Securing Smart Home Networks with Software-Defined Perimeter

Ahmed Sallam*‡, Ahmed Refaey*†, and Abdallah Shami*
*Western University, London, Ontario, Canada; e-mails: {asallam3, ahusse7, abdallah.shami}@uwo.ca
†Manhattan College, Riverdale, New York, USA; e-mail: {ahmed.hussein}@manhattan.edu
‡Suez Canal University, Ismailia, Egypt; e-mail: {sallam_ah}@ci.suez.edu.eg

***Abstract*—Internet of Things (IoT) allows households to have real-time access to various services through a range of smart-home technologies. These technologies allow owners to reduce expenses for security, heating, cooling, lighting, electricity, and water. In general, using IoT and smart-home technologies enables new sharing and service economy paradigms to unlock the value of surplus resources. This requires a more holistic approach, exploring synergies across and between verticals to maximize efficiencies on a wider scale. Further, there is a demand to optimize the use of existing assets and resources while maintaining the security and privacy of users through exploring new appropriate frameworks. Indeed, Software-Defined Perimeters (SDP) can play a crucial role in security and present an ideal network security framework for smart-home technologies. There are several benefits as a result of adopting SDP in smart-home technologies, such as providing light-weight authentication for devices as well as dynamically updating firewall rules. To this end, the integration between the SDP and smart-home infrastructure is examined through virtualized network testbeds. The testing results prove that SDP can provide reasoning capabilities to repel different attacks on the smart-home infrastructures, such as flooding and spoofing types of attacks, intrusion detection, and eavesdropping activities.**

***Index Terms*—Smart Home Networks, Security, Privacy, DoS Attack, Software-Defined Perimeter (SDP)**

I. Introduction

According to Zion Market Research, the global Smart Home market is expected to reach approximately USD 53.45 billion in 2022[1]. Nevertheless, the full potential of the Smart Home industry are yet to be achieved. In a report revealed by Forbes Technology Council, many experts hold the view that security and privacy become fundamental concerns that will shape this industry [2].

The basic idea behind the Smart Home is to create a convenient technology which connects all devices and appliances to be in communication with one another as well as the homeowner. Recent developments in computer communication technology, allows these connections to go even further with the ability now to access the Smart Home from anywhere and at any time. This includes the possibility to link the Smart Home to public networks and service platforms, such as Cloud technology. As a result, many technologies have emerged to connect smart devices using radio waves, including ZigBee, Z-Wave, Bluetooth, and WiFi [3]. The connections in the home network are established with the combination of a set of sensors and Internet of Things devices, which control appliances and various devices such as, temperature gauges, home lighting, and entertainment systems. Subsequently, a unified controller/server generates a link between these smart devices to either a local or public network. (see Fig. 3).

One major issue with IoT devices is that they have limited computation capabilities and memory capacity. In addition, there are anticipated privacy issues, based on the Garter report which revealed an expected growth of embedded sensors and IoT connected devices to reach 20.4 billion by 2020 [4]. Consequently, malicious intruders have a greater possibility of taking advantage as many IoT manufacturers are focused on functionality and often overlook privacy issues. Unfortunately, many state-of-the-art security frameworks are highly centralized and are difficult to scale, thus not necessarily well-suited for Smart Home. The issues mentioned above, provide the justification for a SDP platform to be used to secure Smart Home networks and prevent potential threats, such as spoofing attacks, intrusions, eavesdropping, and malwares [5].

SDP was developed as an improved method to replace physical security appliances with software modules that can be remotely managed. The principles of SDP were first implemented by multiple organizations within the Department of Defense (DoD) and Intelligence communities (IC). In 2014 the Cloud Security Alliance (CSA) outlined the initial protocol for the Software Defined Perimeter specification[6] and later in 2016 Waverley labs developed the first open source SDP platform. Recently, SDP platform has gained considerable attention due to a set of key principles, including abstraction, programmability, automation, and centralized management. These principles clarify some interesting and potential integration points between the SDP and various modern network platforms, such as the Smart Homes' network. The contributions of this paper can be summarized as follows:

- Integrate the SDP platform into a smart home's network to provide a more secure networking platform and ensure seamless integration between the two paradigms.
- Build a virtualized network testbed to introduce and evaluate the aforementioned architecture.

The rest of this paper is organized as follows: Section 2 reviews the previous work of smart homes' security and privacy. Section 3 gives insight into SDP architecture and its current security tools. Section 4 presents the SDP structure and explains its functionality. Section 5 proposes a new network security architecture for smart homes using SDP. Section

978-1-5386-7747-6/19/$31.00 ©2019 IEEE

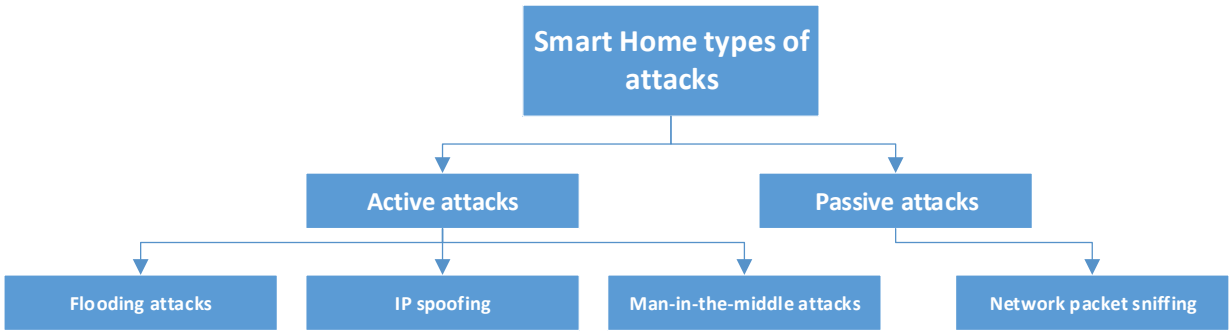

Fig. 1. Smart home's network

6 evaluates the new integrated platform. Finally, Section 7 concludes this work.

II. Previous Work

Recently, network vulnerabilities have experienced unprecedented growth. Therefore, Smart Home security and privacy has received considerable critical attention. This in addition to the the fact that the tools and information needed to penetrate the security of Smart Home networks are widely available has increased that concern. There are multiple opportunities for attacking the Smart Home network, as well as public networks. Fig. 1 exhibit the common methods of attack that present opportunities to compromise a Smart Home Network. The passive attacks attempt to learn or make use of information from the system without affecting system resources. The active attacks, on the other hand, attempt to alter system resources or effect their operations [7], [8]. For example, Denial of Service (DoS) or Distributed DoS (DDoS) aim to expose the resources of a dedicated server or to slow legitimate users causing a denial of service.

The common method to mitigate active attacks such as flooding attacks is accomplished by installing hardware-based middle-boxes deployed with dedicated security functions such as intrusion detection (IDS), firewalls, and anti-malware [9]. However, hardware security appliances is too expansive to be used in Smart Home and may not provide a means to protect networks from all possible attacks. IP spoofing and network packet sniffing on the other hand are usually repelled using different cryptography techniques includes symmetric key algorithms, also known as private-key cryptography and asymmetric key algorithms, also known as public key cryptography.

To date, a few number of studies begun to develop a dedicated systems to secure Smart Home based on modern security trends. For instance, in [10] the authors proposed a simple architecture consists of a home gateway and several IoT devices connected via a WiFi network. In this architecture each IoT device is detached and can be controlled separately using a mobile device, in other words there is no controller to manage the IoT devices. The gateway uses Elliptic Curve Cryptography for authenticating and monitoring the communication between devices in the system. The authors of [11] proposed a more complicated architecture based on the Blockchain technology to protect the privacy. The architecture consists of three core tires that are: Smart Home, cloud storage, and overlay. The IoT devices inside the Smart home are centrally managed by a minor which responsible for handling all authentication and communication within and external to the home. The authors claim their new architecture eliminating significant challenges related to the Blockchain technology such as: high resource demand and long latency. However, the mining process still a challenge not to mention the miner cost.

Although the above systems consider the network privacy, they don't not address other critical threats such as flooding attacks. Furthermore, these systems do not consider the total cost as a critical factor in Smart Home networks.

III. Software-defined Perimeters

Typically, the SDP platform consists of three main components: SDP Initiating Host (IH), SDP Accepting host (AH), and SDP controller (CTRL). These components are used to create secure perimeters between legitimate clients and the available service in the network. A very simple scenario will include a legitimate client which is provided with an IH module installed. This client is trying to access a service/s that hides behind a gateway in a private subnet. The gateway is provided with an AH module installed and a firewall that has a drop-all policy established for all traffic. Additionally, a third machine (SDP controller) running the CTRL module is presented to authenticate the communication process. During the network setup, the network administrator should access the CTRL module to identify the legitimate clients and identify the services that they can access. Moreover, the network administrator has to create credential keys and certificates and distribute these credential among the IH and AH components to authenticate their access to the CTRL.

A. SDP connection setup

For a legitimate client to access a service, first, the IH software already installed in its side (also known as SPA-client) sends a valid Single Packet Authorization (SPA) packet (encrypted, non-replayed, with an HMAC SHA-256). When the CTRL component authenticates the SPA packet, it will

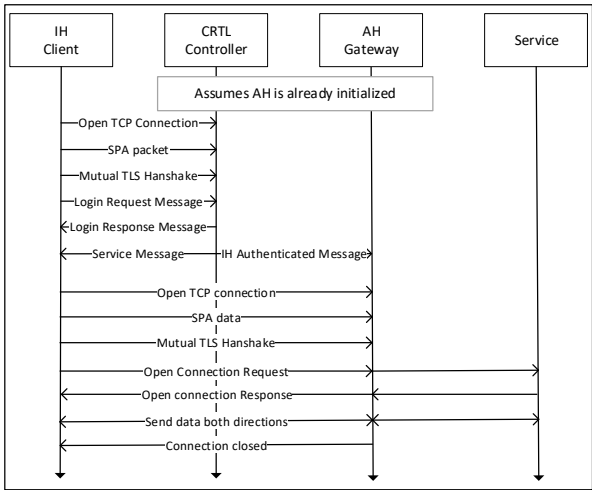

Fig. 2. SDP connection setup

message the AH component at the gateway to configure proper rules for the client for a predefined period to access the services. Even though the firewall is enabled at the gateway the AH daemon still can receive the message and authenticate the client and then establishing Mutual Transport Layer Security (mTLS) connection. After a configurable timeout, the rule to accept the incoming connection will be deleted, but the connection will remain open by using a connection tracking mechanism provided by the firewall [12]. It is worth mentioning that, the AH has to follow a similar sequence as well as the IH to initiate the connection with the controller (see Fig. 2).

Based on the Cloud Security Alliance SDP specifications, in a typical scenario, the legitimate client has a direct access to the SDP controller. This was to illustrate how communication is done according to the CSA standard specifications. However, this scenario does not represent a secure placement for the SDP controller and exposing it to direct attacks that can lead to critical risk where an intruder can take off the whole network. This can be avoided by hiding the controller behind the gateway.

B. SDP protection

According to the results of 2015 third annual Hackathon contest, SDP withstood tens of thousands of attacks to remain undefeated to deprive many of the contestants of $10,000 prize offered by Cloud Security Alliance [13].

The fact that the gateway's firewall has a drop-all policy makes SDP worthwhile to repel flooding attacks and port scanning attacks successfully. SPA mitigates these attacks because it allows the server to discard the TLS DoS attempt before entering the TLS handshake [6].

Moreover, in SDP platform the connection between all hosts (IH and AH) must use TLS or Internet Key Exchange (IKE) with mutual authentication to validate the client as a legitimate member of the SDP prior to further device validation and/or user authentication. This has the ability to prevent Network Manipulation attacks, MITM attacks, and Traffic sniffing attacks.

IV. Proposed Framework

This section draws together the SDP components and Smart Home network in a new architecture. Firstly, the network architecture is described, and then the control flow of the connection is explained.

In this architecture, the Smart Home network consists of a set of sensors and a controller/server connected together and forming the local Smart Home network. It is worth mentioning that, in this work the Message Queue Telemetry (MQTT) network protocol was adapted to manage the communication between the different network components. MQTT is a publish subscribe messaging protocol for use on top of the TCP/IP protocol. Is alight weight protocol designed where a small code footprint is required. The simplest form of MQTT connection consists of a broker, publisher, and subscriber, each component is acting as follows:

- **Broker:** is responsible for managing the transmissions between subscribers and publishers. it can also buffer the messages while subscribers are offline.
- **Publisher:** publishes a message which contains a peace of information such as a temperature read to the broker with a particular name/topic.
- **Subscriber:** listens for a messages with a particular topic.

To integrate the SDP platform with this network, a separate gateway machine is added to the network which runs an AH module to authenticate and manage the communication to the public network. Additionally, the SDP CTRL module is added to the MQTT broker to represent the SDP controller. Finally, each authorized subscriber client runs an IH module to gain access to the Smart Home broker/server (see Fig. 3).

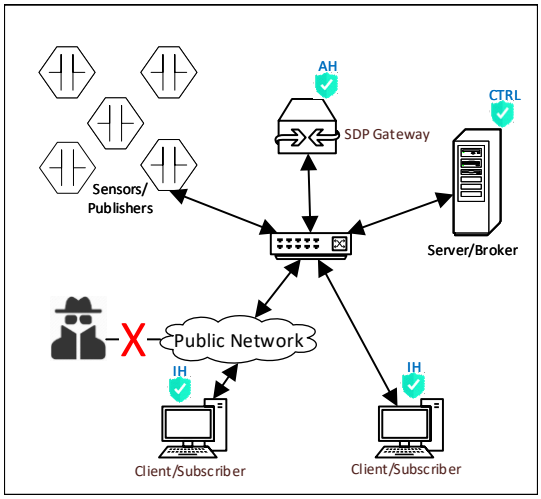

Fig. 3. The proposed framework

V. Performance Evaluation

Three evaluation matrices are considered to evaluate the performance of this architecture:

- The service ability, that is, the network ability to repel potential threats and attacks such as DoS attack.
- The data privacy.
- The network throughput.

These metrics were compared with and without SDP installed. Furthermore, two types of attacks were initiated to test the performance of SDP protection, namely, packet sniffing attack and Denial of Service (DoS) attack. The packet sniffing attack was chosen to represents a threat to the data privacy and DoS attack was chosen to represents a threat to the service availability. In other words, these attacks represents a common threats in the Smart Home network.

A. Testbed settings

The experiments were run using a testbed consists of a set of virtual machines hosted by a physical machine running linux Ubuntu 18.04 and VirtualBox 5.2. The test bed is divided into two parts, the Smart Home network and the SDP platform. The Smart Home network consists mainly of five VMs running MQTT publisher clients. These VMs represent a different IoT devices that can publish different information. The publisher VMs connected to another virtual machine running an MQTT broker/server. Additionally, another VM is created to run an MQTT subscriber client. The SDP platform used in this testbed utilize the modules of the Open Source SDP project developed by Waverly. The SDPController module was installed on an the server VM to represent the SDP controller, this decision was taken to reduce the complexity of the network and the resources used. The fwknop module was installed on a subscriber VM to represent the SDP IH on a legitimate client. Furthermore, the fwknop module was installed again with different settings on a separate VM to represent the SDP AH and act as the gateway. Finally, one more VM was created to act as an attacker. Table I gives a brief description of the entire testbed environment and its configuration.

TABLE I
TESTBED SETUP

Machine	Software	Specifications
Host	- Linux Ubuntu 18.04 - VirtualBox 5.2 - Open vSwitch 2.5.6	- 4.20GHz Intel® Core™ i7-8650U, 8MB Cache - 32 GB DDR4 - 2 GB NVIDIA Quadro P500 - 1.0 Gbps FDX NIC
VM1 (SDP Controller)	- Linux Ubuntu 16.04 - Mosquitto server 3.1.1 - Waverely SDPcontroller module	- 1 vProcessor - 1 GB RAM - 2 NICs
VM2 (Gateway)	- Linux Ubuntu 16.04 Waverely fwknop module	- 1 vProcessor - 2 GB RAM - 3 NICs
VM3 (Subscriber)	- Linux Ubuntu 16.04 - Mosquitto clients 3.1.1 - Waverely fwknop module	- 1 vProcessor - 1 GB RAM - 2 NICs
VM4 (Attacker)	Linux Ubuntu 16.04	- 1 vProcessor - 1 GB RAM - 2 NICs
VM5-VM9 (Publishers)	- Linux Ubuntu 16.04 - Mosquitto clients 3.1.1	- 1 vProcessor - 1 GB RAM - 2 NICs

B. Attacks setup

Two types of attacks where chosen to represent both Passive and Active attacks respectively as follows:

- **Packet sniffing attack:** This attack was applied by sniffing the packets transmitted between the publisher and the broker using Wireshark v2.6.
- **DoS attack:** Syn Flood Attack was launched via spoofed broadcast of TCP connection requests with Syn flag using hping3 utility.

C. Experiments formulation

The network throughput was reported by initiating the five VM publishers, such that each VM is sending messages at steady rate. The message rate is set to 1 message per second per publisher asynchronously. The message payload is fixed for each publisher and set to approximately 30 bytes. This test does not try to reach the maximum message throughput. However, the goal is to show how the network throughput would differ with and without the SDP platform. In this way, the number of publisher gives the global message throughput.

Secondly, the network throughput was reported again. But this time, the traffic was captured after lunching the DoS attack for two scenarios, when the SDP is enabled and when the SDP is disabled. Finally, the transmission packets were sniffed at the subscriber VM during the authentication time of the MQTT subscriber client with the broker. The reason behind this is to investigate the data privacy of the subscribers in MQTT before and after the SDP platform.

D. Result analysis

The network traffic was captured at the broker/server VM with and without the SDP platform. Fig. 4 shows that there is insignificance delay by running the SDP modules in the Smart Home network. In fact, the only difference after activating the SDP platform would be the time required to authenticate the different hosts with the controller. That is, to authenticate the subscriber client VM and the gateway VM with the controller. This authentication process happens only once unless the host end the session for some reasons.

Again, the traffic was captured at the broker VM with and without the SDP platform. However, the gateway was hit by a flood of SYN packets of a DoS attack around second 14. Fig. 5 shows that, without SDP, the network throughput was dropped down and almost stopped. The reason for this is that, the Dos attack was able to exhaust the network resources at the gateway VM. On the other hand, with SDP platform, the DOS attack was mitigated and the legitimate clients were retained connected. The reason for this is that, any packets received at the gateway interface are considered an attack except the SPA packet. Thus, the gateway drop these packets before entering the TLS handshake. Of course, the DoS attack still affect the

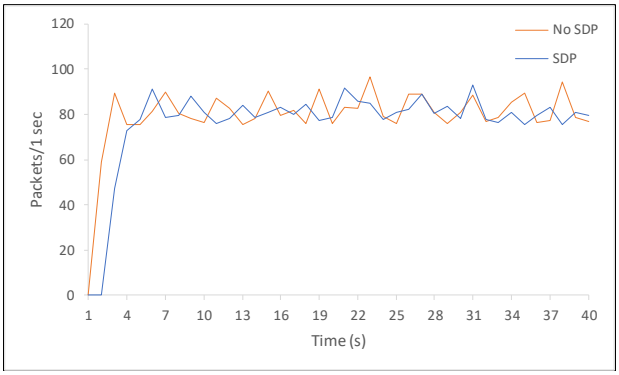

Fig. 4. Network throughput

network throughput but to an acceptable limits. It is worth mentioning that, it takes 4.182067 s and 2.068458 s for the AH and IH to start up respectively. This is a one time process and doesn't pose any further overhead on the network throughput.

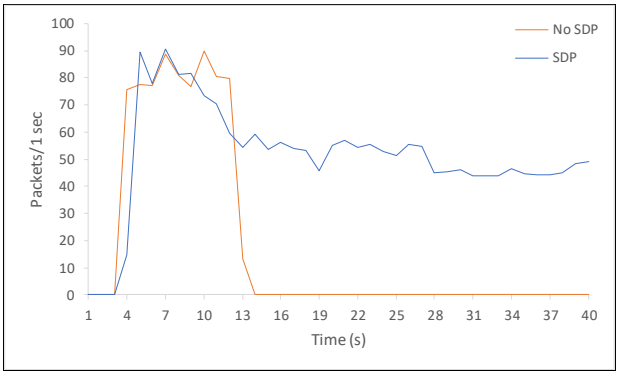

Fig. 5. Network throughput under DoS attack

From the privacy point of view, by inspecting the packets sniffed during the authentication of the MQTT subscriber client with the broker. The user name and the password of the MQTT subscriber client where spotted as shown in Fig. 6. Even though encryption across the network can be handled with SSL network security, independently of the MQTT protocol itself. In fact, this is not something built-in to the protocol in order to keep it simple and lightweight [14]. Therefore, SDP platform will add another layer of security to the authentication procedure through the SPA packet which add insignificant overhead only once at the authentication time.

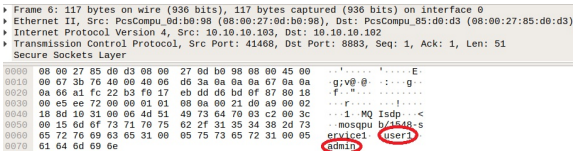

Fig. 6. Network throughput under DoS attack

VI. Conclusions

This paper set out to develop a new secure Smart Home network using Software-defined parameters platform. First, a brief summary of the Smart Home security challenges were explained. Then, a client-gateway SDP architecture was adopted to build a new secure Smart Home network. Furthermore, the performance of the network was evaluated in terms of network throughput and connection setup time under denial of service attack. These experiments confirmed that SDP architecture was successfully able to mitigate the DoS attack. Additionally, SDP will also be able to prevent other types of attacks such as IP spoofing and sniffing attacks due to the SPA key and the mutual TLS connection provided by the SDP platform. These findings clearly indicate that SDP platform can provide a complete secure and flexible environment for Smart home networks. Another major finding was that, SDP platform has insignificant effect on the Smart Home network throughput. Finally, the third major finding was that, SDP platform would introduce an applicable and affordable solution to reduce the total cost of securing Smart Home networks.

References

[1] Zion Market Research, "Smart Home Market Size & Share will hit $53.45 Billion by 2022," 2017. [Online]. Available: https://globenewswire.com/news-release/2017/04/12/959610/0/en/Smart-Home-Market-Size-Share-will-hit-53-45-Billion-by-2022.html

[2] Forbes Technology Council, "14 Predictions For The Future Of Smart Home Technology," 2018. [Online]. Available: https://www.forbes.com/sites/forbestechcouncil/2018/01/12/14-predictions-for-the-future-of-smart-home-technology

[3] R. J. Robles and T.-h. Kim, "A Review on Security in Smart Home Development," *International Journal of Advanced Science and Technology*, vol. 15, 2010.

[4] Gartner, "Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016," 2017. [Online]. Available: https://www.gartner.com/newsroom/id/3598917

[5] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT Security Techniques Based on Machine Learning," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 41–49, 2018.

[6] Cloud Security Alliance, "SDP Specification 1.0," 2014.

[7] N. Komninos, E. Philippou, and A. Pitsillides, "Survey in smart grid and smart home security: Issues, challenges and countermeasures," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, pp. 1933–1954, 2014.

[8] P. Kumar, A. Moubayed, A. Refaey, A. Shami, and J. Koilpillai, "Performance analysis of sdp for secure internal enterprises," in *2019 IEEE Wireless Communications and Networking Conference*, 2019.

[9] S. Hoque, A. Rahim, and F. Di Cerbo, "Cyber Security and Privacy," *Communications in Computer and Information Science*, vol. 470, pp. 89–96, 2014.

[10] F. K. Santoso and N. C. Vun, "Securing IoT for smart home system," in *2015 International Symposium on Consumer Electronics (ISCE)*, 2015.

[11] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops*, 2017.

[12] A. Moubayed, A. Refaey, and A. Shami, "Software defined perimeter: State of the art secure solution for modern networks," *To appear in IEEE Network Magazine*, 2019.

[13] K. Griffith, "Software-Defined Perimeter Remains Undefeated in Hackathon," 2018. [Online]. Available: https://www.sdxcentral.com/articles/news/software-defined-perimeter-remains-undefeated-in-hackathon/2015/08/

[14] R. Light, "mosquitto.conf-the configuration file for mosquitto," 2019. [Online]. Available: https://mosquitto.org/man/mosquitto-conf-5.html