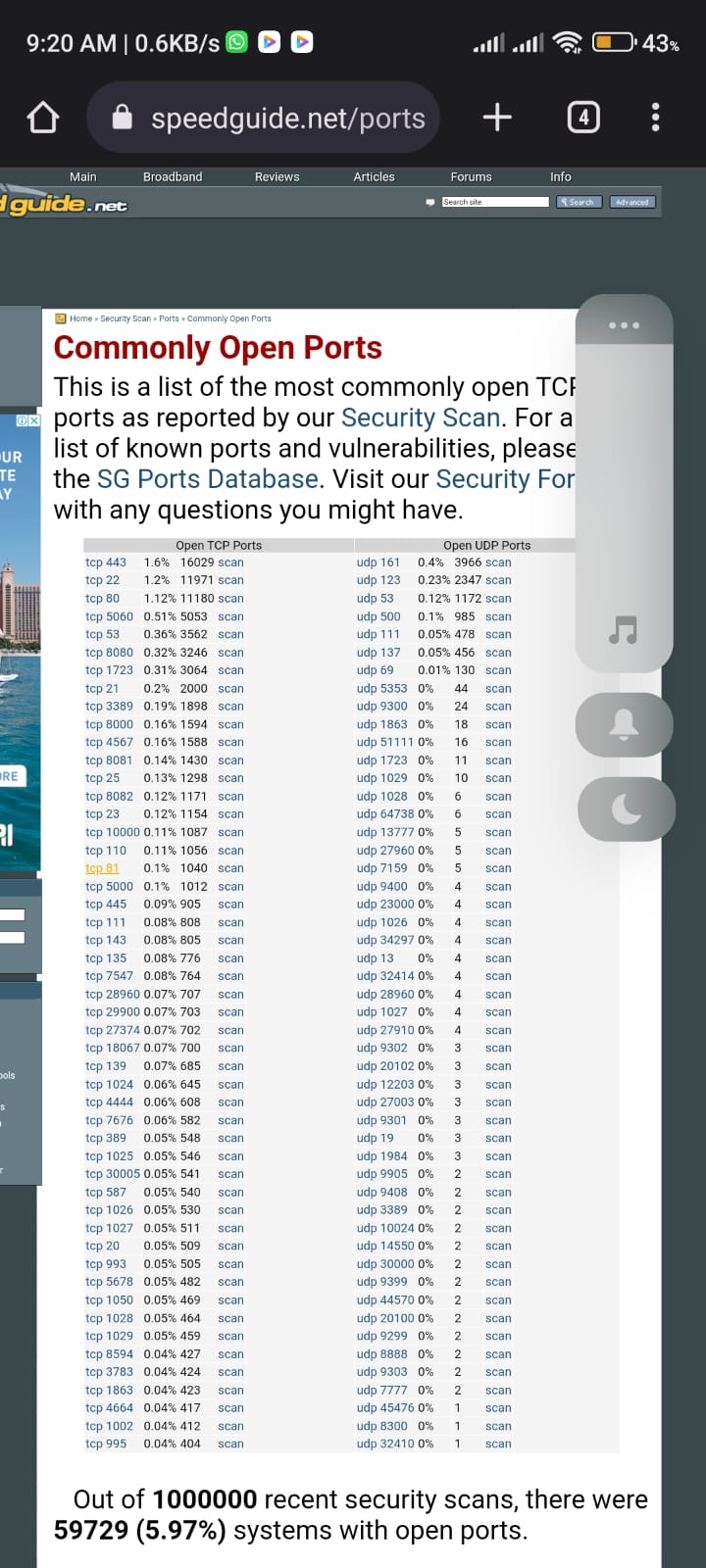
**Ports:** An open port is a network port that accepts traffic either using TCP or UDP and allows communication with underlying server technologies. Open ports are required when hosting remote services to which end-users can connect. There are 65535 total ports of TCP and UDP.

**Common ports open:**

* [**TCP port 21 FTP (File Transfer Protocol)**](https://specopssoft.com/blog/tcp-port-21-ftp-vulnerabilities/) – Provides a way to transfer files between computers that operate on the simple *get*and *put* concepts to either receive or send files to a remote endpoint. FTP was not intended to be a secure means of communication
* **TCP port 22 SSH (Secure Shell)** – The purpose of SSH is to provide administrators the ability to connect to an endpoint over an unsecured network securely
* **TCP port 23 Telnet** – allows interacting with a network endpoint from the command line and is sometimes used as a tool for remote management
* **TCP port 25 SMTP (Simple Mail Transfer Protocol)** – a protocol used to relay mail from email server to email server
* **TCP and UDP port 53 DNS (Domain Name System)**– a protocol used for name resolution across the Internet.  It traverses port 53 using TCP and UDP connections. DNS is responsible for converting the IP addresses that are non-intuitive to the human-friendly domain names typed into a web browser
* **TCP port 110 POP3**– Known as the Post Office Protocol, it is used by email clients to synchronize and download mail from remote mail servers
* **TCP port 145** **IMAP** – Internet Message Access Protocol synchronizes and displays emails without the need to download them
* **TCP ports 80 & 443 HTTP and HTTPS** – HTTP (Hypertext Transport Protocol) and HTTPS (Hypertext Transport Protocol over SSL) are today’s web servers’ standard protocols and ports. HTTP port 80 is the legacy, insecure protocol and port in use, while HTTPS is the secured web server protocol and port used for encrypted web communications. Most organizations have deprecated the use of HTTP across the board as it is clear text and insecure communication
* **TCP port 81** – Commonly used as a web proxy port
* **TCP and UDP port 135, 137, 139** — Windows Remote Procedure Call (RPC) and Windows NetBIOS over TCP/IP are well-known in Windows networking. These communicate over TCP and UDP ports 135, 137, and 139 and historically have many vulnerabilities
* **TCP port 1433 SQL** – Microsoft SQL Server, used throughout many enterprise organizations today communicates over TCP port 1433.
* **TCP port 3306 MySQL** – This port is used for MySQL database communication
* [**TCP port 3389 RDP**](https://specopssoft.com/blog/remote-desktop-protocol-port-3389-vulnerabilities/) – Remote Desktop Protocol (RDP) is used to display remote GUI desktop sessions of a remote Windows computer. It is commonly used with Virtual Desktop Infrastructure (VDI) environments.
* **TCP port 5900** **VNC** – VNC is a tool that is commonly used for remote access administration.  It communicates on TCP port 5900.



**How To Protect A Port:**

When it comes to protecting open ports, the first step is only to use ports that encrypt traffic. It means that an attacker cannot easily capture network traffic and decipher sensitive information. Organizations also want to scrutinize and decide if an open port is necessary. As mentioned at the outset, any open port increases the attack surface and the potential for compromise due to vulnerabilities, misconfiguration, and other factors.

Open ports to the Internet should sit behind a modern firewall or another filtering device to scrutinize traffic connecting to the open port. It ensures the communication is meant to be valid communication with the technology listening on that port or if it has malicious intentions.

Open ports should also be segmented from your internal network in what is known as a DMZ or demilitarized zone. If an attacker compromises the open network port, it is easier to contain the malicious activity if segmented from the rest of the network.

Password-protected services that are listening on an open port require organizations to protect the passwords used to authenticate services on these ports. Aside from never using cleartext protocols, this point emphasizes the need to have [strong password policies](https://specopssoft.com/blog/nist-password-standards/) that prevent weak, breached, or easily guessable passwords.

Last, but perhaps the most important, make sure the underlying technologies and systems listening and answering on the open port are fully patched, and security updates are applied as soon as possible. Attackers often look for unpatched vulnerabilities to compromise an open port. In addition to the strategies mentioned above for protecting open ports, auditing, and other cybersecurity strategies, businesses must enforce strong passwords and use strong password policies.

**SMTP (mail) Port Open:**

**Step 1: Gather IP addresses used to send email**

The first step to implement SPF is to identify which mail servers you use to send email from your domain. Many organizations send mail from a variety of places. Make a list of all your mail servers and their IP addresses, and be sure to consider whether any of the following are used to send email on behalf of your brand:

* Web server
* Your email service provider’s (ESP) mail server
* In-office mail server (e.g., Microsoft Exchange)
* The mail server of your end users’ mailbox provider
* Any other third-party mail server used to send email on behalf of your brand

If you’re unsure of what your IP addresses are, reach out to your ESP to get a list of the addresses associated with your account or your IT System Administrator to compile a list of IP addresses your business uses.

**Step 2: Make a list of your sending domains**

Chances are, your company owns many domains. Some of these domains are used to send email. Others aren’t.

It’s important to create SPF records for all the domains you control, even the ones you’re not mailing from. Why? Once you’ve protected your sending domains with SPF, the first thing a criminal will do is try to spoof your non-sending domains.

**Step 3: Create a SPF record**

SPF authenticates a sender’s identity by comparing the sending mail server’s IP address to the list of authorized sending IP addresses the sender publishes in the DNS record.

Here’s how to create your SPF record:

* Start with v=spf1 (version 1) tag and follow it with the IP addresses that are authorized to send mail. For example, v=spf1 ip4:1.2.3.4 ip4:2.3.4.5
* If you use a third party to send email on behalf of the domain in question, you must add an “include” statement in your SPF record (e.g., include:thirdparty.com) to designate that third party as a legitimate sender
* Once you have added all authorized IP addresses and include statements, end your record with an ~all or -all tag
* An ~all tag indicates a soft SPF fail while an -all tag indicates a hard SPF fail. In the eyes of the major mailbox providers ~all and -all will both result in SPF failure. Validity recommends an -all as it is the most secure record.
* SPF records cannot be over 255 characters in length and cannot include more than ten include statements, also known as “lookups.” Here’s an example of what your record might look like:
  + v=spf1 ip4:1.2.3.4 ip4:2.3.4.5 include:thirdparty.com -all
* For your domains that do not send email, the SPF record will exclude any modifier with the exception of -all. Here’s an example record for a non-sending domain:
  + v=spf1 –all

Congratulations! You’ve created your SPF record. Now, it’s time to publish it.

**Step 4: Publish your SPF to DNS**

Work with your DNS server administrator to publish your SPF record to DNS so mailbox providers can reference it.

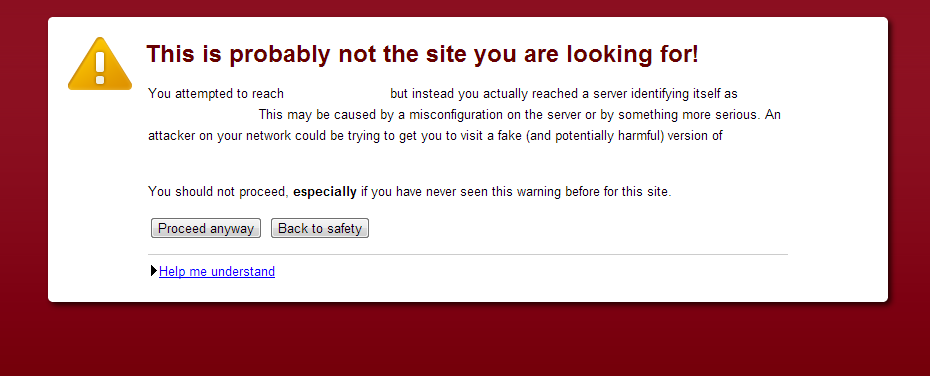
If you’re using a hosting provider such as 123-reg or GoDaddy, then this process is fairly simple. If your ISP administers your DNS records or if you aren’t sure, contact your IT department for support. Email service providers typically publish SPF records for sending domains on your behalf.

**Step 5: Testing SPF records!**

Test your SPF record with an[SPF check tool](https://www.proofpoint.com/us/cybersecurity-tools/dmarc-spf-creation-wizard). You’ll be able to see what recipients see: A list of the servers authorized to send email on behalf of your sending domain. If one or more of your legitimate sending IP addresses is not listed, you can update your record to include it.

**HTTP Open Port:**

You can go ahead and just type ‘HTTPS’ into the address bar on any old site. If the site has a valid certificate, then the server hosting the site will present that certificate to your browser (which has a built-in list of trusted certificate issuers) and nothing really noticeable happens. Simply put, a certificate is the thing that certifies that a site or service is what it claims to be. Depending on the browser, you can click on that little padlock (or its equivalent) to find certain information about the certificate and its issuer. However, if you try to access the ‘HTTPS’ version of a site that does not have a valid certificate, you’ll see an SSL warning like this one:

[](https://media.kasperskydaily.com/wp-content/uploads/sites/92/2013/05/06050924/badsslcert.png)

This warning is just your browser’s way of telling you that it cannot verify the identity of the site’s server you are trying to connect with. You can probably imagine why it’s important to have a certificate system like this to verify that sites and services are who they say they are. If your imagination is failing you, then read up on [man-in-the-middle attacks](https://www.kaspersky.com/blog/man-in-the-middle-attack/).

**FTP open port:**

If FTP port open we send to STP to secure file sharing

Moreover, there are many ports open but we aware that any application port is not open.