

Article

Introducing the Slowloris E-DoS Attack: A Threat Arising from Vulnerabilities in the FTP and SSH Protocols

Nikola Gavric , Guru Bhandari and Andrii Shalaginov

School of Technology and Innovation, Kristiania University of Applied Sciences, Kirkegata 24, 0153 Oslo, Norway; guru.bhandari@kristiania.no (G.B.); andrii.shalaginov@kristiania.no (A.S.)

* Correspondence: nikola.gavric@kristiania.no

Abstract

Slowloris is a well-known application-layer Denial of Service (DoS) attack that is challenging to detect due to its low-rate nature, allowing it to blend with legitimate traffic and remain unnoticed. Our hypothesis is that deliberate prolongation of the pre-authentication stage in stateful protocols induces unnecessary CPU utilization. In this study, we repurpose Slowloris as an energy-oriented (E-DoS) attack that exploits pre-authentication statefulness of the most prevalent remote access protocols, the Secure Shell Protocol (SSH) and File Transfer Protocol (FTP). We employ a Raspberry Pi-based experimental setup with different software implementations of the mentioned protocols to validate our hypothesis. Our experiments confirm the susceptibility of SSH and FTP to Slowloris E-DoS attacks, and we quantify the consequential impact on power consumption. We find that the Slowloris E-DoS attack exhibits an asymmetrical nature, causing a disproportionate computational demand on victim systems compared to the resources invested by the attacker. The results of this study indicate that battery-powered single-board computers (SBCs) are critically affected by these attacks due to their limited power availability. This research demonstrates the importance of understanding and mitigating Slowloris E-DoS vulnerabilities in the SSH and FTP protocols, offering valuable insights for enhancing security measures. Our findings show that millions of SBCs worldwide may be at risk and highlight a deeper structural weakness: the stateful design of widely deployed protocols can turn service availability into an energy liability. This systemic risk extends beyond SSH and FTP, with implications for IoT devices and backends that depend on stateful communication protocols.

Keywords: Raspberry Pi Security; protocol security; IoT power consumption; low-rate DoS; E-DoS attack; SSH; FTP; slowloris attack



Academic Editors: Jordi Mongay Batalla, Pascal Lorenz and Ioannis Chatzigiannakis

Received: 18 March 2026

Revised: 9 April 2026

Accepted: 15 April 2026

Published: 17 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Denial of Service (DoS) attacks aim to degrade the quality of the targeted service or cause a complete outage by depleting the crucial resources required to run the service. These resources include network and system aspects, such as bandwidth, number of connection slots, CPU usage, and available memory. A distributed variant of DoS attacks, where the attacks are simultaneously launched from many sources, is called Distributed Denial of Service (DDoS) and presents an even more severe threat than the DoS attacks. In the context of the expanding Internet of Things (IoT) paradigm, which refers to the interconnected network of physical devices embedded with sensors and software for data exchange, the impact of such attacks can disrupt network functionality and degrade service quality [1].

As IoT devices become more integrated into critical infrastructure, the potential for DoS and DDoS attacks increases significantly due to the high level of connectivity involved [2].

Some of the most challenging-to-detect DoS attacks are low-rate due to their ability to blend with legitimate traffic and remain under the radar on the network level [3]. Among such attacks is an application layer attack called Slowloris [4], which creates and maintains many connections with the targeted service indefinitely. The attack initiates incomplete requests and periodically sends keep-alive headers to prevent the server from releasing the connections. Therefore, Slowloris keeps the victim's threads assigned to the connections busy, making the server unable to handle new connections once it runs out of threads. Once the server cannot process new connections, there is a service outage for legitimate users. Slowloris is known for its effectiveness and stealthy nature, allowing it to go undetected by many traditional defense mechanisms [5]. Such properties form an ideal attack vector for the potential adversaries.

There is a grey area between the legitimate operating state and the service outage state of the targeted service caused by DoS attacks. Inside that area is an ongoing DoS attack where the service is still available. If the attacker intentionally keeps the attack inside this area, an Energy-oriented DoS (E-DoS) attack may occur [6]. The goal of E-DoS attacks is to maximize the victim's power consumption, which in the case of IoT may last indefinitely, as the attacks are often discovered only after service outage [7]. In the case of targeting a battery-powered device, such an attack would result in an outage of all services once the battery is drained. Additionally, the increased frequency of battery replacements significantly raises operational costs, which may occur up to 50 times more often, as demonstrated in Narrowband-IoT devices by a decrease in battery lifespan from 17 years to just four months [8].

In this study, we exploit and analyze an E-DoS vulnerability in the ubiquitous remote access protocols called Secure Shell Protocol (SSH) and File Transfer Protocol (FTP). According to Shodan [9], there are currently over 26 million devices on the public internet across the globe using SSH protocol and 3.3 million using FTP. SSH and FTP can serve multiple users simultaneously, and they rely on username and password authentication by default, which makes them an ideal target for Slowloris attacks. To exploit this, we set up corresponding SSH and FTP servers on a community-accepted, widely available off-the-shelf device, Raspberry Pi 4B.

The Raspberry Pi is a series of small, single-board computers (SBC) developed by the Raspberry Pi Foundation [10]. Raspberry Pi's appeal lies in its affordability, compact design, General Purpose Input/Output pins, low power demand, and extensive software support. There are also many battery packs available for it to allow for remote deployment. Therefore, Raspberry Pi finds applications across various fields and industries. Its compact form factor and low power consumption make it ideal for embedded systems and applications, where it can monitor environmental data, control appliances, or facilitate communication between devices [11]. Raspberry Pi commonly utilizes the Raspbian operating system, a Debian-based Linux distribution optimized for the platform. It is also important to mention Raspberry Pi security concerns, as it is vulnerable to many threats, among which our study focuses on DoS attacks. The Raspbian operating system lacks built-in virus protection and an intrusion detection system (IDS), making it susceptible to DoS attacks [12]. Raspberry Pi sales increase annually as nearly 50 million devices were manufactured up to mid-2023 (Figure 1). In 2023, the manufacturer was set to reach consistent production of a million devices a month [13]. Hence, we opted for Raspberry Pi in this study to experimentally validate our hypothesis, as the results may impact millions of users as well as other Linux-based SBCs using the same software.

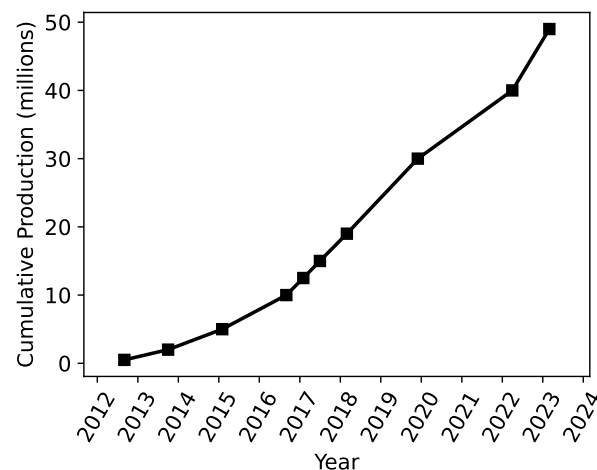


Figure 1. Cumulative Raspberry Pi production numbers [14–17].

In this paper, we repurpose Slowloris as an E-DoS attack targeting the most common remote access protocols, and we explore its impact on the victim’s power consumption. Our study encompasses a widely adopted hardware device, the Raspberry Pi, used by millions worldwide, utilizing the ubiquitous software implementations of the SSH and FTP protocols. Therefore, our study is representative of real-world scenarios affecting a substantial user base. The main contributions of this paper are the following:

- We identify an E-DoS threat that potentially affects millions by repurposing the Slowloris DoS attack. Furthermore, we empirically showcase and analyze this difficult-to-detect threat that arises from the vulnerabilities in the FTP and SSH application-layer communication protocols.
- We experimentally validate the hypothesis using a Raspberry Pi SBC and quantify the identified threat’s effects on the victim’s power consumption.
- Based on the obtained results, we propose mitigation strategies urging both the industry and the research community to investigate E-DoS attacks further.

2. Background and Related Work

This section explains common terminologies and discusses the related work covering DoS, DDoS and E-DoS attacks, Slowloris attacks, FTP and SSH protocols, and their vulnerabilities. Additionally, we discuss the characteristics and prevalence of the hardware and software covered by this study. The need to address DDoS attacks against IoT devices is critically emphasized in recent research. IoT devices, due to their limited computational resources, are particularly vulnerable to DDoS attacks, making effective defense mechanisms essential [18]. Furthermore, detection of attacks against authentication protocols is still an open issue [19]. These statements can be further extended to E-DoS attacks alike since they can be interpreted as limited-intensity DoS and DDoS attacks.

2.1. E-DoS Attacks

E-DoS attacks’ effects in a cloud environment were demonstrated in [6], where, in addition to power consumption, these attacks caused the occupation of additional cloud resources, resulting in increased operational costs. The adverse effects of network-based DoS attacks causing fraudulent energy consumption were explored in [20]. The attacks affected energy costs, increased greenhouse gas emissions, and caused increased operating temperatures. Increased temperatures reduce the chip’s lifetime determinism in computational operations and require more power to dissipate the heat. In the case of batteries, high temperature also reduces capacity and lifespan. The authors also suggested SSH as a

potential threat that could be exploited to cause high CPU usage while pointing out that the CPU could be the biggest driver of power consumption. The E-DoS effects regarding power consumption in common IoT devices were assessed in [21]. The IoT devices exposed to high-rate DoS attacks exhibited high power consumption, demonstrating a linear dependency of power consumption as a function of the DoS attack rate. Al-Haidari et al. demonstrate that DoS attacks aiming to maximize economical damage can cause cloud service costs to surge up to 15 times higher than normal, by exploiting the cloud's scalability to generate excessive resource usage and financial strain [22]. In this paper, we explore the relation between the DoS attack rate and power consumption of a Raspberry Pi 4 model B while targeting the FTP and SSH protocols with a low-rate attack.

In spite of the authors addressing the E-DoS threats and identifying some vulnerabilities, further research is needed to explore this topic comprehensively as we argue that E-DoS vulnerabilities are more prevalent than previously thought. Compared with prior E-DoS studies, our work focuses on a different attack surface. Existing literature has examined E-DoS attacks in cloud environments, network-based attacks that increase energy cost, and high-rate DDoS/E-DDoS effects on IoT and smart-home devices. In contrast, this paper studies a low-rate application-layer attack that exploits pre-authentication statefulness in SSH and FTP services. We focus on shedding light on a vulnerability potentially affecting millions of devices worldwide (Table 1). We argue that a difficult-to-detect low-rate DoS attack is sufficient to exploit the identified vulnerability, thus posing a potent threat.

Table 1. Prevalence of online devices (rounded) with selected software according to Shodan.

Protocol/Software	Number of Devices
SSH	26,000,000
FTP	3,350,000
proftpd	815,000
vsftpd	538,000
OpenSSH	54,000
Dropbear	365,000
Raspbian	154,000
Raspbian and SSH	81,000
Raspbian and FTP	200

It should be noted that Shodan may not always accurately identify the specific software implementing SSH and FTP protocols, potentially resulting in lower numbers displayed in Table 1 than the actual prevalence. Considering the widespread deployment of protocols such as SSH and FTP, understanding their specific vulnerabilities is crucial in the context of E-DoS attacks. Therefore, a thorough examination of these protocols and their associated vulnerabilities is essential for a comprehensive understanding of the E-DoS threats.

2.2. SSH

Commonly known as SSH, Secure Shell Protocol is a protocol that provides secure access to remote systems over an unsecured network. The protocol was designed to replace previous less secure protocols, such as telnet. It was formalized in 2006 via RFCs 4250 [23], 4252 [24], 4253 [25], and 4254 [26]. A Request for Comments (RFC) is a publication in a series from the principal technical development and standards-setting bodies for the Internet, primarily the Internet Engineering Task Force (IETF). SSH typically operates using port 22, and its presence can be easily discovered using widely available free tools such

as nmap [27]. DoS attacks against SSH have been known for a while, and there are many suggested measures when configuring the SSH servers to protect against them [28]. These include logging connection attempts and limiting the number of simultaneous connections as well as connections per user. Additionally, it is recommended to introduce a short timeout for the negotiation phase, during which the server and the client agree upon the encryption algorithm. Proposed network IDS solutions usually do not take protocol-specific behavior into consideration, thus disregarding idle unauthenticated users as a potential threat [29].

In the initiation of an SSH connection, a preliminary handshake occurs. This handshake entails the server's initial effort to establish key exchange (kex) algorithms to define the encryption method for the connection. Subsequently, agreement on the host key and cipher algorithms is reached, followed by exchanging host keys. This means that every connection is encrypted, even before authentication takes place, implying that each new connection consumes additional CPU time because of the encryption process. Without a password parameter in the connection, the server defaults to public-key authentication. The connection is terminated if no key is provided (along with no password parameter). Over 65% of SSH servers on the public internet allow password authentication [30], which is the least secure option due to susceptibility to brute-force attacks [31]. In our use case, we also examine attacks against an SSH server that only allows key access to see whether it will drop the connections or keep them alive, as maintaining the connections is crucial for the E-DoS attacks to work. In addition to SSH, we also examine FTP, another prevalent stateful communication protocol with its own set of vulnerabilities relevant to our study.

2.3. FTP

FTP is a simple file transfer protocol that was developed in 1971 and formalized in RFC 114 [32]. FTP operates in a client-server topology where clients send requests to the server and send or receive files. TCP usually establishes two connections, one for control and another one for data transfer, typically utilizing ports 21 and 20, respectively. Upon successful authentication, the control connection is established and the client receives a list of directories and file names. Papers concerning FTP and SSH security typically discuss authentication exploits [33] and brute force attacks [31,34], while omitting DoS and DDoS.

Similarly to SSH, FTP is a stateful protocol that requires a persistent connection and allows serving multiple users simultaneously, making it a good candidate for the Slowloris vulnerability. Unlike SSH, FTP traffic is not encrypted, thus requiring less CPU usage to maintain connections. While FTP does not support key-based authentication, there are more secure versions of the protocol such as FTP Secure (FTPS) and SSH File Transfer Protocol (SFTP). However, according to Shodan, only around 36 thousand devices host SFTP servers, and less than 2000 host FTPS, so we omitted these protocols from our study. Building on the vulnerabilities of SSH and FTP, we proceed to examine vulnerable hardware platforms that commonly use these protocols.

Both FTP and SSH protocols are stateful by design, requiring servers to allocate resources before authentication is completed. This asymmetry between client and server work creates a potential attack surface, which we investigate in the remainder of the paper.

2.4. Raspberry Pi

Raspberry Pi is a massively used device serving a wide range of applications. However, by default, it is vulnerable to malware, man in the middle, and DoS attacks [12]. Raspberry Pi hardware is designed in such a way as to reduce production cost rather than increase security. The device is also physically insecure, as it is easy to manipulate the environment if there is physical access. Moreover, default usernames, passwords, and insecure services

pose an additional security challenge [35]. Overall, in the default environment, Raspberry Pi presents a threat to the entire network in which it operates. Although SSH is disabled by default on Raspberry Pi, it is still a ubiquitous way of access. Raspberry Pi 4 model B utilizes a Quad-core Cortex-A72 (ARM v8) 64-bit processor operating at 1.8 GHz frequency [36]. In our study, we experimentally validated the hypothesis on a Raspberry Pi. However, our study is not limited to the chosen device since the same principles apply to numerous Linux-based SBCs utilizing the same protocols and software implementations.

Prior work has also examined Raspberry Pi from the perspective of power modeling. Kaup et al. [37] measured an old Raspberry Pi Model B and showed that its power consumption can be approximated using CPU and network utilization, reporting a clear linear dependency between CPU load and power draw. While their work was not security-oriented and focused on normal operating conditions rather than attack traffic, it provides useful background for interpreting how increased computational and networking activity can translate into increased energy consumption on Raspberry Pi platforms.

Considering Raspberry Pi's susceptibility to various attacks, it faces an even greater challenge defending against stealthy threats like the Slowloris attack. This warrants a detailed examination of how Slowloris can exploit the aforementioned vulnerabilities in such devices and be repurposed as an E-DoS attack.

2.5. The Slowloris Attack

Slowloris is a low-rate DoS attack that has traditionally been challenging to detect due to its ability to blend with legitimate traffic and remain undetectable. The attack consists of two main parameters: the number of open connections and the period between sending the keep-alive headers, commonly known as the sleep parameter. Many authors explored ways to combat Slowloris attacks [38–40], but these detection methods are often difficult to implement. Slowloris attack is not known for consuming many system resources, and in cases of attacks against HTTP(S), its impact is barely noticeable [41]. However, handling SSH and FTP requests typically requires more CPU usage than HTTP(S) due to the need for maintaining session persistence and connection states. This study examines the relation between the set Slowloris parameters and measured power consumption to assess their importance in conducting E-DoS attacks. The goal is to emphasize the severity of the threat by showing that low-rate DoS attacks can cause substantial power consumption while remaining undetectable.

To our knowledge, nobody has considered repurposing Slowloris as an E-DoS attack. In our study, we identify two application-layer protocols where Slowloris' impact on the victim's hardware substantially differs from what was previously known. We argue that Slowloris attacks against stateful protocols cause an increase in CPU usage and consequently power consumption, turning this DoS attack into an E-DoS one.

3. Methodology and Experiment Setup

To assess the impact of E-DoS attacks against SSH and FTP, we launch Slowloris attacks with different parameters while keeping track of the upload speed on the attacker side and power consumption on the victim's end. Figure 2 shows the threat model used in the experimental setup in which we used Raspberry Pi 4 model B as a victim device, and Figure 3 provides a detailed picture of the actual test setup. Linux-based SBCs are often utilized in various applications, including battery-powered ones in IoT, where they typically do not operate at high CPU usage for prolonged periods.

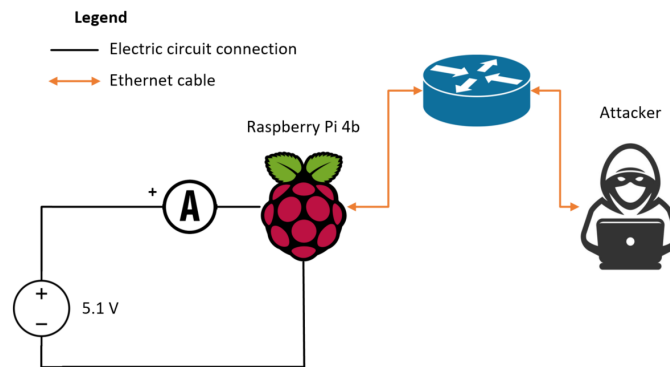


Figure 2. Threat model. The attacker accesses Raspberry Pi over Ethernet through a router, while the victim device is powered by a 5.1 V source and monitored with an ammeter.

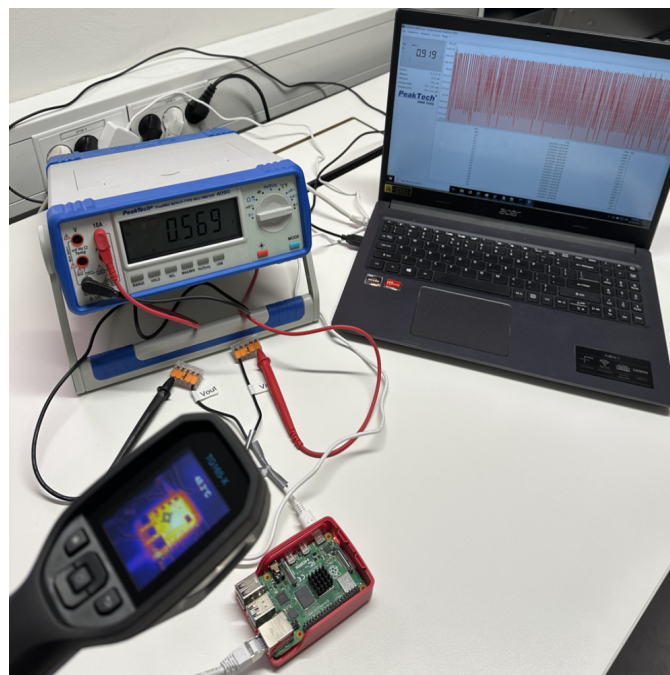


Figure 3. The experimental setup showing a Raspberry Pi connected to a source through a digital multimeter. The laptop in the background displays the recorded measurement logs, while the thermal imaging camera used for temperature observations is visible in the foreground.

3.1. Experimental Setup and Measurements

Raspberry Pi utilizes Raspbian OS with default settings, assuming default package configurations on Raspbian unless explicitly stated otherwise. We tested two software implementations of each SSH (OpenSSH [42], Dropbear [43]) and FTP (vsftpd [44], proftpd [45]). While conducting the attacks, we measured the current flow using a digital multimeter PeakTech 4090. The multimeter exhibits ± 15 mA absolute and $\pm 0.8\%$ relative error while measuring DC current. We multiplied the measured current by the constant voltage of 5.1 Volts to obtain the power consumption. The attacker PC and Raspberry Pi are connected to a router via a 1 Gbps Ethernet connection. Additionally, we used a thermal imaging camera (FLIR TG165-X) to assess the thermal impact on the device. By default, Raspberry Pi arrives without a CPU heat sink, but we glued one to ours to avoid possible damage.

3.2. Experiments

To evaluate the impact of the Slowloris attack, we conducted two types of experiments for each of the aforementioned implementations of the FTP and SSH protocols. One

for external observations only and the other with the addition of the internal, on-device measurements, where we additionally measured CPU usage remotely. In both cases, the attacks lasted for five minutes each, for every tuple of the parameters. The evaluated attack parameter space consisted of nine socket-count settings (1, 10, 100, 200, 300, 400, 500, 1000, and 2000) and four sleep values (0, 1, 2, and 5 s), selected to represent a broad range of Slowloris attack intensities, resulting in 36 attack configurations per experiment. After each five-minute attack session, we left the Raspberry Pi idle for five minutes to allow it to cool down before the next run. To obtain the baseline power consumption, we allowed regular and undisrupted device operation, during which no attacks occurred and only the necessary Raspbian system processes were active.

3.2.1. External-Only Observations

In this series of experiments, we measured the victim's current and the attacker's upload speed, using average measured values as a reference. To cause unnoticeable E-DoS, we aimed to preserve service availability while maximizing power consumption. Therefore, we were occasionally trying to access the server to assess service availability throughout the attack process. Moreover, we conducted attacks under three different SSH modes and two FTP access options. Specifically, we tested SSH and FTP access with password authentication enabled, SSH access restricted to key-based authentication only, and SSH and FTP access configured with an IP whitelist, allowing only a predefined set of IP addresses to connect. More precisely, in the case of the whitelist configuration, we attempt the attack from an IP address that is not allowed. We took a thermal image of the Raspberry Pi during the final minute of each attack run.

3.2.2. Mixed Observations

In the second series of experiments, we repeated the attacks using the same parameter combinations as in the external-only observations, but additionally collected internal measurements from the victim device. Specifically, we accessed the Raspberry Pi remotely via OpenSSH and used `psutil` [46] to monitor CPU utilization in real time and to record average CPU usage during each attack run. OpenSSH was selected for remote access because it provided stable operation throughout the measurements and allowed us to keep the monitoring procedure consistent across all tests. The purpose of this experiment was to quantify the contribution of CPU activity to the observed increase in power consumption. Since the attacks considered in this study are low-rate, we expected CPU usage rather than network throughput on the victim side to be the primary driver of the additional energy demand, in line with prior Raspberry Pi power-modeling work [37]. For this reason, internal CPU measurements were used to validate the hypothesis that increased power consumption during the attacks is primarily associated with sustained CPU load.

We conducted the mixed-observation experiments separately from the external-only observations to avoid influencing the primary power measurements with the monitoring software and the remote access session. Thus, the external-only observations were used to assess the attack's effect on power consumption with minimal interference, whereas the mixed observations were used to explain the same effect in terms of CPU utilization. Although the thermal images qualitatively indicate increased CPU activity, the additional CPU measurements were necessary to quantify its extent and to identify the attack configurations that maximize CPU load and, consequently, power consumption.

4. Results

This section presents the relation between the Slowloris attack's parameters, and the observed measurements. We begin with external measurements and by measuring the

baseline power consumption while no malicious activity occurs on the victim device and proceed with conducting the attacks. Based on the current measurement of 0.468 Amperes, we estimate the baseline power consumption to be around 2.39 watts. To emphasize the increase in power consumption in our graphs, we subtracted the baseline power consumption from the measurements obtained through further experiments. This allowed us to clearly illustrate the net impact of the Slowloris attacks on power usage.

In the first iteration, we used password authentication on the victim-end SSH and FTP servers. Subsequently, we tried SSH with key-based authentication, where we observed the same detrimental effects of the attacks. In the case of whitelist access, there were no detrimental effects due to refused connections. Therefore, we can conclude that the attack is feasible only if the SSH and FTP servers respond to connection requests.

To illustrate the way Slowloris functions, we present the upload speed pattern of the Slowloris attack with 200 concurrent connections and different sleep intervals, emphasizing the periodic communication nature (Figure 4). The number of connections influences bandwidth by contributing to higher upload speed values. In contrast, the sleep parameter affects the duration of the displayed periods.

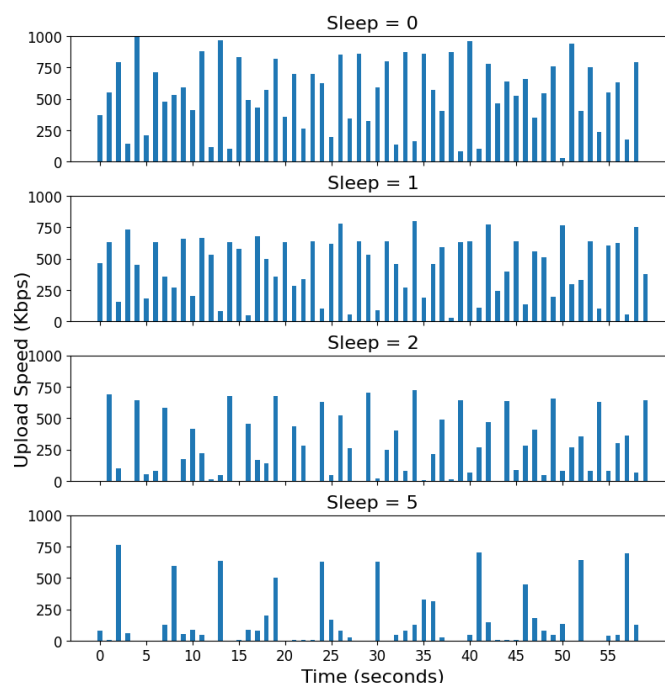


Figure 4. Upload speed on the attacker side caused by Slowloris attacks utilizing 200 connections and different timeout intervals.

In terms of software implementation, Dropbear SSH server showed excellent resistance against Slowloris due to a hard-coded limit of five unauthenticated users per IP address and 30 in total [47]. Such a small number of allowed connections was insufficient to launch an impactful low-rate attack or cause a service outage. It is worth noting that CPU usage and power consumption spiked by roughly 40% with just five simultaneous connections, but only with the sleep parameter set to 0, resulting in a high-rate attack requiring more than 1.5 Mbps of upload speed. Therefore, stopped testing Dropbear due to the hard-coded limitations, making any further Slowloris tweaking attempts futile.

OpenSSH on the other hand, does not have connection limitations in the default settings and has consequently allowed for extensive E-DoS attacks to take place. OpenSSH's implementation exhibited resilience against Slowloris showing no signs of service outage, thus allowing E-DoS attacks to remain unnoticed and last indefinitely. It is worth noting that OpenSSH settings allow manually adjusting the maximum number of unauthenticated

connections [48]. In Figure 5a–c, we present the relations between the Slowloris attack parameters and measured the attacker’s upload speed as well as the victim’s induced power consumption. In Figure 5a, the decision to use the square function for the ratio between measured power and upload speed, rather than a linear function, was made to emphasize the significance of changes in power consumption relative to changes in upload speed.

Regarding the FTP software implementations, vsftpd could not maintain service availability even against attacks of low intensity. Due to continuous crashes that required manual restarts, we discontinued further testing E-DoS attacks because of the lack of attack persistence. However, proftpd was more resilient, allowing for uninterrupted service and even faster response time than SSH, allowing for a two to three times faster process of building and renewing connections. The results of E-DoS attacks against proftpd are shown in Figure 5d–f. Both vsftpd and proftpd allow for limiting the overall number of maximum connections and connections per IP, but without discriminating between authenticated and unauthenticated ones.

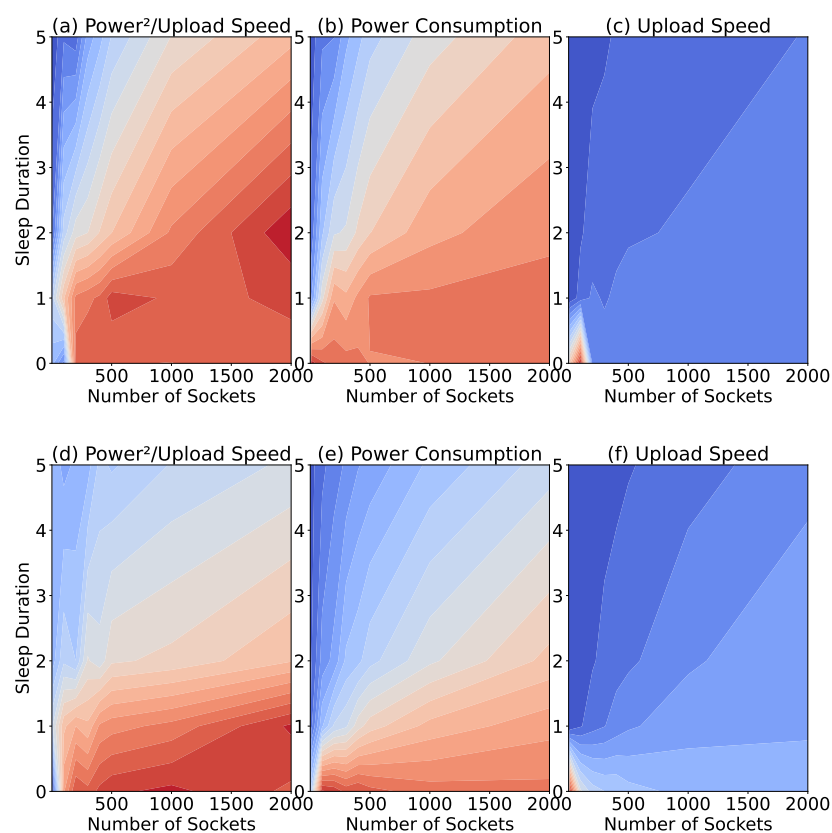


Figure 5. Induced power and upload speed measurements as a function of Slowloris parameters, where the attack is targeting an OpenSSH SSH server (a–c) and a proftpd FTP server (d–f). The color maps indicate value intensity, with red representing higher values and blue representing lower values.

As shown in Figure 5c,f, when sleep time is set to zero, with a low number of connections, Slowloris turns into a high-rate attack, causing the highest power consumption (Figure 5b,e). Without a sleep interval and an increasing number of connections, sending keep-alive headers consumes more time, making it impossible to achieve an infinite real-time loop of keep-alive messaging that caused the attack rate to spike. Hence, these anomalies are present only when there is a small number of concurrent connections. Therefore, as this study focuses only on low-rate attacks, we excluded these few outliers from Figure 6, which shows an approximately linear relation between the rate of Slowloris attack and the induced power consumption. Regarding finding the optimal attack strategy,

the best tuples of parameters are in the dark red area in Figure 5a,d since they induce the highest power consumption in relation to the invested bandwidth, which has to be minimized to remain undetected by the IDS. Table 2 summarizes our findings regarding E-DoS susceptibility of the most prevalent software implementations of the FTP and SSH protocols.

Table 2. E-DoS Susceptibility of FTP and SSH protocols' software implementations.

Software (Version)	E-DoS Susceptibility	Comment
OpenSSH (9.6)	True	The most vulnerable software under default settings.
Dropbear (2024.84)	Partially true	Due to the default unauthenticated connections limit, attacks have to be of high-rate (1.5 Mbps or more) for only a 40% increase.
vsftpd (3.0.3-13)	False	Crashes even during low-intensity attacks.
proftpd (v1.3.8b)	True	Less affected than OpenSSH.

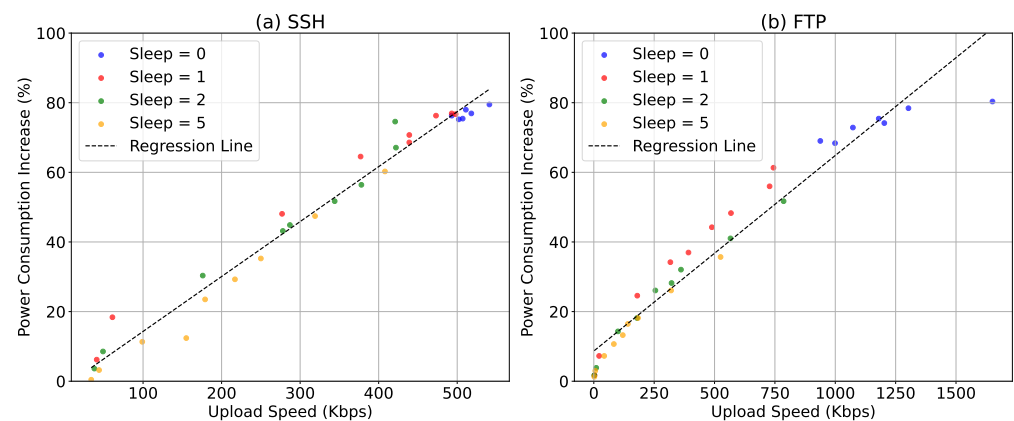


Figure 6. The impact of Slowloris attack upload speed (source end) on power consumption at the victim end.

Regarding CPU Usage during the separate tests, we observed real-time peaks ranging from 5% to 100%, increasingly proportional to the attack intensity. The peaks significantly affected power consumption and co-occurred periodically with the renewal of the established connections (keep-alive messages). We were unable to notice the difference in power consumption between external-only and combined measurement methods. The difference appears to be negligible due to the minimal impact of running SSH and `psutil` on the overall power consumption, and it may also fall within the measuring instrument's margin of error. Figure 7 shows CPU utilization changing as a function of Slowloris upload speed, exhibiting approximately linear dependency. We observed a strong correlation between CPU usage and power consumption, with a coefficient of 0.987 for SSH and 0.996 for FTP. Additionally, as the intensity of the attacks increases, whether through shorter sleep intervals or more connections, periods between the peaks begin to shorten, eventually causing constantly high CPU utilization.

These results are consistent with the expected protocol behavior: shorter sleep intervals and larger numbers of maintained connections force the server to refresh unauthenticated state more frequently, reducing idle time between bursts of activity and driving CPU usage upward. This interpretation is also in line with prior Raspberry Pi power-modeling work, which reported an approximately linear dependence between CPU utilization and power

consumption [37]. Although that study considered an older Raspberry Pi model and benign workloads rather than attack traffic, it supports our interpretation that the increased energy demand observed here is primarily driven by sustained CPU activity. The stronger effect observed for SSH is consistent with the additional cryptographic overhead of handling encrypted pre-authentication sessions, whereas FTP shows a milder but still clear impact.

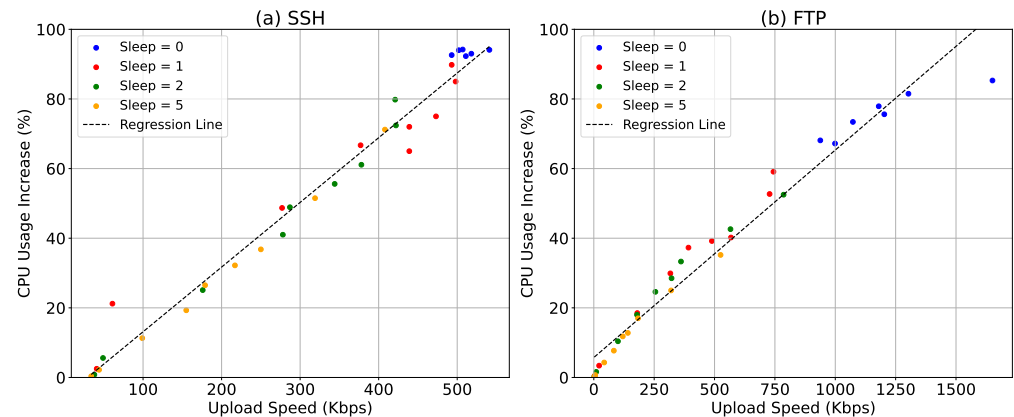


Figure 7. The impact of Slowloris attack upload speed (source end) on CPU usage at the victim end.

Additionally, we use thermal imaging as a qualitative illustration of how the Raspberry Pi looks before and during the attack (Figure 8). The images show the additional heat generated when the device is exposed to continuously high CPU load. This indicates that the CPU works harder, thus converting more energy into heat, as it consumes additional power. In Figure 8a, power consumption is at the baseline of 2.39 watts, while in Figure 8b, calculated power consumption is around 4.30 watts. For a compact SBC such as Raspberry Pi, prolonged operation under such heavy CPU load is undesirable. The results unequivocally identify the CPU as the primary source of power consumption during the attacks.

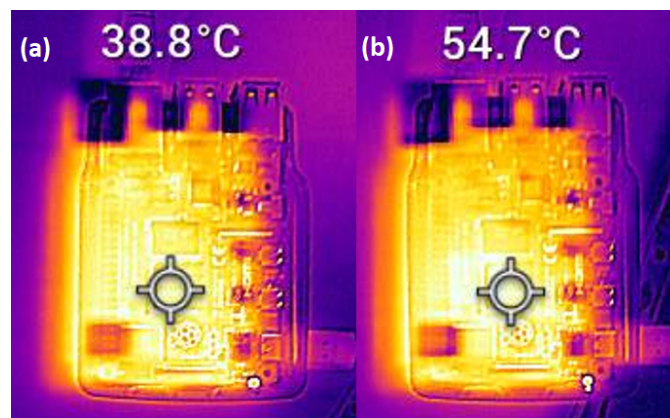


Figure 8. Thermal image of Raspberry Pi before (a) and during a Slowloris attack (b).

5. Discussion and Future Work

SSH and FTP both suffer from statefulness-induced asymmetries because the protocol requires servers to maintain expensive pre-authentication state of the unauthenticated connections, while the attacker can cheaply extend that state for a long time. Our experiments demonstrate how these asymmetries translate into disproportionate CPU and energy consumption under Slowloris attack traffic. The results unambiguously confirm that the tested FTP and SSH protocol implementations under default settings are highly susceptible to Slowloris E-DoS attacks.

The experiments in this study were conducted on a Raspberry Pi 4 Model B, which we selected as a representative and widely used SBC for experimental validation of the hypothesis. The purpose of this setup was to demonstrate the feasibility of repurposing Slowloris as an E-DoS attack in a realistic IoT-oriented environment, rather than to claim identical quantitative effects across all hardware platforms. Absolute power consumption, CPU utilization, and thermal response may vary across devices depending on their hardware characteristics. Prior Raspberry Pi power-modeling work has reported the same general trend that higher CPU utilization leads to higher power consumption under controlled workloads. Our results extend this observation to an attack setting, where the additional CPU and energy cost is driven by sustained handling of unauthenticated connections. However, the underlying issue highlighted in this work arises from the protocol/software-level handling of unauthenticated state in SSH and FTP services, and is therefore not specific to Raspberry Pi itself. In this sense, the presented results should be interpreted as a proof of concept showing what is possible when such stateful behavior is subjected to a Slowloris attack.

In servers resilient against the attacks, the vulnerability against Slowloris E-DoS attacks manifests in increased CPU utilization and, consequently, power consumption and device temperature. The results also reaffirm that Raspberry Pi is vulnerable to DoS attacks due to lacking in-built defense mechanisms. Knowing how difficult it is to detect low-rate application layer attacks such as Slowloris and the number of Raspberry Pi and other Linux-based devices that use the same software, the obtained results strongly suggest there are potentially millions of devices on the public internet vulnerable to the presented E-DoS threat. Within these devices, battery-powered ones could be critically affected because prolonged E-DoS attacks lead to power outages and total service outages. Furthermore, high operating temperatures decrease batteries and chips' lifespan, contributing to extra costs in addition to ones imposed by more frequent battery changes and charging.

Despite Slowloris being a recognized DoS threat for over a decade, it still poses a formidable challenge. The results show that this attack exhibited low-rate properties while causing computational asymmetry on the victim's end. Asymmetry implies that the attacker, with minimal effort, can impose substantial computational demands on the victim's resources. This characteristic underscores the potency of such attacks with minimal investment from the attacker's side. In our case, Slowloris exhibited low-rate asymmetrical application-layer DoS attack properties. Since FTP does not involve encryption, new connections pose a lesser burden on the CPU and allow for faster responses, leading to higher bandwidth and lower CPU usage than attacks against SSH due to lesser computational asymmetry.

Approximately linear relation between the invested upload speed and caused power consumption allows for fine-tuning of the DoS attack to avoid detection by the IDS. In the case of the SSH and FTP protocols, the results show that one of the possible solutions lies in zero-trust security policies [49], such as IP address whitelists that prevent any unnecessary traffic, including malicious. While we know many other application-layer E-DoS vulnerabilities could exist, the same solution applies in numerous other instances where public access is not necessary. There are also additional protocol-specific DoS countermeasures, such as limiting the maximum number of concurrent unauthenticated connections or connections per IP. General countermeasures include logging the service events and system behavior [28,49]. While such measures may thwart or reduce the impact of attacks in some scenarios, they may also be ineffective against DDoS attacks. Nevertheless, orchestrating DDoS attacks demands more significant resources than DoS attacks, requiring multiple attack sources, thus increasing the associated costs, which could act as a deterrent.

In the future, we plan to expand this research and assess the vulnerability of E-DoS attacks on other devices, services, and operating systems and ways to protect against them. These attacks affect various systems, such as cloud environments and IoT infrastructure. E-DoS attacks have the potential to significantly impact victims' power consumption, prompting a need to explore this research area further.

6. Conclusions

In this study, we examined the ramifications of the Slowloris E-DoS vulnerability within the context of the SSH and FTP protocols. We empirically verified its effects using a widely available off-the-shelf Raspberry Pi SBC. By exploiting this vulnerability, which entails executing a DoS attack to intentionally increase the victim's power consumption rather than cause an immediate service outage, we investigated the correlation between the invested upload speed and induced power consumption. The results show that with minimal effort, a single attacker can repurpose Slowloris to cause a low-rate asymmetrical application-layer E-DoS attack, resulting in up to 80% increased power consumption at the cost of just 500 Kbps. Moreover, the relation between the measured upload speed and power consumption is approximately linear, allowing the attacker to fine-tune the upload rate to evade detection mechanisms. The resilience of the ProFTPD and OpenSSH implementations of the FTP and SSH protocols under default settings against Slowloris enables uninterrupted service availability, thus allowing E-DoS attacks to remain unnoticed.

Our findings reveal a deeper structural weakness: stateful protocols that maintain pre-authentication state for unauthenticated users create inherent asymmetries, where an inexpensive attacker action results in disproportionately high server resource consumption. In SSH, this vulnerability is further exacerbated by the computational overhead of encrypting each unauthenticated connection. We suggest applying zero-trust security policies and limiting the maximum number of concurrent unauthenticated connections as the most effective countermeasure. The key findings of this study expose a vulnerability potentially affecting millions of devices on the public internet. Beyond SSH and FTP, similar unauthenticated connection handling is common in other stateful protocols used in IoT backends and edge systems, making it an important design consideration for future secure deployments. In summary, the stateful design of widely deployed protocols can turn service availability into an energy liability. This study lays the groundwork for deeper investigations into E-DoS threats, urging researchers and industry alike to bolster security measures and pave the way for future advancements in E-DoS protection strategies.

Author Contributions: Conceptualization, N.G. and G.B.; methodology, N.G. and A.S.; validation, N.G. and G.B.; formal analysis, N.G.; investigation, N.G.; writing—original draft preparation, N.G. and A.S.; writing—review and editing, all authors contributed; visualization, N.G.; supervision, G.B. and A.S.; project administration, A.S. and G.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Kristiania University of Applied Sciences.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Adedeji, K.B.; Abu-Mahfouz, A.M.; Kurien, A.M. DDoS attack and detection methods in internet-enabled networks: Concept, research perspectives, and challenges. *J. Sens. Actuator Netw.* **2023**, *12*, 51. [\[CrossRef\]](#)
2. Khan, W.Z.; Aalsalem, M.Y.; Khan, M.K. Communal Acts of IoT Consumers: A Potential Threat to Security and Privacy. *IEEE Trans. Consum. Electron.* **2019**, *65*, 64–72. [\[CrossRef\]](#)

3. Wu, Z.; Li, W.; Liu, L.; Yue, M. Low-Rate DoS Attacks, Detection, Defense, and Challenges: A Survey. *IEEE Access* **2020**, *8*, 43920–43943. [CrossRef]
4. Cambiaso, E.; Papaleo, G.; Chiola, G.; Aiello, M. Slow DoS attacks: Definition and categorisation. *Int. J. Trust. Manag. Comput. Commun.* **2013**, *1*, 300–319. [CrossRef]
5. Rios, V.D.M.; Inácio, P.R.M.; Magoni, D.; Freire, M.M. Detection and Mitigation of Low-Rate Denial-of-Service Attacks: A Survey. *IEEE Access* **2022**, *10*, 76648–76668. [CrossRef]
6. Ficco, M.; Palmieri, F. Introducing Fraudulent Energy Consumption in Cloud Infrastructures: A New Generation of Denial-of-Service Attacks. *IEEE Syst. J.* **2017**, *11*, 460–470. [CrossRef]
7. Ngu, A.H.; Gutierrez, M.; Metsis, V.; Nepal, S.; Sheng, Q.Z. IoT Middleware: A Survey on Issues and Enabling Technologies. *IEEE Internet Things J.* **2017**, *4*, 1–20. [CrossRef]
8. Ionescu, V.; Roedig, U. Battery depletion attacks on NB-IoT devices using interference. In *Proceedings of the European Symposium on Research in Computer Security*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 276–295.
9. Shodan. Available online: <https://www.shodan.io> (accessed on 22 March 2024).
10. Foundation, R.P. Teach, Learn, and Make with the Raspberry Pi Foundation. Available online: <https://www.raspberrypi.org/> (accessed on 22 March 2024).
11. Zhao, C.W.; Jegatheesan, J.; Loon, S.C. Exploring iot application using raspberry Pi. *Int. J. Comput. Netw. Appl.* **2015**, *2*, 27–34.
12. Arreaga, N.X.; Enriquez, G.M.; Blanc, S.; Estrada, R. Security Vulnerability Analysis for IoT Devices Raspberry Pi Using PENTEST. *Procedia Comput. Sci.* **2023**, *224*, 223–230. [CrossRef]
13. Manners, D. Raspberry Pi Anticipates Consistent Million Unit Months. Available online: <https://www.electronicweekly.com/news/business/raspberry-pi-anticipates-consistent-million-unit-quarters-2023-06/> (accessed on 22 March 2024).
14. The Epic Story of the Raspberry Pi—RaspberryTips. Available online: <https://raspberrytips.com/raspberry-pi-history/> (accessed on 22 March 2024).
15. Raspberry Pi Has Now Sold 30 Million Tiny Single-Board Computers. Available online: <https://www.zdnet.com/article/raspberry-pi-now-weve-sold-30-million/> (accessed on 22 March 2024).
16. Collins, S. The Life of Pi: Ten Years of Raspberry Pi. Available online: <https://www.cam.ac.uk/stories/raspberrypi> (accessed on 22 March 2024).
17. Collins, S. Raspberry Pi foundation. Available online: <https://www.raspberrypi.org/blog/pi-day-2023-support-computing-education/> (accessed on 22 March 2024).
18. Liu, Y.; Tsang, K.F.; Wu, C.K.; Wei, Y.; Wang, H.; Zhu, H. IEEE P2668-Compliant Multi-Layer IoT-DDoS Defense System Using Deep Reinforcement Learning. *IEEE Trans. Consum. Electron.* **2023**, *69*, 49–64. [CrossRef]
19. Ebrahimpour, E.; Babaie, S. Authentication in Internet of Things, protocols, attacks, and open issues: A systematic literature review. *Int. J. Inf. Secur.* **2024**, *23*, 1583–1602. [CrossRef]
20. Palmieri, F.; Ricciardi, S.; Fiore, U. Evaluating Network-Based DoS Attacks under the Energy Consumption Perspective: New Security Issues in the Coming Green ICT Area. In *Proceedings of the 2011 International Conference on Broadband and Wireless Computing, Communication and Applications*; IEEE: Piscataway, NJ, USA, 2011; pp. 374–379. [CrossRef]
21. Tushir, B.; Dalal, Y.; Dezfouli, B.; Liu, Y. A Quantitative Study of DDoS and E-DDoS Attacks on WiFi Smart Home Devices. *IEEE Internet Things J.* **2021**, *8*, 6282–6292. [CrossRef]
22. Al-Haidari, F.; Sqalli, M.; Salah, K. Evaluation of the impact of EDoS attacks against cloud computing services. *Arab. J. Sci. Eng.* **2015**, *40*, 773–785. [CrossRef]
23. Ylonen, T.; Lonvick, C. The Secure Shell (SSH) Protocol Assigned Numbers. Request for Comments RFC 4250, Internet Engineering Task Force. 2006. Available online: <https://www.rfc-editor.org/info/rfc4250> (accessed on 22 March 2024).
24. Ylonen, T.; Lonvick, C. The Secure Shell (SSH) Authentication Protocol. Request for Comments RFC 4252, Internet Engineering Task Force. 2006. Available online: <https://www.rfc-editor.org/info/rfc4252> (accessed on 22 March 2024).
25. Ylonen, T.; Lonvick, C. The Secure Shell (SSH) Transport Layer Protocol. Request for Comments RFC 4253, Internet Engineering Task Force. 2006. Available online: <https://www.rfc-editor.org/info/rfc4253> (accessed on 22 March 2024).
26. Ylonen, T.; Lonvick, C. The Secure Shell (SSH) Connection Protocol. Request for Comments RFC 4254, Internet Engineering Task Force, 2006. Available online: <https://www.rfc-editor.org/info/rfc4254> (accessed on 22 March 2024).
27. Nmap: The Network Mapper-Free Security Scanner. Available online: <https://nmap.org/> (accessed on 22 March 2024).
28. SecureBlackbox 16: Counteracting Denial-of-Service (DoS) Attacks in SSH and SFTP Servers. Available online: <https://www.nsoftware.com/kb/articles/legacy/sbb/dos-attacks-in-ssh-sftp-servers> (accessed on 22 March 2024).
29. Abdulganiyu, O.H.; Ait Tchakoucht, T.; Saheed, Y.K. A systematic literature review for network intrusion detection system (IDS). *Int. J. Inf. Secur.* **2023**, *22*, 1125–1162. [CrossRef]
30. Andrews, R.; Hahn, D.A.; Bardas, A.G. Measuring the Prevalence of the Password Authentication Vulnerability in SSH. In *Proceedings of the ICC 2020–2020 IEEE International Conference on Communications (ICC)*; IEEE: Piscataway, NJ, USA, 2020; pp. 1–7. [CrossRef]

31. Hossain, M.D.; Ochiai, H.; Doudou, F.; Kadobayashi, Y. SSH and FTP Brute-Force Attacks Detection in Computer Networks: LSTM and Machine Learning Approaches. In *Proceedings of the 2020 5th International Conference on Computer and Communication Systems (ICCCS)*; IEEE: Piscataway, NJ, USA, 2020; pp. 491–497. [\[CrossRef\]](#)
32. File Transfer Protocol. Request for Comments RFC 114, Internet Engineering Task Force. 1971. Available online: <https://www.rfc-editor.org/info/rfc114> (accessed on 22 March 2024).
33. Dey, D.; Dinda, A.; Kundapur, P.P.; Smitha, R. Warezmaster and Warezclient: An Implementation of FTP Based R2L Attacks. In *Proceedings of the 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6. [\[CrossRef\]](#)
34. Al-Husseini, H.; Hosseini, M.M.; Yousofi, A.; Alazzawi, M.A. Whale optimization algorithm-enhanced long short-term memory classifier with novel wrapped feature selection for intrusion detection. *J. Sens. Actuator Netw.* **2024**, *13*, 73. [\[CrossRef\]](#)
35. Sainz-Raso, J.; Martin, S.; Diaz, G.; Castro, M. Security Vulnerabilities in Raspberry Pi—Analysis of the System Weaknesses. *IEEE Consum. Electron. Mag.* **2019**, *8*, 47–52. [\[CrossRef\]](#)
36. Ltd, R.P. Raspberry Pi 4 Model B Specifications. Available online: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/> (accessed on 22 March 2024).
37. Kaup, F.; Gottschling, P.; Hausheer, D. PowerPi: Measuring and modeling the power consumption of the Raspberry Pi. In *Proceedings of the 39th Annual IEEE Conference on Local Computer Networks*; IEEE: Piscataway, NJ, USA, 2014; pp. 236–243.
38. Black, S.; Kim, Y. An Overview on Detection and Prevention of Application Layer DDoS Attacks. In *Proceedings of the 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*; IEEE: Piscataway, NJ, USA, 2022; pp. 791–800. [\[CrossRef\]](#)
39. Tajane, V.; Sharma, D. Effective Detection and Prevention of DDoS in Cloud Computing Environment. In *Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*; IEEE: Piscataway, NJ, USA, 2018; pp. 1–5. [\[CrossRef\]](#)
40. Gillen, R.E.; Carter, J.M.; Craig, C.; Johnson, J.A.; Scott, S.L. Assessing Anomaly-Based Intrusion Detection Configurations for Industrial Control Systems. In *Proceedings of the 2020 IEEE 21st International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM)*; IEEE: Piscataway, NJ, USA, 2020; pp. 360–366. [\[CrossRef\]](#)
41. Yevsieieva, O.; Helalat, S.M. Analysis of the Impact of the Slow HTTP DOS and DDOS Attacks on the Cloud Environment. In *Proceedings of the 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*; IEEE: Piscataway, NJ, USA, 2017; pp. 519–523. [\[CrossRef\]](#)
42. What is OpenSSH? Available online: <https://www.ssh.com/academy/ssh/openssh> (accessed on 22 March 2024).
43. Dropbear SSH. Available online: <https://matt.ucc.asn.au/dropbear/dropbear.html> (accessed on 22 March 2024).
44. Vsftpd-Secure, Fast FTP Server for UNIX-like Systems. Available online: <https://security.appspot.com/vsftpd.html> (accessed on 22 March 2024).
45. The ProFTPD Project: Home. Available online: <http://www.proftpd.org/> (accessed on 22 March 2024).
46. Psutil · PyPI. Available online: <https://pypi.org/project/psutil/> (accessed on 22 March 2024).
47. Dropbear/Src/Default_options.h at MFaster · Mkj/Dropbear. Available online: https://github.com/mkj/dropbear/blob/master/src/default_options.h (accessed on 22 March 2024).
48. Hansteen, P.N.M. OpenSSH Cookbook: Load Balancing. Available online: https://en.wikibooks.org/wiki/OpenSSH/Cookbook/Load_Balancing (accessed on 22 March 2024).
49. Zero Trust Model-Modern Security Architecture | Microsoft Security. Available online: <https://www.microsoft.com/en-us/security/business/zero-trust> (accessed on 22 March 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.