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. [5]

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. [6]

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.

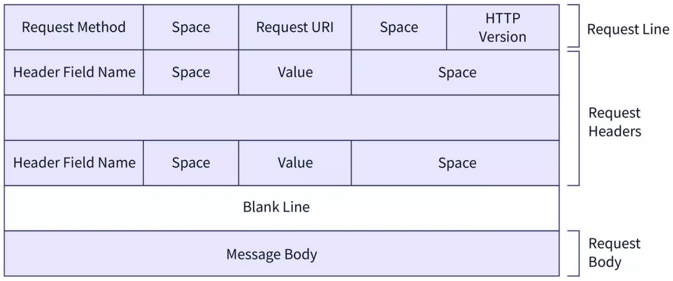

Fig. 1. The HTTP model.

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. [7] 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. [6]

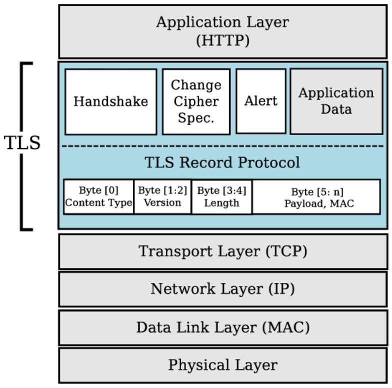

Fig. 2. The TLS model.

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?: Some text.

2) Functional description: Some text.

3) Bypassing techniques: Some text.

D. Hybrid IP and Proxy Filtering

1) Who's likely to implement it?: Some text.

2) Functional description: Some text.

3) Bypassing techniques: Some text.

E. DNS Deregistration

1) Who's likely to implement it?: Some text.

2) Functional description: Some text.

3) Bypassing techniques: Some text.

F. BlindTLS

1) Who's likely to implement it?: Some text.

2) Functional description: Some text.

3) Bypassing techniques: Some text.

G. HTTPS SNI Filtering

1) Who's likely to implement it?: Some text.

2) Functional description: Some text.

3) Bypassing techniques: Some text.

H. 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] Fejrskov, M., Vasilomanolakis, E., Pedersen, J.M. (2022). **A Study on the Use of 3rd Party DNS Resolvers for Malware Filtering or Censorship Circumvention**. In: Meng, W., Fischer-Hübner, S., Jensen, C.D. (eds) *ICT Systems Security and Privacy Protection*. SEC 2022. IFIP Advances in Information and Communication Technology, vol 648. Springer, Cham. https://doi.org/10.1007/978-3-031-06975-8$_7$

[2] Sambhav Satija and Rahul Chatterjee. (2021). **BlindTLS: Circumventing TLS-based HTTPS censorship**. In *Proceedings of the ACM SIGCOMM 2021 Workshop on Free and Open Communications on the Internet (FOCI '21)*. Association for Computing Machinery, New York, NY, USA, 43–49. https://doi.org/10.1145/3473604.3474564

[3] W.Ph. Stol, H.K.W. Kaspersen, J. Kerstens, E.R. Leukfeldt, A.R. Lodder. (2009). **Governmental filtering of websites: The Dutch case**. *Computer Law & Security Review*, Volume 25, Issue 3, Pages 251-262. ISSN 0267-3649. https://doi.org/10.1016/j.clsr.2009.03.002

[4] Wazen M. Shbair, Thibault Cholez, Antoine Goichot, Isabelle Chrisment. (2015). **Efficiently Bypassing SNI-based HTTPS Filtering**. In *IFIP/IEEE International Symposium on Integrated Network Management (IM 2015)*, Ottawa, Canada. pp.990-995. https://doi.org/10.1109/INM.2015.7140423

[5] Zimo Chai, Amirhossein Ghafari, and Amir Houmansadr. (2019). **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. https://www.usenix.org/conference/foci19/presentation/chai

[6] Ronald Deibert, John Palfrey, Rafal Rohozinski, Jonathan L. Zittrain (eds). (2008). **Access Denied: The Practice and Policy of Global Internet Filtering**. The MIT Press. DOI: https://doi.org/10.7551/mitpress/7617.001.0001. ISBN (electronic): 9780262255998.

[7] KARTHIKEYAN, V. K. T. **Web Content Filtering Techniques: A Survey.** International Journal of Computer Science Engineering Technology (IJCSET), 2014, vol. 5, no 03, p. 203-208.