Unveiling the Nexus: Harnessing IoT Ecosystems for Evading Internet Censorship

Patrick Tser Jern Kon Yining Shi Wyatt Ashley

University of Michigan

{eddshi, patkon, wyattash}@umich.edu

1 Introducttion

In the ever-expanding landscape of technology, Internet censorship by sovereign states continues to evolve. This dynamic presents a constant challenge akin to what many in the field call an endless "cat-and-mouse" game, where censors adapt their tactics in tandem with technological advancements, while those seeking to circumvent censorship endeavor to stay ahead [4]. Despite persistent efforts to safeguard the flow of free and unbiased information, including the risks of physical harm, the struggle persists[4] [12] [9]. However, concurrently, the emergence of Internet of Things (IoT) devices [8] within Autonomous Systems (AS) introduces a new dimension, offering vast untapped computational resources on a global scale. This proliferation not only raises concerns about potential misuse, as exemplified by instances such as the Mirai Botnet but also presents an opportunity for positive utilization, which this paper aims to explore and propose.

With this, our study aims to investigate IoT ecosystems as a tool for evading censorship. Through a comprehensive characterization and fingerprinting of IoT ecosystems at both the Application and Kernel layers [7], we seek to differentiate them from traditional computing devices such as Personal Computers (PCs) or Server Centers. Leveraging this understanding, we explore avenues to harness the distributed nature and diverseness of IoT devices to establish a federated circumvention system. This research underscores the need for innovative approaches to address evolving censorship challenges in the digital age.

To do so, this paper will be divided into three main sections:

1. **Potential fingerprinting methods might be implemented by the adversaries.** Despite having no evidence found yet to presume that state-sponsored Internet Censorship operations are targeting IoT devices explicitly, it is vital to preemptively establish potential attack models to consider to develop counter-measurements. Hence, this section will be dedicated to exploring past works that might enlighten such methods. [2] [3]

2. **Design of the IoT-based distributed network.** In this section, we will introduce our new method of internet censorship circumvention, utilizing the vastly untapped computational power of IoT devices. In addition, the sheer number of such devices in existence [10] also offers a unique nature with the difficulty of tracking activities.

3. **Methods of counter-measurement.** With the established premise from the first section, we will delve into developing algorithms to achieve mimic and/or obfuscation when data is transmitted through our distributed system, and prove the correctness of such algorithms.

2 Research Questions

The goal we try to achieve also raises questions:

1. How may censors (on-path attackers) detect and differentiate IoT devices' traffic from regular Personal Computers' (PCs') traffic?

2. How can censors identify devices as IoT devices through PROBING, and how can we design a system that prevents such identifiability?

3. What is the recent trend of new methods of censorship discovered that are deployed by nation-states.

4. Is it possible to achieve unreadability of traffic as a means to circumvention in lieu of obfuscation?

5. Is it possible to dissect transmission packets, distributedly send them to a network of IoT devices, and aggregate them at the end to form a flow of connection with the destination server?

6. Will the proposed system lower the risk of participants?

3 Proposed Frameworks

3.1 Unreadable file algorithm

We define a file as "unreadable" when it cannot be decoded with conventional decoders, such as ASCII, and can be arbitrarily treated as meaningless data by the man-in-the-middle (MITM), who is the censor in our context. The proposed algorithm above attempts to detect ASCII characters in a file (such as an HTTP request), and separate the files at bit level, attempting to render the byte-encoded ASCII characters "unreadable" and cannot be reassembled at a feasible time complexity (linear).

Algorithm 1 Process a file and dissect packets

```
function ProcessFile(file)
    IoT_proxy_list{1...N}          ▷ Let IoT_proxy_list contain the list of all IoT proxy devices
    bytes_read ← {}                ▷ Let bytes_read be 0-indexed array in bytes
    id ← rand()
    while not end of file do
        bytes_read.APPEND(file.next_byte)
    end while
    IsReadable(bytes_read, id, IoT_proxy_list)
    return
end function
```

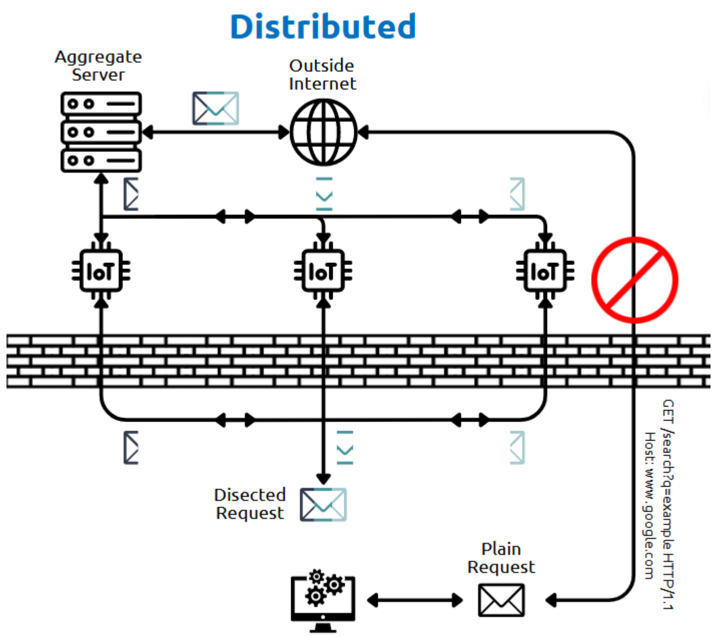
Algorithm 2 Check if the byte sequence is readable and manipulate accordingly

```
function isReadable(bytes_read, id, IoT_proxy_list)
    packet_to_send ← {}                    ▷ Let packet_to_send be 0-indexed array of bytes to send
    never_dissected ← true
    for each 1 byte in bytes_read, index i do
        if bytes_read[i] is ASCII (ISO-8859-1) then
            never_dissected ← false

            packet_to_send.APPEND(i-th byte's FIRST bit)              ▷ Cut off the readable ASCII byte
            packet_to_send.INSERT_FRONT(len(packet_to_send)- a number with fixed-size of 2-byte)   ▷ Size of actual payload
            packet_to_send.INSERT_FRONT(len(id)- a number with fixed-size of 2-byte + id)   ▷ Size of session ID
            bytes_read.POP(i-th byte's FIRST bit)
            RoundRobinSend(packet_to_send, IoT_proxy_list)   ▷ Use naive Round-robin algorithm to select IoT proxy
            packet_to_send ← {}                                        ▷ Clear the packet array
        end if
        packet_to_send.APPEND(bytes_read[i])
        bytes_read.POP(bytes_read[i])
    end for
    if never_dissected == true then                             ▷ Entire file is not human-readable
        packet_to_send.INSERT_FRONT(len(packet_to_send)- a number with fixed-size of 2-byte)
        packet_to_send.INSERT_FRONT(len(id)- a number with fixed-size of 2-byte + id)
        RoundRobinSend(packet_to_send, IoT_proxy_list)
    end if
    RoundRobinSend(packet_to_send, IoT_proxy_list)
    return
end function
```

3.2 Distributed transmission through IoT Networks

We propose a robust system, under which we control a set of IoT devices with low computation power. With the "unreadability" algorithm, we may separate the files into an arbitrary amount of pieces and send them to the IoT devices (assuming no TLS). During the process of transmission, we assume that the censor will have access to the content of each packet and will inspect it. However, the censor would not be able to determine if the content is prohibited with a correct implementation of our algorithm under a black-list censorship system. With success, we assume that the packets will arrive at the controlled set of IoT devices at a controlled order with our implementation of the Round-Robin algorithm. With the byte-order book-kept at the IoT sub-network, the file will then be sent to a proxy server that can be implemented on a separate IoT device to unify the IP address, avoiding confusion from the destination server. With a successful design of our proposed framework, it is assumed that a circumvented connection can be established.



4 References

References

[1] Bock, K. (2022). *Automating the Discovery of Censorship Evasion Strategies*. PhD thesis, University of Maryland, College Park.

[2] Herwig, S., Harvey, K., Hughey, G., Roberts, R., and Levin, D. (2019). Measurement and analysis of hajime, a peer-to-peer iot botnet. *Proceedings 2019 Network and Distributed System Security Symposium*.

[3] Hong, J., Levy, A., Riliskis, L., and Levis, P. (2018). Don't talk unless i say so! securing the

internet of things with default-off networking. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 117–128.

[4] Houmansadr, A., Brubaker, C., and Shmatikov, V. (2013). The parrot is dead: Observing unobservable network communications. pages 65–79.

[5] Houmansadr, A., Riedl, T., Borisov, N., and Singer, A. (2012). I want my voice to be heard: Ip over voice-over-ip for censorship circumvention.

[6] Jia, W., Eichenhofer, J., Wang, L., and Mittal, P. (2023). Voiceover: Censorship-circumventing protocol tunnels with generative modeling.

[7] Kostas, K., Just, M., and Lones, M. A. (2022). Iotdevid: A behavior-based device identification method for the iot. *IEEE Internet of Things Journal*, 9(23):23741–23749.

[8] Kumar, S., Tiwari, P., and Zymbler, M. (2019). Internet of things is a revolutionary approach for future technology enhancement: a review. *Journal of Big Data*, 6.

[9] Makhdoom, I., Abolhasan, M., and Lipman, J. (2022). A comprehensive survey of covert communication techniques, limitations and future challenges. *Computers & Security*, 120:102784.

[10] Ronen, E., Shamir, A., Weingarten, A.-O., and O'Flynn, C. (2017). Iot goes nuclear: Creating a zigbee chain reaction. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 195–212.

[11] Sun, Z. and Shmatikov, V. (2023). Telepath: A minecraft-based covert communication system. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2223–2237, Los Alamitos, CA, USA. IEEE Computer Society.

[12] Wu, M., Sippe, J., Sivakumar, D., and et al (2023). How the great firewall of china detects and blocks fully encrypted traffic. *Proceedings of the USENIX Security Symposium.*

[13] Wustrow, E., Wolchok, S., Goldberg, I., and Halderman, J. A. (2011). Telex: Anticensorship in the network infrastructure. In *20th USENIX Security Symposium (USENIX Security 11)*, San Francisco, CA. USENIX Association.

[14] Xue, D. and Ensafi, R. (2023). The use of push notification in censorship circumvention.

[15] Xue, D., Ramesh, R., Jain, A., Kallitsis, M., Halderman, J. A., Crandall, J. R., and Ensafi, R. (2022). Openvpn is open to vpn fingerprinting. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 483–500.