Hiding from Big Brother

Martin Schanzenbach, B.Sc.
Betreuer: Dipl.Inf. Matthias Wachs
Seminar Future Internet WS2013
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: schanzen@in.tum.de

ABSTRACT

The present state of Internet surveillance and censorship has prompted the development of sophisticated anonymization and encryption protocols. The importance of anonymity and data encryption has already set foot in people's minds. In this paper, we discuss why it is necessary to not only hide the contents and involved parties of communications, but also the communication itself. We present some of the techniques suitable for hiding communications effectively and finally elaborate possible future implications the use of those technologies would have on Internet usage.

Keywords

Big Brother, Censorship, Obfuscation, Steganography, Morphing

1. MOTIVATION

Our Internet communications today are not only heavily censored in some parts of the world, but also heavily monitored by intelligence agencies, corporations and nation states. Specialized industrial sectors produce technology for surveillance, censorship and with it, oppression of civil liberties[18]. We are no longer dealing with attackers of limited power and influence. Today's communication systems are under attack by states and large corporations with sheer endless capabilities. Especially considering recent revelations that states are performing large scale surveillance of the Internet[7], protocol designers of privacy enhancing software must rethink their threat model. The current state of our security infrastructure is in a sorry state[11] and even implementations of some basic cryptographic routines might be affected[23, 24].

A prime example for this problem in today's world are China's efforts to censor and monitor the Internet. Dissident blogs and non conforming opinions are not tolerated content[19]. At the same time, tools like Tor, that provide anonymization to allow free expression of opinions, are fought and it's users incriminated. Tor is a tool that allows users to browse the Internet anonymously and provides Chinese users with the ability to access the Internet beyond the Great Firewall of China[27]. However, the Chinese government has long blocked the access to the public Tor servers needed to connect to the service. As a result, non-public, "intermediate" servers have been emerging called "bridges", that allow access to the Tor network. But even the undisclosed IP addresses of bridges are blocked after use in many cases[21]. This indicates, that services and traffic are actively monitored and traced.

Hiding only the contents of your communication using encryption is no longer enough. If you are talking with a known dissident or you are using censorship circumventing software it does not really matter what the contents of your communication are. It makes you suspicious and in some places of the world this is enough to put you in danger. The above is the classical dissident versus state scenario. One could say that any attempt in hiding is the same as criminals trying not to get caught by the police. After all, in the affected states the dissident is treated as a criminal. As this is the case, we have to hide this data and communication. We need solutions to completely hide our information, services and data traffic. Only then can we assure that we are not being labeled "suspicious" by mentioned authorities when using other anonymizing tools. In this paper we will introduce the concepts "anonymous clients", "anonymous services" as well as "anonymous traffic and content" along with real world examples. Finally, we will discuss the viability of those technologies and the impact wide-spread use might have on the Internet. In this paper we will discuss some anonymity tools that are used today. Furthermore, we will discuss their viability in our current situation.

There are further cases where communication partners do not want a third party to actually notice their communication in the first place. Copyright holders often watermark their content, invisible to the user but readable either by devices that can process the content or becoming readable if the content is copied. Both methods are used to enforce Direct Right Management (DRM) for intellectual property. Telecommunication companies might as well want to secretly add information to network traffic and packets to discriminate between traffic flows and "prefer" some packets over others. For example an internet service provider could charge companies that provide services over the internet for a prioritized treatment of their traffic, giving them an edge over competitors. Even though a clear violation of net neutrality, it is a use case for hiding information. In this work we will largely focus on the dissident versus state scenario.

The remainder of this paper is structured as follows: First we will introduce our attacker in Section 2. Then we will present some related work on information hiding in Section 3. In Section 4 we present various anonymization techniques. Finally, in Section 5 and Section 6 we will reflect on the viability and implications of those anonymization techniques.

doi: 10.2313/NET-2014-03-1_08

2. THREAT MODEL

In our threat model the adversary is unable to break cryptographic primitives. But, we acknowledge the fact that the implementations of such primitives can be compromised[23, 24]. The attacker is powerful enough to block, disrupt or alter the network communication between two parties. In particular, the adversary is more powerful than the communication partners, but does not control the software they use. The adversary is suspicious of unknown or unreadable information like encrypted communication that does not match any known protocol. He knows our employed techniques and will censor accordingly. In other words, if he is not able to identify our communication as "acceptable" by his standards, he will try to attack our communication. We also assume that our attacker has extensive legal power, as any nation state has. Thus it is possible for him to coerce legal entities like companies to redact any information, including user information, related to any services they provide.

As [26] we define a "whitelisting censor" that has defined a set of allowed technologies and protocols. He monitors communication at least on vital crossings and collects metadata like IP addresses and communication frequency as well as communication content. It is enough to imagine a state forcing ISPs to backup all connection data and perform deep packet inspection, as well as using them to block certain technologies or hosts all together. Any technology, protocol, host etc. not on the whitelist is considered suspicious and will be attacked or blocked. On the other hand whitelisted traffic is considered, for example economically, essential by the censor and will not be blocked.

Accordingly we define a "blacklisting censor" that, upon identifying an unacceptable communication over some technology or protocol, will add an entry to a blacklist. However, this means that if new technologies, protocols or hosts emerge, the blacklisting censor would always first have to identify this and put a new entry on his list, while the whitelisting censor doesn't. The blacklisting censor is obviously less restrictive. So, in general, we consider the whitelisting censor to be stronger. If we can hide from a whitelisting censor, we can also hide from a blacklisting censor.

3. RELATED WORK

Hiding information is an old idea, very useful for wartime communication of allied forces. Secretly communicating can give one side of a conflict an advantage. Especially political or military espionage comes to mind. One concept to hide information is Steganography[22]. Steganography is the art, or science, of hiding information inside information. The resulting information including the hidden part is then called "Steganogram", "Stegofile" or "Stegotext". It should be noted that by successfully using Steganography no third party can read the hidden information unless it knows the Steganography technique used and is actively looking for it. In this matter it serves the same purpose as encryption. However, it might still be useful to encrypt the hidden information to make it more "random" and thus look like noise. As with encryption Kerckhoffs' principle can be applied to Steganography. The security of the system must not depend on the attackers ignorance of the used algorithms that encode and transform the information payload into a Stegotext. In the early days of Steganography this was not an issue, as the attacker was usually a human being whose detection tools were limited to his senses. Today, however, digital communication and forensics tools are a valid reason to keep Kerckhoffs' principle in mind. One example for Steganography is "Echo Hiding". Echo Hiding uses the features of the human auditory system. When listening to audio from speakers what we hear is the music itself including echoes coming from walls and furniture. However, we do not consciously recognize those echoes. Echo Hiding hides data in an audio stream that when heard sound like "natural" echoes[6].

Information hiding was already employed a long time ago using letters, newspapers or custom contraptions. As those techniques are only of limited use in digital information hiding this section will only give a brief overview. However, as the techniques are very simple the concept can be easily grasped and for digital information hiding the basic idea is the same. For instance, using "invisible" ink make of lemon juice can be used to hide texts on seemingly empty paper, or even better, between the lines of other undiscriminating texts. The receiver can make the hidden message visible by applying heat to the paper. Any intermediate party involved in the transport of the text or actively spying on the communication cannot read the hidden message unless he knows that it has been added and how it has been added. Another technique called "Microdot", conceived by Emanuel Goldberg[2] and mostly used in World War 2, is a lot more sophisticated than the invisible ink, but basically the same concept: The information is hidden in the dots of an "i" or a punctuation character of an inconspicuous text like a newspaper article. A picture of the information to be hidden is taken and its size scaled down to the size of a dot in the text. Because of the small size of the resulting dot it cannot be distinguished from a regular dot by the human eye. Additionally, it is chemically treated to appear as black as the other characters. The receiver uses a microscope and inverse chemical processes to retrieve the hidden message. The theoretical issue of those approaches is that if someone is aware of the technique employed and looks for hidden messages explicitly, it is easy to expose the hidden information. This is because the techniques violate the Kerckhoff principle[14]: The viability of the systems depend on the fact that the attacker does not know how it works and that it is applied.

Finally, transportation mediums that are not actually intended to be used for communication at all can be (mis-)used for exactly that purpose. This has been mostly an issue in regards to information security. For example, the electromagnetic field of a CRT computer monitor can be easily measured and used to recreate the displayed image. Also, electromagnetic fields of a PC change depending on the operations the CPU (or other components) perform. This effect can be used by malicious software to "radiate" otherwise inaccessible information in the device to a remote attacker. However, covert channels can also be used to secretly communicate because it is simply not expected to be used in this way. A very sophisticated example is "meteor burst communication", which uses "the transient radio paths provided by ionized trails of meteors entering the atmosphere to send data packets between a mobile station and a base"[20].

4. ANONYMITY

Initially, we need to to clarify our concept and understanding of "anonymity". In anonymity discussions the "level" of anonymity greatly varies from conversation partner to conversation partner. Ground zero in such a scale would be the discussion about the display of real names in social networks or forum comments. However, for us, anonymity should guarantee that companies, states etc. are unable to find a connection between your data and your identity[1]. Anything in between those two is not considered to be anonymity but "pseudonymity". A prime example are IP addresses in Internet communication. Every user uses at least one IP address to communicate. The address itself does not reveal a lot of information about it's user. However, in combination with the customer data in the ISP's databases the IP address is the key to trace back the user's communications and personal data.

We need to anonymize communication partners as well as the content of the exchanged information to assure anonymous communication. In the following we present concepts that provide anonymity of clients, service and content, repectively.

4.1 Anonymous Clients

Anonymizing the source of a communication on the Internet usually involves obfuscation or hiding of the respective source IP address. Proxies and Virtual Private Neworks are common tools to achieve this. However, there is also more sophisticated software like Tor.

4.1.1 Proxies

Using proxies is a straight forward way to hide the source of communication. Proxies are surrogates that are used to hide the source IP address. If a user wants to browse a website he contacts the proxy server and tells it to do so for him. The webserver will only ever communicate with the proxy's IP address (Figure 1). A common protocol for proxies is SOCKS. One major disadvantage of proxies is that they only support HTTP and sometimes HTTPS. Any other protocol will not be proxied and the IP address not hidden. Another issue is that the proxy operator's integrity greatly determines the viability of this anonymization service. All connection information might be stored on the proxy server.

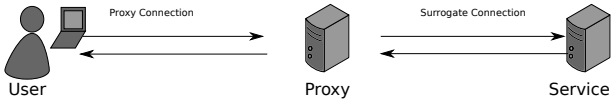

Figure 1: Illustration of a proxied connection.

4.1.2 Virtual Private Networks

Virtual Private Network (VPN) tunnels provide the user with an encrypted tunnel that can be used to access services. The user connects to a VPN-Gateway and redirects his traffic though it. All the traffic exits the tunnel at the exit point and to the service it looks like the user's IP is that of the exit point (Figure 2). The user's real IP remains unknown to the service and service and source cannot be correlated. VPN tunnels are usually fee-based services operated by companies. However, as with proxies, the VPN-Gateway is the first target for any attacker that wants to learn the user's IP addresses. Thus, the anonymity provided depends on the integrity of the service provider. In terms of anonymity, the VPN tunnels have no advantage over proxy's. But they offer broader protocol support, higher data rates and reliability. The latter two usually only if it is a paid service.

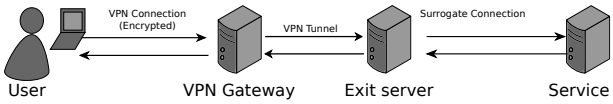

Figure 2: Illustration of a VPN tunnel.

4.1.3 The Onion Router

A popular anonymizing tool is called "The Onion Router"[4], Tor. In the Tor system, the user uses a client called the "Onion-Proxy" or the "Tor-Browser" to connect to the Tor network, a set of connected Tor servers listed in a directory on directory servers. When the user wants to send data to a service it selects a subset of Tor servers and retrieves their public keys from the directory servers. The data is encrypted successively with the public keys and sent to the first server (the one corresponding with the last public key used to encrypt the data). Upon receiving the encrypted packet the server will decrypt the first encryption layer and send the resulting packet to the next server. This scheme continues until the final encryption layer is decrypted. This final server is called the "Exit-Node" and is the surrogate for the user's connection (Figure 3). Client-to-Server and Server-to-Server communication inside the Tor network is also encrypted. In Tor the only server that learns the user's IP is the first server the user sends the encrypted packet to. However, this server does not know the destination of the IP packet, as it cannot decrypt the contents. Any intermediate servers learn nothing about the user and the Exit-Node only learns about the destination of the IP packet. But as the Exit-Node can read the contents of the IP packet it is important that the payload is encrypted using End-to-End encryption like TLS/SSL with the service.

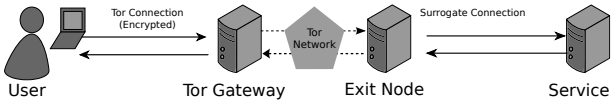

Figure 3: Illustration of a Tor connection.

4.1.4 Anonymous Remailer

For email, anonymous remailer systems can be used[8]. Basic anonymous remailer receive emails from users and strip all headers that can be used to identify the sender. Four kinds of Remailer can be distinguished: Type 0-3. The higher the number the more anonymity can be provided. A Type-3 Remailer is the Mixminion system[3]. One feature of Mixminion is that there is not one remailer, but a set of

doi: 10.2313/NET-2014-03-1_08

anonymizing remailer that communicate using encryption. The sender of an email encrypts the email with the public keys of the remailer servers, similar to the Tor system. A large problem with Mixminion is it's small user base and the fact that the software is still in alpha stadium.

4.2 Anonymous Services

Anonymizing the destination of a communication means we have to hide the service that is communicated with. The approach is called "service hiding". Hiding the service allows the host to remain inconspicuous. If the host is not suspected of running a certain service the censor is looking for, it might fall under the radar and will not be as actively surveilled like a host obviously running the service.

4.2.1 Well Known Ports

As simple and trivial as it sounds, using a different port than the "well known"[13] port of a service can be considered hiding a service. In fact, it is common practice to fool port filters[30].

A technique used to detect running services is the "port scan". A port scan is an attack, where the attacker attempts to connect to all possible ports on the host, trying to enumerate all services that are running. In this process the attacker often learns other important information like operating system, software versions and computer architecture. A firewall can block port scans in general, but as the service provider wants to provide access to the respective services to its users, those ports cannot be blocked.

4.2.2 Port Knocking

A solution to the port scan problem is called "port knocking". Initially, the port is blocked and it is not possible to connect. Only after the client "knocks" by sending a designated knock packet to a predefined port the actual service port will open and can be accessed. The knock packet can be a simple empty protocol data unit directed to the service port or a more sophisticated scheme where it contains a cryptographic identifier[29] or consists of a series of knocks[15]. In either case, an attacker cannot know if a port knocking scheme is used simply by examining the running services on a server. A port scan will always yield no results, as ports are blocked by default.

However, it is important that the port knocking service itself is not detectable, since that service will become the target for an attacker. Not to mention that a port knocking service is anything but inconspicuous in our threat model. The SilentKnock[25] technique was designed with this in mind. SilentKnock assumes that a key and synchronization parameters are exchanged out of band between all the clients and the server. A client initiates a connection using TCP and the respective initial packet (SYN) contains a hidden authenticator token generated from the data exchanged out of band. If the server can verify the token, TCP connection establishment continues, otherwise it fails. The authentication token is a keyed Message Authentication Code (MAC) that is hidden inside TCP header fields containing sequence number and time stamp[17]. It is important to note here that if connection establishment fails, the service will remain silent, instead of returning an error message or actively terminate the connection request, to counteract any probing.

4.2.3 Tor Hidden Services

Another technique to hide services is part of Tor. The goal of "Tor hidden services" is to hide the location and the existence of the service in the network[4]. Initially the service provider chooses a public/private key pair. If a user wants to connect to the service it uses the public key of the service to anonymously connect to a public "introduction point (IP)" using the Tor software. The service is also connected to the IP. Using the now existing connection a "rendezvous point" is negotiated, that is subsequently used by user and service to establish a connection. The user does not learn the IP address of the service and vice versa. Since the service only accepts connections via Tor, attacks like port scanning are useless, considering that the connection establishment in this scheme is additionally relying on a lot of computing intensive cryptography.

4.3 Anonymization of Traffic and Content

Hiding the content and existence of our communication requires sophisticated approaches based on the concept of "Steganography". Content and traffic can be obfuscated in various ways. The difficult problem is making them look inconspicuous.

4.3.1 Obfuscation

Obfuscation aims to alter the communication beyond recognition for an attacker. A very simple way to obfuscate traffic is *encryption*. Encryption protocols are advertised as providing data confidentiality for services on the Internet[5]. An encryption algorithm uses an "encryption key" and "encryption function" to transform plaintext into "cyphertext". The cyphertext can be decrypted using a "decryption key" and a "decryption function". In cryptography we differenciate between two types or encryption: Public and private key encryption. Private key encryption uses the same "secret key" for encryption and decryption. Public key encryption uses different keys for encryption, the "public key", and decryption, the "private key".

Encrypted messages result in protocol data and message content that is no longer readable to any attacker. Thus forbidden conversations (content) and conversation mechanisms (protocols) can be hidden. In particular, encryption defeats any deep package inspection (DPI) mechanisms. Common protocols and software using encryption are HTTPS via TLS/SSL or Skype[1]. Skype audio and video calls are encrypted. HTTPS and the x.509 public key infrastructure (PKI) are used by banks, shops and email providers to encrypt transactions on the Web and protect the users from any third party learning personal information like credit card numbers. This is done by encrypting all sensitive data on the user's PC and sending it to the service where it is decrypted, called "End-to-End encryption". Any third party intercepting the data in-flight will not be able to extract the information.

Two major issues using encryption like TLS/SSL should be mentioned here: First, only the content of the communication between two parties in encrypted (anonymized), not the communication itself. It is easy to learn the identity of the communication partners because the IP addresses are not

[1] http://www.skype.com

doi: 10.2313/NET-2014-03-1_08

encrypted. The second problem is the x.509 PKI. The PKI forms the corner stone of the TLS/SSL system and must be integer. However, as [11] has analyzed, the x.509 PKI is easy to compromise. Furthermore, statistical methods allow attackers to identify traffic patterns, like those of Skype or HTTP traffic[10]. Even if packet contents cannot be read using DPI because the attacker does not have enough resources, traffic patterns like timings and packet size can reveal the protocol used to communicate. An example of such a "statistical classification technique" is the SPID algorithm by Hjelmvik and John[9].

4.3.2 Traffic Hiding

In the face of a very powerful attacker, as defined in our threat model, it is not unlikely that each and every communication is monitored. This includes all the network traffic that is occurring. Using anonymization tools like Tor, it is possible to conceal the identities of either one or the other communication partner. An entity observing the traffic can only know the entity on one side of the communication. However, through statistical analysis an attacker can determine the protocol and in some cases even the content of the communication[9]. Consequently, given a communication between A and B using the protocol P, the attacker can either learn that A is talking using protocol P to somebody *or* that B is talking using protocol P to somebody, but never both. However, as stated in our threat model, simply the use of protocol P might prompt the attacker to block or otherwise attack the communication. As such, it is also necessary to hide the traffic itself.

One approach is to modify traffic patterns in such a way, that the protocol employed is no longer recognizable. Modifying the packet size and the timings the packets are sent will obfuscate the traffic flow. An example implementation of this scheme is the Tor software. Tor servers exchange packets in equally spaced "cells"[4], fixed length messages of 512 bytes. Actual payload data is sliced into 512 byte messages and payload slices smaller than 512 bytes are *padded*. Tor cells are packed into TLS/SSL application data, adding a layer of obfuscation discussed above. Unfortunately, it is exactly those 512 byte cells that make Tor detectable as shown by [26] using statistical analysis. But even if such a sophisticated classification of traffic is not employed by the attacker, the encrypted traffic itself is suspicious. As already mentioned, in China SSL/TLS connections are automatically probed, quickly exposing any Tor activity[27]. Furthermore, in the case of a whitelisting censor, obfuscated traffic not matching whitelisted traffic will automatically be blocked.

A more recent idea to hide from censoring authorities is called "traffic morphing". Whitelisted network traffic is, in our threat model, essential and, if censored, could result in economic disadvantages or other negative effects for the censor. In other words it is inconspicuous because whitelisted (or not blacklisted) by the censor. A traffic "morphing function" can transform any traffic pattern into such a whitelisted pattern, without the censor being able to detect this transformation. A concept best described as "hiding conspicuous traffic inside normal traffic". Even if the censor is aware that there is a technique that allows this transformation and there are users using this technique, he should be unable to distinguish between traffic that contains hidden traffic and normal traffic. The only option he has is to blacklist the previously whitelisted pattern. However, as elaborated above, the censor might be uncomfortable doing that.

StegoTorus[26] is a plugin for the Tor software that aims to add the features of "undetectability" and "unblockability" to Tor. Tor itself tries to conceal it's traffic already using obfuscation techniques. However, as the authors of [26] state, this traffic can be identified as Tor traffic and it can be attacked to learn the protocol of it's original traffic. To counter this, StegoTorus applies Steganography to make Tor traffic look like traffic produced by other client software. Additionally, the equally sized "cells" output by Tor are "chopped", resulting in variable-length "blocks" encrypted with a novel cryptosystem that makes the cyphertext indistinguishable from random data. The authors have created plugins to make the traffic look like either an encrypted peer-to-peer protocol or HTTP. Furthermore, StegoTorus is pluggable to contain more sophisticated Steganography techniques. This is useful as the authors themselves claim[26] that the current Steganography plugin implementations are vulnerable in the face of powerful attackers performing targeted attacks. In the presence of our censor it might be necessary to adjust the cover protocol to one that is on the whitelist or not on the blacklist.

Another approach that also uses Tor's plugin system is SkypeMorph[16]. SkypeMorph aims to hide Tor traffic inside Skype traffic. Skype traffic, or more specifically encrypted video chat traffic, is very suitable for hiding information. First of all in a regular Skype session there is constant flow of information in the form of audio and video data. This constant flow allows the source traffic to be morphed into Stegotext with very little delay, unlike for example HTTP traffic, which does not usually exhibit a constant flow of packets and thus the data rate is not very high. Also, Skype traffic is encrypted which means our traffic is obfuscated by design. A SkypeMorph connection is established by calling a contact using Skype. This results in three prerequisites: Tor, the SkypeMorph plugin, a SkypeMorph bridge to connect to and Skype accounts. The SkypeMorph session is initiated by exchanging public key material over the Skype text chat with an out of band selected bridge. The bridge's Skype ID needs to be added to the client's contact list beforehand. Once a shared secret has been generated by client and bridge, a Skype video call is initiated. Using this "Skype tunnel" the actual Tor traffic is shaped to look like a Skype video call and this data is sent instead of audio and video data. However, we consider this method to have a major flaw: A Skype account is needed and the integrity of the Skype authentication servers as well as the official API kit (which is used by SkypeMorph) is crucial. In our threat model relying on the integrity of a U.S.-based company is a major flaw, as it can easily be coerced to submit potentially incriminating data (at least the Skype account information) and disable bridge accounts due to suspicious activity.

ScrambleSuit[28] is a thin protocol layer above TCP. It aims to negate the shortcomings of Tor by adding the feature of non-blockability using a polymorphic payload and a simple authentication mechanism. ScrambleSuit connections can only be established if both parties can prove their knowledge

of a secret that is shared out-of-band. It is proposed that the Tor bridge distribution mechanism should be used for this purpose. The authors extensively discuss possible authentication mechanisms including Uniform Diffie-Hellman and Session Tickets. As mentioned ScrambleSuit also provides traffic analysis resistance by flexibly generating "protocol shapes" that resemble common "whitelisted" protocols. Protocol shapes are determined in ScrambleSuit by packet length and inter-arrival times between packets. ScrambleSuit servers can, unlike SilentKnock systems, be actively probed. However, since the probing client cannot authenticate itself without the shared secret, the server will simply not answer. The attacker will only learn that the server is online and accepting connection on the given port. Clearly ScrambleSuit has an advantage over SkypeMorph because it doesn't rely on a service provided by a U.S.-based company. Choosing ScrambleSuit over StegoTorus is also advisable, as it already supports common, by today's censors whitelisted , protocols and includes a scheme where our bridge cannot easily be probed.

5. ISSUES

In this Chapter we want to look at some of the issues of the presented technologies. We clearly defined our attackers as very powerful and highly suspicious. But, we only considered low-level network anonymization and hiding practices. In reality, it might be easier for an attacker to deanonymize users from higher-layer protocols, such as plain (as in unencrypted) HTTP, DNS or email. Furthermore, the proposed solutions all trade anonymity for convenience and performance. A development that, we think, is undesirable in a free Internet.

5.1 The High-Level Issue

All the software and techniques discussed above require the user to always have privacy in mind. A careless user can be deanonymized no matter how sophisticated his traffic or services and information is hidden. In this respect, anonymization technologies can be deceptive. For instance, HTTP usage over Tor can lead to information leakage[12] that cannot be contained by such low-level protocols presented here. While we consider the presented techniques, unless otherwise stated, technically sound, careless usage of higher level protocols such as HTTP, email or DNS can also lead to deanonymization. Spy- and malware, i.e a compromised host system, will circumvent any deanonymization software. Without a user's privacy conscience, the best anonymization protocol is useless. A first rule of thumb can be to always use encrypted End-to-End protocols over the hidden and anonymized channels. But even then, the user needs to carefully select the information passed on to the other side and judge how well it can be trusted with it.

5.2 Interdependencies of Hiding Tools

In recent years something that looks more and more like an arms race between censoring authorities and dissident censors can be observed. Whenever a new or improved privacy enhancing technology is employed, there is a response designed to counteract and render it useless. At the same time, when such a tool becomes blocked or otherwise compromised, it is no longer used or, if possible, improved to be immune against the attack.

The problem that arises, though, is that the power, capabilities and knowledge of our attacker are usually unknown. It is generally assumed that an adversary can not theoretically break cryptographic primitives, but he can exploit bugs and weaknesses in the implementations. Those are, of course, not disclosed by an attacker, as it is his own personal back door.

Paradoxically, it might also not be in the victims interest to immediately disclose and fix vulnerabilities in the software's implementation. At first, this seems counter intuitive but it is actually a smart move in the presence of an active attacker on our systems: If a vulnerability is quickly disclosed and a fix released, the adversary will no longer waste his time to find this exploit. On the other hand, if there is a fix and the vulnerability is not disclosed before the attacker can produce an exploit, valuable time is bought for the service provider and users. When the time comes, and the adversary exploits this vulnerability, we can simply patch your software with an already prepared fix, rendering the attackers efforts useless instantly. This is an approach taken for example by the Tor project.

Both sides of this battle have evolved over the past years and employ complex techniques to either impose or circumvent censorship. For users, if this trend continues, it means they will have to use increasingly inefficient and complex software systems to freely communicate. An arms race like this threatens the usability and stability of the Internet and with it today's primary social interaction medium.

We defined our attacker as an entity is not looking to block all communication (i.e. "turn off the internet"). Only communication that he deems necessary to censor will be censored. He will try to walk a thin line between generating digital civil unrest and major economic losses because of censorship and annoying but acceptable limitations. Here, we have reached a point in our discussion where the power of our attacker is limited by the actions and reactions of the general public and private sector instead of science and technology. Maybe a time will come where the Internet has become so over-engineered, heavily surveilled and probed that users no longer accept the status quo and rise by creating a new one.

6. DISCUSSION

We have examined various technologies under the assumption that hiding communication is as important an issue as anonymization communication partners in today's censored and monitored computer networks. While the presented technologies offer various degrees of effectiveness they all share the common problem of complexity. Also, they provide the user with a deceptive feeling of security while at the same time being technically sound. In this regard, we have shown how, in our point of view, the development of hiding and anonymization technology might continue in the face of monitoring and censoring attackers. Not all is lost in the battle between anonymity and incisive intrusion of digital life. However, unless fundamental changes in our social and political mind happen, the resulting technologies are but a crutch to move in a broken Internet.

doi: 10.2313/NET-2014-03-1_08

7. REFERENCES

[1] H. Bleich. Mythos anonymität. *c't 2013 Heft 20*, 2013.

[2] M. K. Buckland. Histories, heritages, and the past: The case of emanuel goldberg. *The History and Heritage of Scientific and Technical Information Systems*, pages 39–45, 2004.

[3] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a type iii anonymous remailer protocol. In *Security and Privacy, 2003. Proceedings. 2003 Symposium on*, pages 2–15. IEEE, 2003.

[4] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. Technical report, DTIC Document, 2004.

[5] A. Exner. *Secure Socket Layer (SSL)-Sicherheit im Internet*. GRIN Verlag, 2008.

[6] D. Gruhl, A. Lu, and W. Bender. Echo hiding. In R. Anderson, editor, *Information Hiding*, volume 1174 of *Lecture Notes in Computer Science*, pages 295–315. Springer Berlin Heidelberg, 1996.

[7] Guardian. The nsa files. *http://www.theguardian.com/world/the-nsa-files, Accessed 21.09.2013*, 2013.

[8] C. Gulcu and G. Tsudik. Mixing e-mail with babel. In *Network and Distributed System Security, 1996., Proceedings of the Symposium on*, pages 2–16. IEEE, 1996.

[9] E. Hjelmvik and W. John. Statistical protocol identification with spid: Preliminary results. In *6th Swedish National Computer Networking Workshop (SNCNW)*, 2009.

[10] E. Hjelmvik and W. John. Breaking and improving protocol obfuscation. Technical report, Department of Computer Science and Engineering, Chalmers University of Technology, 2010.

[11] R. Holz, L. Braun, N. Kammenhuber, G. Carle, and T. U. München. The ssl landscape - a thorough analysis of the x.509 pki using active and passive measurements.

[12] M. Huber, M. Mulazzani, and E. Weippl. Tor http usage and information leakage. In *Communications and Multimedia Security*, pages 245–255. Springer, 2010.

[13] IANA. Service name and transport protocol port number registry. *http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml, Accessed 21.09.2013*, 2013.

[14] A. Kerckhoffs. *La cryptographie militaire*. University Microfilms, 1978.

[15] M. Krzywinski. Port knocking: Network authentication across closed ports. *SysAdmin Magazine 12(6),*, 2003.

[16] H. M. Moghaddam, B. Li, M. Derakhshani, and I. Goldberg. SkypeMorph: Protocol Obfuscation for Tor Bridges. In *Computer and Communications Security*, Raleigh, NC, USA, 2012. ACM. http://www.cypherpunks.ca/ iang/pubs/skypemorph-ccs.pdf.

[17] S. J. Murdoch and S. Lewis. Embedding covert channels into tcp/ip. In *Proceedings of the 7th international conference on Information Hiding*, IH'05, pages 247–261, Berlin, Heidelberg, 2005. Springer-Verlag.

[18] K. Page. Gamma attempting to export surveillance tech out of switzerland. *https://www.privacyinternational.org/blog/gamma-attempting-to-export-surveillance-tech-out-of-switzerland, Accessed 21.09.2013*, 2013.

[19] W. S. J. Paul Mozur. An inside look at china's censorship tools. *http://blogs.wsj.com/chinarealtime/2013/08/30/an-inside-look-at-chinas-censorship-tools, Accessed 21.09.2013*, 2013.

[20] F. A. P. Petitcolas, R. Anderson, and M. Kuhn. Information hiding-a survey. *Proceedings of the IEEE*, 87(7):1062–1078, 1999.

[21] Phobos. China blocking tor: Round two. *https://blog.torproject.org/blog/china-blocking-tor-round-two, Accessed 15.09.13*, 2013.

[22] N. Provos and P. Honeyman. Hide and seek: An introduction to steganography. *Security & Privacy, IEEE*, 1(3):32–44, 2003.

[23] B. Schneier. Nsa surveillance: A guide to staying secure, 2013.

[24] B. Schneier. The us government has betrayed the internet. we need to take it back, 2013.

[25] E. Y. Vasserman, N. Hopper, J. Laxson, and J. Tyra. Silentknock: practical, provably undetectable authentication. In *Proceedings of the 12th European conference on Research in Computer Security*, ESORICS'07, pages 122–138, Berlin, Heidelberg, 2007. Springer-Verlag.

[26] Z. Weinberg, J. Wang, V. Yegneswaran, L. Briesemeister, S. Cheung, F. Wang, and D. Boneh. StegoTorus: A Camouflage Proxy for the Tor Anonymity System. In *Computer and Communications Security*, Raleigh, NC, USA, 2012. ACM. http://web.mit.edu/frankw/www/papers/ccs2012.pdf.

[27] P. Winter and S. Lindskog. How the Great Firewall of China is Blocking Tor. In *Free and Open Communications on the Internet*, Bellevue, WA, USA, 2012. USENIX Association. https://www.usenix.org/system/files/conference/foci12/foci12-final2.pdf.

[28] P. Winter, T. Pulls, and J. Fuss. ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship. In *Workshop on Privacy in the Electronic Society*, Berlin, Germany, 2013. ACM. http://www.cs.kau.se/philwint/pdf/wpes2013.pdf.

[29] D. Worth. Cok: Cryptographic one-time knocking. *Talk slides, Black Hat USA*, 2004.

[30] S. Zander, T. Nguyen, and G. Armitage. Automated traffic classification and application identification using machine learning. In *Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on*, pages 250–257. IEEE, 2005.