Fig. 2. SkyF2F architecture.

A. Attacks

A censor may block access to various parts of the Internet based on IP address or prefix block, DNS name, or port number. Additionally, a censor can block access to content by filtering out content that contains keywords. In Skype F2F network, users are identified by a pseudonym and end to end communication is based on peerid. Furthermore, all messages forwarded in the network are encrypted. So blocking by Internet address or keywords is no longer feasible.

A censor might mount a passive attack in an attempt to discover a SkyF2F communication tunnel. This type of attack becomes significantly harder to mount in a global overlay with thousands of nodes and a large mount of normal traffic.

A censor might build up a database of "fingerprints" containing file sizes and access patterns for targeted websites. He can later confirm a user's connection to a given site simply by consulting the database. This attack has been shown to be effective against SafeWeb [6]. It maybe less effective against SkyF2F, since streams are multiplexed within a same circuit, fingerprinting while be limited to the granularity of cells (currently 512 bytes).

A censor might setup a SkyF2F server and publish it to public, in the hope that some user might connect it. Therefore, these unlucky users' are being censored. Currently, we rely on each SkyF2F user trusting the legitimacy of any responder it contacts.

A censor might host a supernode in the Skype network. Because some conversations are being relayed via supernode, in spite of the encrypted conversation, traffic flow can still be analyzed. A censor who targets a suspected SkyF2F server can observe all connections to the target server and discover its users. Since there are thousands of supernodes in the Skype network, the possibility that the supernode forwards traffic for the target server is small. To discover more users correlated with the target server, the censor should setup more supernodes. It is a high resource cost work. Additionally, a SkyF2F client user can only use the "out-of-control" supernodes to avoid this type of attack.

A censor might mount DoS attacks against a target SkyF2F server. By attacking the server to shut it down, reduce its reliability. For example, a censor could act like clients, and establish many tunnels until the server reach its limitations. As a public service, this is a real problem. we take a simple measure of limiting each client's connections and bandwidth. The best defense is to setup a friend-to-friend service provides service only for trusted clients.

For some political reasons or pressures, a service userid might be banned or removed by Skype network administrators. However, such operation has trivial effect on the service, because it is easy to change or register a new userid in Skype network than to change an IP address in the Internet.

B. Anonymity

In Skype peer-to-peer overlay which implements a layer of virtual addressing and message routing on top of the Internet addressing and packet routing infrastructure, users are identified by a pseudonym, messages are targeted to overlay addresses rather than Internet addresses. This mechanism ensures a certain amount of Internet address anonymity. Anonymity can enable censorship resistant, freedom of speech without the fear of persecution, and privacy protection.

Suppose Alice established tunnel with her friend Bob to a web server. If there was a direct connection between each other, the communication would proceed directly, Bob act as a single-hop proxy. If connection is restricted between each other, the communication would be relay through another node. Obviously, the latter guarantees better anonymity.

However, Skype does not guarantee strong anonymity. To improve anonymity, existing anonymity systems could be tailored to our system. For example, we could design Crowds [9] like system, when a SkyF2F server receives the request, it flips a biased coin to determine whether or not to forward the request to another SkyF2F server. We leave this for future work.

C. Limitations

Because our system is mainly based on a black-box system, there should be several security considerations about Skype system.

First, Skype is a close-source software, little is known about Skype's proprietary and secret protocols. One should question the validity of assuming censorship resistant on Skype network. As we know so far, Skype appears to offer significantly security according to some analysis [18], [16]. Because of its proprietary P2P communication model and ability to tolerate restrictive networks, Skype could continue operation in the presence of censorship.

Second, because Skype relies on a central login server, Skype can still be blocked. However, we think blocking a major service like Google or Skype can do actual economic damage.

Finally, we should consider the security inside Skype system. It is unknown if the design of the Skype network makes it possible for some nodes to monitor all conversation traffic. Skype could have security vulnerabilities that a third-party could exploit. It is likely that the Skype system could be compromised by an exceedingly capable engineer with experience in reverse engineering, or by a suitably-motivated insider.

VI. Conclusion

In this paper we have presented a system which enables users to circumvent Internet censorship and surveillance by establish covert tunnel across Skype overlay network. Our system allows a user to establish a covert tunnel with other user or provide a censorship resistant service. We have also described that the system can successfully circumvent several sophisticated censoring techniques, guarantee a certain amount of anonymity and limitations of our system.

We believe that the design presented here could be adapted to other popular IM overlays, for a practical purpose we choose Skype. We have argued for building a censorship resistant system on top of existing overlays and demonstrated the feasibility of doing so to provide availability and robustness guarantees. We hope that the fact that it is based on top of an existing overlay will make the job easier. The principles behind our system can be more broadly applied for other censorship resistant systems.

Since SkyF2F does not guarantee strong anonymity, our next step is to tailor existing anonymity systems to our system and design a censorship-resistant anonymity system over Skype overlay network.

Acknowledgment

The work was supported by the National 863 (No. 2007AA01Z430), National Basic Research Program of China (No. 2007CB310704, 2007CB311100), National Natural Science Foundation of China (No. 60821001), and the 111 Project (No. B08004).

References

[1] Anonymizer. http://www.anonymizer.com.

[2] A. Back, I. Goldberg, and A. Shostack, "Zero-knowledge systems," White Paper, Zero Knowledge Systems, Inc., May 2001.

[3] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger, "Infranet: Circumventing web censorship and surveillance," in *Proc. 11th USENIX Security Symposium*, 2002, pp. 247–262.

[4] R. Dingledine and N. Mathewson. Design of a blocking-resistant anonymity system. Internet draft. [Online]. Available: http://www.torproject.org/svn/trunk/doc/design-paper/blocking.pdf

[5] Skype API reference. https://developer.skype.com/Docs/ApiDoc.

[6] D. Martin and A. Schulman, "Deanonymizing users of the safeweb anonymizing service," in *Proc. 11th USENIX Security Symposium*, Aug. 2002.

[7] S. Köpsell and U. Hilling, "How to achieve blocking resistance for existing systems enabling anonymous web surfing," in *Proc. 2004 ACM workshop on Privacy in the electronic society (WPES 2004)*, Washington, DC, USA, Oct. 2004, pp. 47–58.

[8] N. Feamster, M. Balazinska, W. Wang, H. Balakrishnan, and D. Karger, "Thwarting web censorship with untrusted messenger discovery," in *Proc. 3rd Workshop on Privacy Enhancing Technologies*, Mar. 2003, pp. 125–140.

[9] M. K. Reiter and A. D. Rubin, "Crowds: anonymity for web transactions," *ACM Transactions on Information and System Security*, vol. 1, no. 1, pp. 66–92, Jun. 1998.

[10] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proc. 13th USENIX Security Symposium*, Aug. 2004.

[11] M. Rennhard and B. Plattner, "Introducing morphmix: Peer-to-peer based anonymous internet usage with collusion detection," in *Proc. Workshop on Privacy in the Electronic Society (WPES 2002)*, Washington, DC, USA, Nov. 2002.

[12] M. J. Freedman and R. Morris, "Tarzan: A peer-to-peer anonymizing network layer," in *Proc. 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, DC, USA, Nov. 2002, pp. 193–206.

[13] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," *Lecture Notes in Computer Science*, vol. 2009, pp. 46–66, 2001.

[14] M. Waldman, A. D. Rubin, and L. F. Cranor, "Publius: A robust, tamper-evident, censorship-resistant, web publishing system," in *Proc. 9th USENIX Security Symposium*, Aug. 2000, pp. 59–72.

[15] M. Waldman and D. Mazières, "Tangler: A censorship-resistant publishing system based on document entanglements," in *Proc. 8th ACM Conference on Computer and Communications Security*, 2001, pp. 126–135.

[16] S. L. Garfinkel, "VoIP and Skype Security," *Skype Security Overview–Rev*, pp. 1–5, Jan. 2005.

[17] J. Liang, R. Kumar, and K. W. Ross, "The KaZaA Overlay: A Measurement Study," *Computer Networks Journal (Elsevier)*, vol. 49, no. 6, 2005.

[18] S. A. Baset and H. Schulzrinne, "An analysis of the skype peer-topeer internet telephony protocol," in *Proc. 25th IEEE International Conference on Computer Communications (INFOCOM 2006)*, 2006, pp. 1–11.