# Review of Web Filtering Processes and Implementation of a Web Circumvention

Ana CEBAN, Lowel BUZZI, Gautier GEORGEON
TELECOM Nancy, Master in Computer Science School,
Part of Université de Lorraine,
193 Avenue Paul Muller, 54600 Villers-Lès-Nancy, France

Email: - ana.ceban@telecomnancy.eu
- lowel.buzzi@telecomnancy.eu
- gautier.georgeon@telecomnancy.eu

Thibault Cholez
Associate professor,
Université de Lorraine,

Email: thibault.cholez@telecomnancy.eu

***Abstract*—Some text.**

***Index Terms*—web filtering, web circumvention, tcp, ip, dns, proxy**

## I. Introduction

As the internet access is growing each time faster, that stills the same for its providers. Creating a website has never been such simple. Thus, people can be interested in blocking a website access, no matter their motivation: governmental decision, personal interests... Depending on the huge technical diversity of internet, we will only focus on web filtering in this article.

This article is intended for an intermediate audience, since we will remind some base knowledge about network packets and internet, and also work to obtain an interesting result for people who might want to filter using our technique.

The world wide web comes from the arpanet project in the USA, for military purposes. That was the first network to transmit packed in peer-to-peer. Things has changed and now everyone can send packets all over the world, considering you have a computer. The TCP/IP model is at the base of this functionality: in fact, both of TCP and UDP are protocols used with the IP protocol. Both are using the routing tables to reach the endpoint, but UDP sends the packets one time, without verifying any reception, while TCP initializes a session with a three steps handshake. Concerning the security, TLS 1.3 is currently used with TCP: that corresponds to HTTPS. Having TLS or not will have an essential impact on our web filtering technique choice. [**?**]

The first step is to define the websites to be filtered, based on various criterias. A realistic dimension has to be take into account: indeed, depending the rights we have, we won't (or we will) be able to restrict the desired accesses. That corresponds to the second step, choosing a filtering method. Ideally, the process used restricts only the website list we made, but that is quite impossible. In fact, we observe that some sites are still reachable despite they're in our list (false positive), and other sites not designated before are restricted (false negative). Both of these rules are inversely proportional. [**?**]

**Place here three fundamental questions and how we'll proceed**

## II. Requirements

We can consider that circumventing a filtering technique fits to either add an equipment in the network, or use a technology at the client side.

## III. Existing filtering techniques

So as to bring an interesting point of view in this article, we have to be aware of the existing techniques about web filtering.

### A. TCP/IP Content Filtering

*1) Who's likely to implement it?:* Nowadays, the main actor who's likely to use this feature is the user himself. That is due to the fact that HTTPS has restricted the possibility to intercept packets during transmission and thus filter especially web pages. Although this feature is not really used the current days, we present what has existed before.

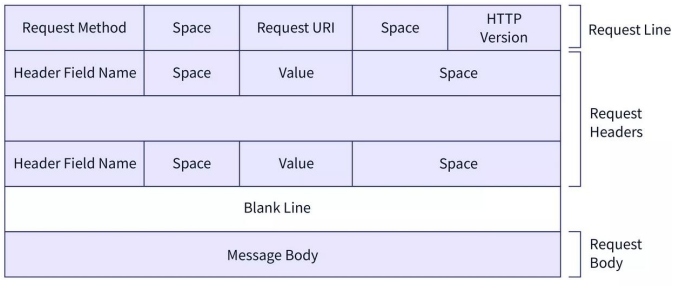


Fig. 1. The HTTP model.

*2) Functional description:* At first glance, we could think that this process is deprecated due to the systematic use of HTTPS nowadays, but in fact, many client features are available in order to achieve web content filtering. [**?**] Indeed, the user could either install a browser addon or an external software. Here, we are accessing the website, but before displaying its content, we check on our computer if it is acceptable. In addition, navigators such as Google or Bing offer to filter inapropriate links for their users. In this case, we don't access the website at all. With these methods, the packets aren't filtered, and we don't have to face the fact that the content is not available due to HTTPS.
However, techniques working on the transmission line exists, in case we are using HTTP. First of all, the internet service provider can check the web pages content before allowing the connexion. Another possibility is to place a proxy between the client and the server, but we have to assert we have the right to place this proxy. It can be transparent: it doesn't modify the application layer, or it can be a web proxy. One idea could be the proxy has a list of forgiven sites, regularly updated based on content, and when a user polls the proxy, it is either accepted or refused (indirectly header filtering).

*3) Bypassing techniques:* Today, web content filtering can be implemented through two ways: select the authorized connexions before using HTTPS, or establish a connexion and decide to display the pages or not. That's why there're no bypassing methods unless in the case the user decides to stop or add censorship on its computer.

### B. TCP/IP Header Filtering

*1) Who's likely to implement it?:* Anyone who has the right to modify the rule of a router or add a proxy server can implement this feature. But in fact everyone is able to forbid addresses on his computer, so everyone can use it.

*2) Functional description:* The header filtering technique consists in blocking a blacklist of IP addresses in a router. It is usually made with a firewall. Because of the web aspect, we are interested in blocking the HTTPS port (443). This mecanism is pretty simple but not precise enough: some DNS domains have the same IP address, causing legal websites to be forbidden. [**?**]

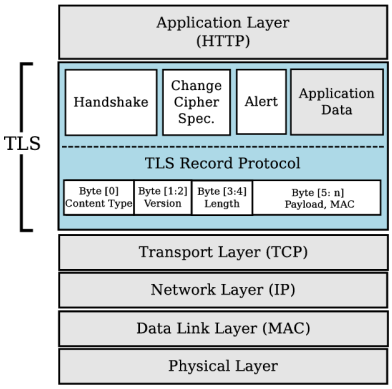


Fig. 2. The TLS model.

IP addresses may be brought to change regularly, that's why the blacklist needs to be updated really often: this can cause slowness and clean sites may be incorrectly considered as forbidden. The way to forbid another IP depends on its algorithm: other filtering techniques may be used with IP filtering.

*3) Bypassing techniques:* This method is local, that is to say we need that the only way to get to the forbidden site be along our filtering router. Otherwise, it is sometimes easy to add a proxy in order to modify the IP address. This is the same idea with a virtual private network.

### C. Proxy Based Filtering

*1) Who's likely to implement it?:* Almost everyone can nowadays buy a cheap proxy on their computer, so that they're able to filter their own content. But that is different from a real server proxy used by companies or the government.

*2) Functional description:* Proxies are ideal for filtering connections considering they're intermediate servers, allowing both to separate a machine from others, or process a HTTP request.
HTTP proxies generally need the user to do an action in order to be used.
They are often used at company scale, in order to permit some content filtering before the proxy sends the request to the desired website.

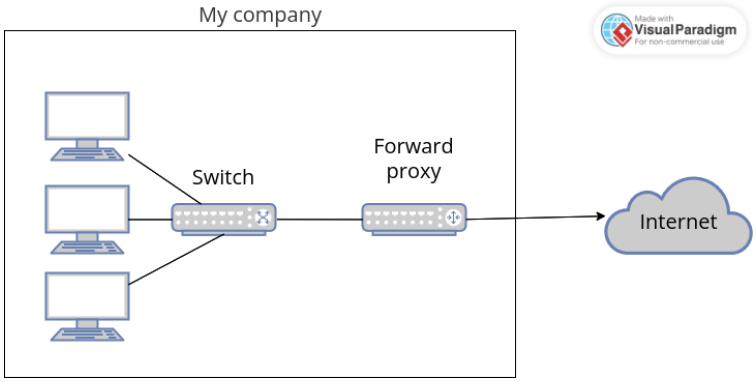


Fig. 3. A classic company proxy.

*3) Bypassing techniques:* Bypassing a proxy depends on the network infrastructure, in some cases, but most of the time using another proxy in order to simulate another ip address can be sufficient. Since proxies are now pretty affordable for personal use on the online market, everyone can try to connect to different websites, switching parameters of the proxy in order to see different results.

### D. DNS Deregistration

*1) Who's likely to implement it?:* DNS (Domain Name System) filtering is mostly implemented in order to regulate the content available online. It is often done by the ISP (Internet Service Provider) as it has the default DNS resolver uless specifically changed by the user. They comply with the gouvernemnt rules to block access to inappropriate or

illegal content but also for censorship purposes. This especially applies to some authoritative coutries where political content is also inclined to be prohibited. Organizations, such as companies and educational institutions, also implement DNS filtering to restrict internet usage. They may block websites that are not work related, sites that are categorised as social media or illegal content. Additionally, they may want to block domains associated with malware, phishing or other cybersecurity attacks.

*2) Functional description:* The process starts when the user types a url query into the search engine then there are a couple of things that happen:

- first the domain name query is being sent to a server called the DNS resolver, usually provided by the ISP (Internet Service Provider) or a third-party service (cf après)
- the DNS resolver will then first look into its cache to see if the domain name has been already resolved or it will redirect the domain query to another DNS servers, each responsible for certain domain names until it obtains the ip address of the corresponding website
- it then sends this ip address to the user so the user can establish a connection with the website

Consecuently no website can be accessed without the DNS resolution, unless of course the user inserts directly the ip address. The way that DNS filtering work is by intercepting the DNS query.

Then the domain name is being analysed, meaning the DNS checks if it contained in a *blocklist* that may be fixed by the organisation filtering policies or another criteria, or if it corrreposponds to a category that is prohibited (ex: social media for a company)

If the site is blocked, the url cannot be accessed and the DNS filter either redirects to a safe page or diplays an error or another predefined page.

DNS filtering can work also with an *allowlist* instead of a blocklist, meaning only domains that are on the list can be accessed and others are blocked by default.

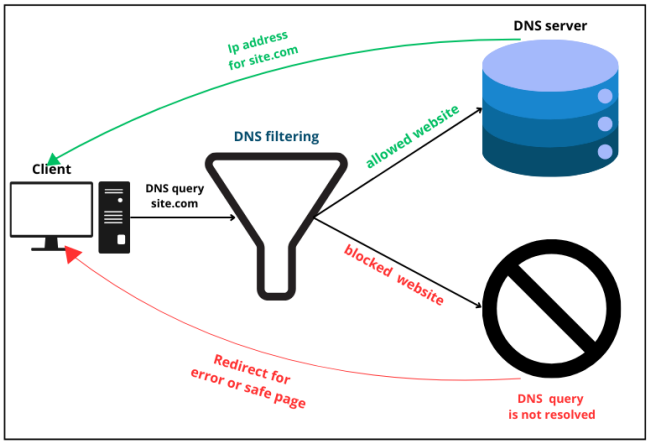


Fig. 4. DNS filtering

*3) Bypassing techniques:*

*a) Third-party DNS:* , one common method to bypass DNS filtering is by using third-party DNS resolvers. AS previously mentioned, the default DNS resolver is typically provided by Internet Service Provider (ISP). This default resolver often lacks advanced security features such as protection agains malware and cyberattacks while being more susceptible to government censorship. So users frequently switch to third-party DNS services like Google DNS (8.8.8.8) (approximately 50% of global DNS traffic). But they are ofter used as a way to bypass censorhip inflicted by local government policies. But this may not be an adequate solution to access illegal website as most of DNS services include filters for this kind of content. Third party DNS is also used to have better protection against malware, cyber security attacks but a study based on IST data in Denmark estimated that only 1.1 % to 1.5% of users employ third-party DNS resolvers for content filtering, this and the fact that DNS responses for censored domains are found to be at least two orders of magnitude more prevalent on third-party resolvers that on ISP-provided DNS, indicating that many users switch resolvers primarily to circunvent censorship and regulations.

*b) Encrypted DNS Protocols::* traditional DNS transmits queries in plain, unencrypted text, exposing the user activity and the domains being accessed. A solution to this would be to use encrypted DNS protocols such as DoH (DNS over HTTPS) or DoT (DNS over TLS).

- **DNS over HTTPS:** this protocol encrypts DNS queries using HTTPS tunneling, it does therefore offer better privacy by preventing monitoring users traffic, meanwhile preventing DNS filtering as the domain name can not longer be obtained
- **DNS over TLS:** work similarily to DoH but it uses a dedicated TLS connection instead

*c) Other Bypassing Methods:* In addition to the methods presented, user may employ alternative techniques to circumvent DNS filtering, in this paper we will not further investigate those methods, we will simply mention them.

- **Virtual Private Networks (VPN)** VPNs encrypt all internet trafic and route it through remote servers, masking the user's IP address
- **Tor Network** anonymizes traffic by routing it through multiple nodes
- **Proxy Servers**
- **Accessing the IP address** however many websites rely on virtual hosting and require domain based access

### *E. BlindTLS*

*1) Who's likely to implement it?:* Some text.
*2) Functional description:* Some text.
*3) Bypassing techniques:* Some text.

### *F. HTTPS Server Name Indication Filtering*

*1) Who's likely to implement it?:* Governments, internet service providers, and corporate networks are the primary entities that implement HTTPS SNI filtering. Authoritarian

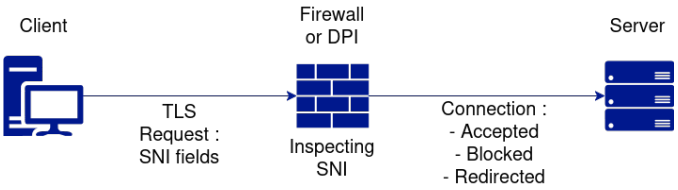


Fig. 5. SNI in TLS protocol

regimes, such as China, often use it for censorship by blocking access to specific websites, while internet service providerse and organizations may deploy it to enforce network policies, restrict access to harmful sites, or optimize bandwidth usage. Companies may also employ SNI filtering for security purposes, especially to prevent employees from accessing phishing sites or dangerous domains.

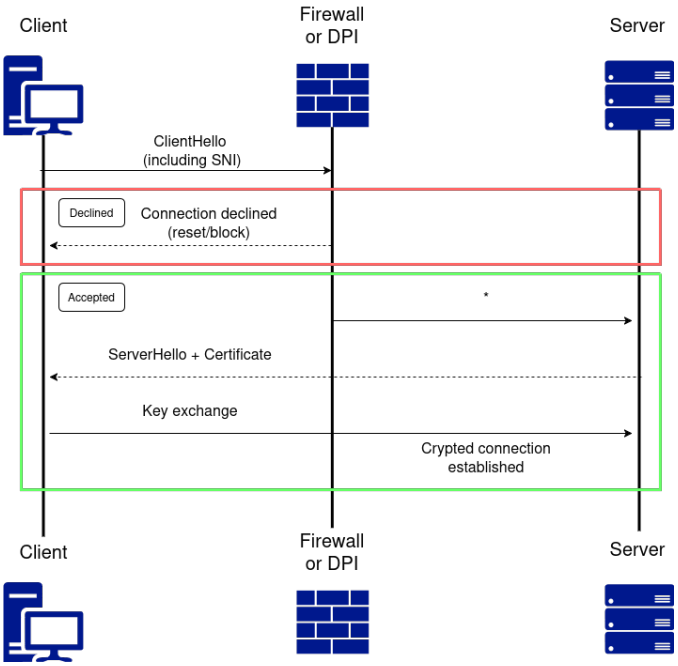


Fig. 6. Functioning of SNI Filtering

*2) Functional description:* SNI is an extension of the TLS protocol that allows a client to indicate the hostname of the server it intends to connect to during the TLS handshake. HTTPS SNI filtering operates by inspecting this plaintext field before encryption takes place. Firewalls or deep packet inspection (DPI) tools analyze the SNI field in ClientHello messages and apply filtering rules based on blacklists, whitelists, or keyword-based blocking mechanisms. Since this process occurs before full encryption, it enables network administrators to regulate traffic without decrypting the payload. However, this method has limitations, such as inability to filter connections using encrypted SNI (ESNI).

*3) Bypassing techniques:* Encrypted SNI (ESNI) and Encrypted ClientHello (ECH):One of the most effective countermeasures against SNI filtering is ESNI (Encrypted SNI). Unlike traditional SNI, ESNI encrypts the server name within the TLS handshake, preventing middleboxes and firewalls from inspecting or blocking connections based on the requested domain. TLS 1.3 handshakes that include ESNI. — TLS 1.3 introduced ECH, which encrypts the entire ClientHello message, including the SNI field, making traditional filtering ineffective.—

Bypassing with backward compatibility enforcement : SNI filtering can be bypass thanks to compatibility concerns. Some systems allow fallback to older TLS versions. Attackers can exploit this by forcing a downgrade attack, causing the client to negotiate an older protocol version where SNI filtering is either less effective or absent. This technique is particularly useful in environments where TLS 1.3 is partially adopted but not universally enforced.

Bypassing on shared certificate hosting : A significant limitation of SNI filtering arises when multiple domains share the same TLS certificate. Many Content Delivery Networks (CDNs) and cloud service providers host numerous domains under a single certificate, making it difficult for filtering entities to block one specific domain without affecting others. Users can access restricted domains by requesting a different but valid hostname that resolves to the same IP address and TLS certificate when a common certificate exists between multiples domains. This method is particularly effective when CDNs serve both blocked and permitted domains, especially when forbidding access to all ip addresses could prevent users from accessing services that are not concerned.

### G. Disuasion techniques

*1) Remote surveillance:* Some text.

*2) Social monitoring:* Some text.

Place here the conclusion table with some text.

## IV. Methodology

Some text.

## V. Results

Some text.

## VI. Conclusion

Some text.

## Acknowledgment

We wish to thank Thibault Cholez, our tutor, for the help he provided to us during researches and practical experiments, and also TELECOM Nancy for allowing us to make this paper.

## References

[1] S. Satija and R. Chatterjee, "Blindtls: Circumventing tls-based https censorship," in *Proceedings of the ACM SIGCOMM 2021 Workshop on Free and Open Communications on the Internet*, ser. FOCI '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 43–49. [Online]. Available: https://doi.org/10.1145/3473604.3474564

[2] W. M. Shbair, T. Cholez, A. Goichot, and I. Chrisment, "Efficiently bypassing sni-based https filtering," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 990–995.

[3] Z. Chai, A. Ghafari, and A. Houmansadr, "On the importance of Encrypted-SNI (ESNI) to censorship circumvention," in *9th USENIX Workshop on Free and Open Communications on the Internet (FOCI 19)*. Santa Clara, CA: USENIX Association, Aug. 2019. [Online]. Available: https://www.usenix.org/conference/foci19/presentation/chai

[4] M. Fejrskov, E. Vasilomanolakis, and J. M. Pedersen, "A study on the use of 3rd party dns resolvers for malware filtering or censorship circumvention," in *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer, 2022, pp. 109–125.

[5] W. Stol, H. Kaspersen, J. Kerstens, E. Leukfeldt, and A. Lodder, "Governmental filtering of websites: The dutch case," *Computer Law Security Review*, vol. 25, no. 3, pp. 251–262, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0267364909000582

[6] W. M. Shbair, T. Cholez, A. Goichot, and I. Chrisment, "Efficiently Bypassing SNI-based HTTPS Filtering," in *IFIP/IEEE International Symposium on Integrated Network Management (IM 2015)*, Ottawa, Canada, May 2015, pp. 990–995. [Online]. Available: https://inria.hal.science/hal-01202712

[7] S. Satija and R. Chatterjee, "Blindtls: Circumventing tls-based https censorship," 08 2021, pp. 43–49.

[8] S. A. Samarakoon, "Bypassing content-based internet packages with an ssl/tls tunnel, sni spoofing, and dns spoofing," 2022. [Online]. Available: https://arxiv.org/abs/2212.05447