The 2023 State of the Internet Report

A Deep Dive into Web Entities, Encryption, and Security Practices on the Internet.

censys

Part One

Introduction

Introduction

The internet is undoubtedly one of the most significant technologies ever invented, having revolutionized the way we communicate, do business, and access information. Since its inception it has evolved to support a vast array of technologies including web servers, content delivery networks, cloud computing, and the internet of things. However the continuous adoption of new technologies and services poses challenges in the realm of security practices. As new vulnerabilities and methods of exploitation emerge, the internet's attack surface continues to evolve, making it crucial for individuals, businesses, and governments to keep pace with the dynamic landscape of internet risks.

Censys maintains a comprehensive map of the internet that offers broad visibility into online entities. In last year's report, we leveraged this data to broadly examine some of the most significant services on the internet, internet-wide risks and vulnerabilities, and the footprint of organizations on the internet. This year, we delve deeper into web entities, or content served over HTTP – think websites, web-based control panels, load balancers, and even APIs. Web entities have become a ubiquitous part of our daily lives, enabling us to shop, read the news, and stay in touch with loved ones.

Our goal is to share our findings and analysis with the community to provide a deeper understanding of the complexities of the internet. We hope that readers can use these findings to enhance their understanding of the services that comprise the web and make more informed decisions about how to safeguard their digital assets.

In particular, we want to better understand the state of security on the modern internet.

First, the good news:

- We know from previous research that over 90% of web traffic is encrypted. Of all hosts we observe using TLS and encrypting traffic, nearly 95% of them support the latest two versions of TLS (1.2 and 1.3), with steady growth over the last year in hosts using TLS 1.3. It's encouraging to see strong adoption of these newer versions of TLS and shifts away from older, less secure versions.

- Let's Encrypt, an organization that provides free TLS certificates, has been a main driver in widespread TLS adoption since they began issuing certificates in 2015. Now responsible for 49% of browser-trusted certificates, Let's Encrypt offers TLS encryption to anyone in a free and convenient manner, a departure from the expensive and time-consuming process of obtaining a TLS certificate in the past.

And the not-so-good news:

- Data exposures via misconfiguration remain a serious problem. *We found over 8,000 servers on the internet hosting potentially sensitive information, including possible credentials, database backups, and configuration files*. These were trivial for us to identify, as they would be for even inexperienced threat actors.
- What's more? Unauthenticated monitoring tools (Prometheus) and API documentation (SwaggerUI) provide another avenue for threat actors to conduct reconnaissance, giving detailed insight into a target's network. We found over 40,000 instances of *each* of these in our data.
- While it isn't *all* bad news, our research suggests that we still haven't reached an ideal state of security on the internet.

We hope you'll enjoy this exploration into web entities and the state of internet security.

 The Censys Research Team

Table of Contents

Glossary

Before we get into the data, we wanted to provide definitions of some common terms we will use throughout this report.

- **Autonomous System (AS):** An AS is a group of hosts with the same routing policy, managed by one network operator or organization. ASes help route traffic across the internet. Each AS receives its own Autonomous System Number (ASN) from a regional internet registry (RIR) such as ARIN, RIPE, or APNIC. These ASNs are used to identify an AS and its associated network prefixes.
- **Certificate Authority (CA):** A trusted entity responsible for verifying the identity of an entity requesting a certificate, signing and issuing the certificate, and ultimately managing the lifecycle of the certificate.
- **Content Delivery Network (CDN):** A Content Delivery Network (CDN) is a geographically distributed network of servers that work together to provide faster and more reliable delivery of web content, including text, graphics, scripts, and videos. By storing cached copies of content closer to end-users, CDNs can help reduce latency and bandwidth consumption, leading to faster load times and improved performance for websites and web applications.
- **Host:**
  - Unnamed Host: A distinct IPv4 address without a corresponding hostname
  - Named / Virtual Host: A distinct IPv4 address or collection of IP addresses that represent a single hostname
- **HTTP:** Hypertext Transfer Protocol (HTTP) is a protocol that facilitates the exchange of data between web servers/websites and web clients.
- **HTTP Status Code:** A numeric code that indicates the status of a requested resource on a web server.
- **Service:** An application running on a host that can communicate with a client over a network. These services are typically identified by the communication protocol used at the OSI-model L7 (application) level. However, Censys also identifies and isolates specific services that operate on top of HTTP, such as Elasticsearch, CWMP, and others.
- **TLS / SSL / Encryption**
  - Unencrypted service: A plain HTTP service without TLS encryption or HTTPS redirection, leaving data transmission over the internet vulnerable to interception, tampering, and unauthorized access.

    - Encrypted service: An HTTPS service utilizing TLS encryption, which safeguards data transmission over the internet by encrypting it and leveraging digital certificates for authentication

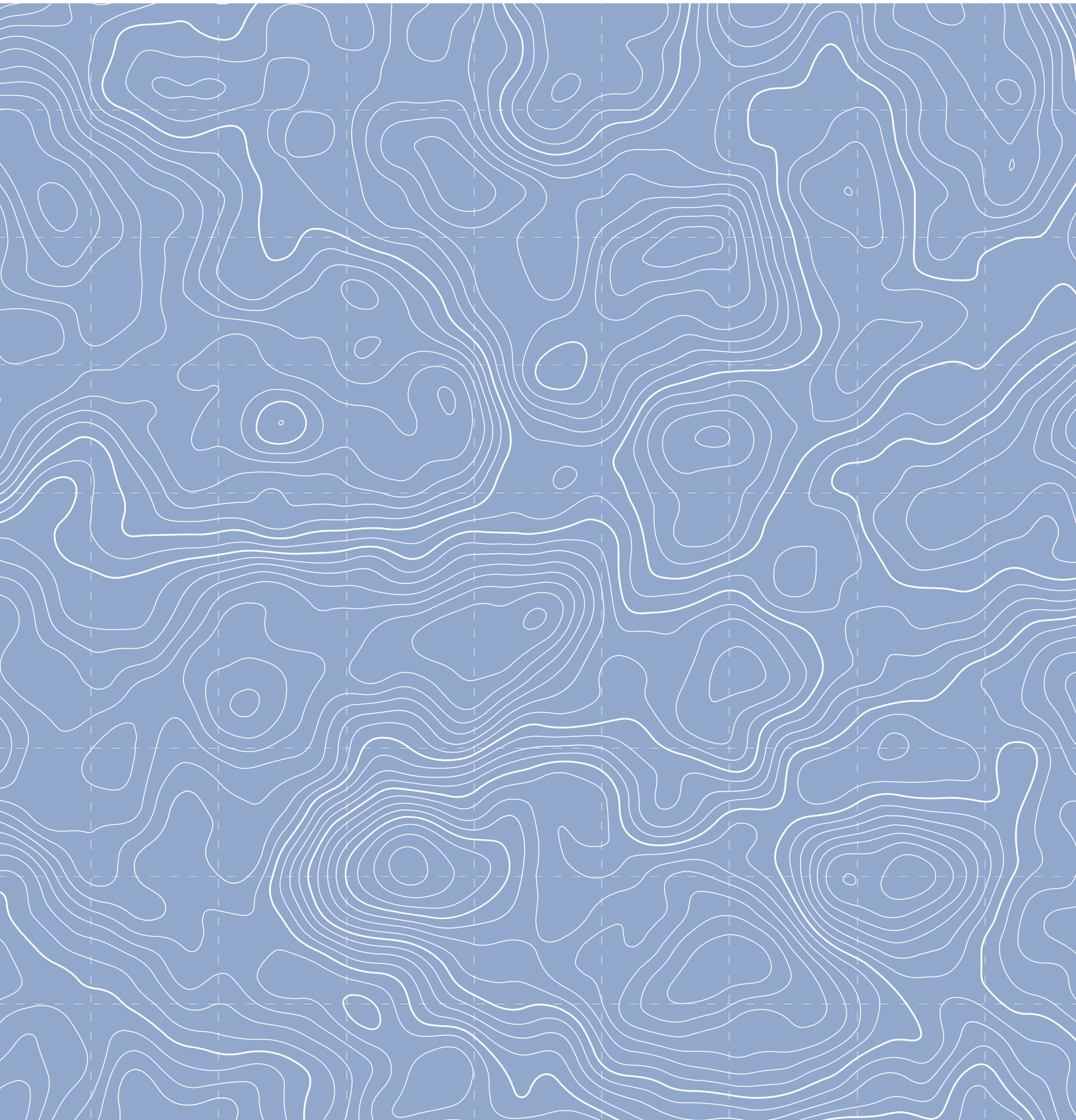
- **Web Entity:** A service running HTTP on the internet that is accessed by IP or by a hostname.
- **X.509 / SSL / TLS Certificate:** Known by several names, certificates are digital artifacts that verify an entity's identity and enable web traffic encryption.

Summary of Findings

- **While misconfigurations don't often make headlines, they remain a major problem.** We identified over 8,000 hosts on the internet hosting various database information, backup files, passwords, Excel worksheets, environment variables, and even some SSL and SSH private keys. These were simple to find and could make a threat actor's job very easy.
- We observed **over 40,000 Prometheus servers exposed to the internet, monitoring over 219,000 endpoints**. This tooling could provide would-be threat actors with detailed reconnaissance and network mapping abilities.
- When examining internet exposure of the monitoring tool Prometheus, we discovered that over 48% of the active metrics on the hosts and services Prometheus monitors **exist *exclusively in private IP space and internal DNS zones*, which would typically not be visible to global internet users**. This visibility would make it trivial for threat actors to conduct reconnaissance on these organizations, learning details about their inner workings and network architecture.
- The popularity of web servers with a history of vulnerabilities or those that have reached end-of-life on unnamed hosts (Hikvision, Boa) **may indicate security practices that have fallen behind on the necessary best practices for the modern internet**.
- **Of all encrypted HTTP services, nearly 95% support the most secure TLS versions available:** 1.2 and 1.3, and the adoption of TLS 1.3 has been in a consistent state of growth over the past year.

Part Two

What *is* the Internet?



What *is* the Internet?

While the terms "internet" and "web" are often used interchangeably, they are in fact distinct concepts. The internet refers to a network-of-networks that connects devices worldwide through physical cables and wireless connections. Conversely, the web is an enormous collection of data accessible via the internet, running on HTTP (Hypertext Transfer Protocol), a simple yet increasingly sophisticated protocol that runs over TCP.

This report defines "web entities" as services running HTTP on the internet that are accessed by IP or by a name. While most users access the web through browsers that request content from domain names, this is only one part of the whole picture.

> *Censys has two perspectives on the internet: the* **unnamed** *and the* **named**. *The unnamed internet view consists of hosts and services that respond equally to requests via IP addresses or hostnames, such as SSH servers. Conversely, the "named internet" is the hosts and services that Censys can view independently of the physical IP and are instead referenced by a name.*

Fortunately, two of the most common protocols found on the internet support such a mechanism, albeit for slightly different reasons:

- The HTTP protocol (starting in version 1.1) specifies that a "Host" header must be included with each client request, informing the server of the specific hostname along with the resource that is being requested. Without this header, every domain name would need a dedicated IP address.
- TLS SNI (Server Name Indication) is an extension of the Transport Layer Security (TLS) protocol that allows a client to "indicate" the hostname of the server it is trying to connect to before establishing a secure connection. Without SNI, the server could not determine the correct hostname and associated underlying certificate and would return whatever default certificate the server had configured – without SNI, an SSL certificate would need a dedicated IP address to function securely, much like it did until 2003.

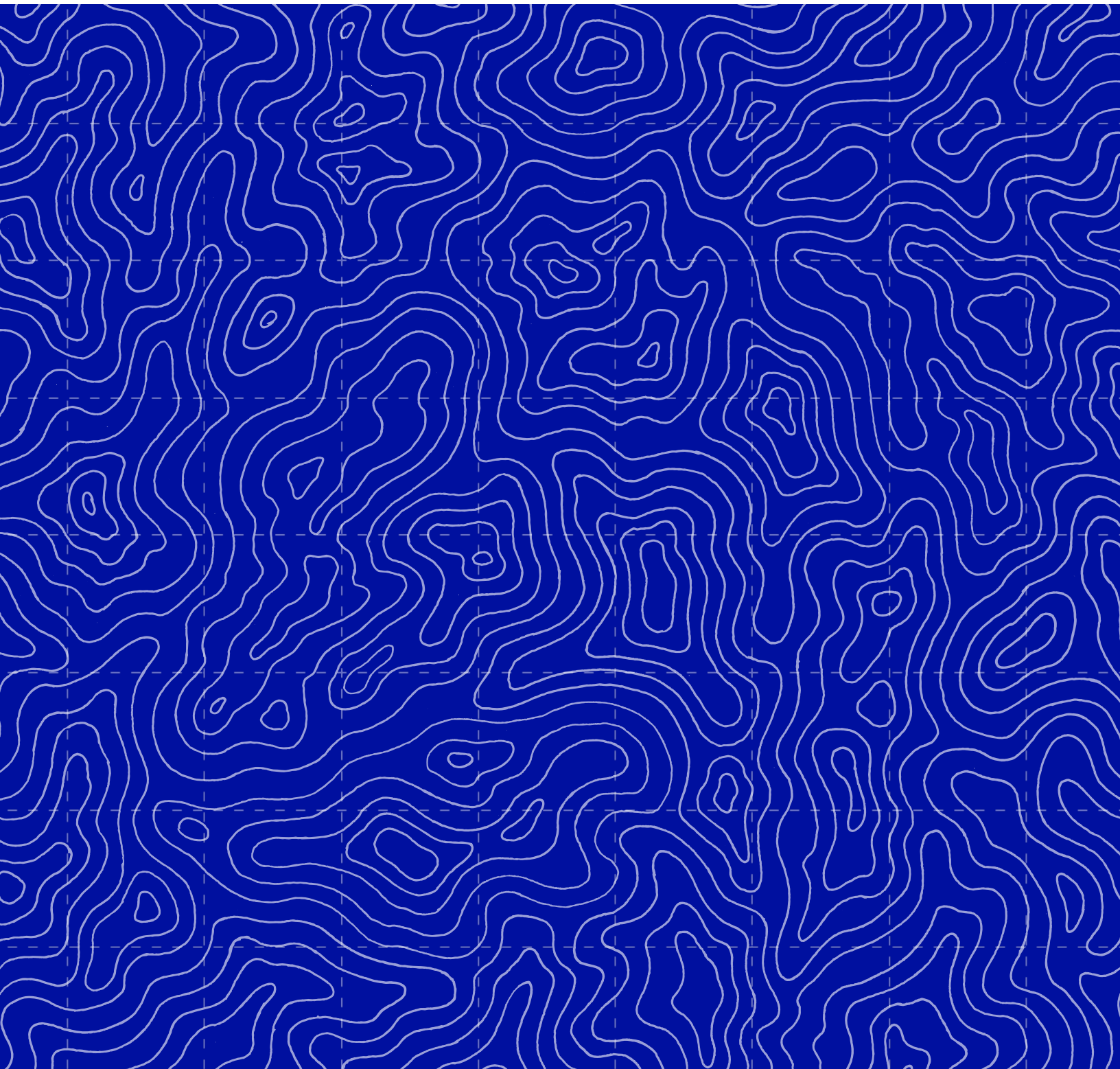
In summary, SNI allows the web server to respond with a hostname-specific certificate, while the HTTP Host header maps the request to a different backend entity, such as a file or directory, a process commonly referred to as "Virtual Hosting."

Given that most web servers will respond differently based on the client's request, if Censys only scanned the world using the bare IP address of hosts, we would have a minimal picture of what the internet genuinely looks like, and our data would be wholly incomplete. This is why we introduced name-based network scanning and a more modern approach to viewing web-based assets, which we call "web entities."

The following sections will attempt to paint a complete picture of HTTP on the modern web.

Part Three

Web Entities on the Internet



Web Entities on the Internet

HTTP At-a-Glance

HTTP, or Hypertext Transfer Protocol, encapsulates many different types of services running on the internet. Services that run over HTTP include web servers, load balancers, web-based APIs, and more. In our previous report, we discussed how HTTP is ***everywhere***. It represents 88% of the services we see on the internet.

On a single daily snapshot of the internet from Censys scan data in early 2023, we observed over 740 million hosts running 1.3 billion HTTP services of some variety. This comprises 165 million unnamed hosts only accessible by bare IPv4 address and over 570 million named or virtual hosts.

Of these services, we observed:

- Nearly 18% running on servers hosted in a major cloud provider (Amazon, Oracle, GCP, or Azure)
- Examining the internet as a whole (beyond the major cloud providers), we observe over 85% of HTTP services running on named hosts, while 14% were running on unnamed hosts
- Web server technologies running on over half of all HTTP services on both named and unnamed hosts, with Apache HTTPD and Nginx being the most popular
- Instances of discontinued and/or scrutinized web server products commonly used on unnamed hosts suggests that non-optimal security practices may be in place on these devices.

Who's Down with HTTP?

GEOGRAPHIC VIEW

Internet hosts are dispersed globally, with a higher concentration in regions with denser populations and advanced digital infrastructures, particularly those housing large data centers. Consequently, the United States, China, and various European countries have some of the most substantial numbers of internet hosts that surpass those of other regions worldwide.

The hosts on the internet running HTTP services mirror general host distribution patterns we observe globally.

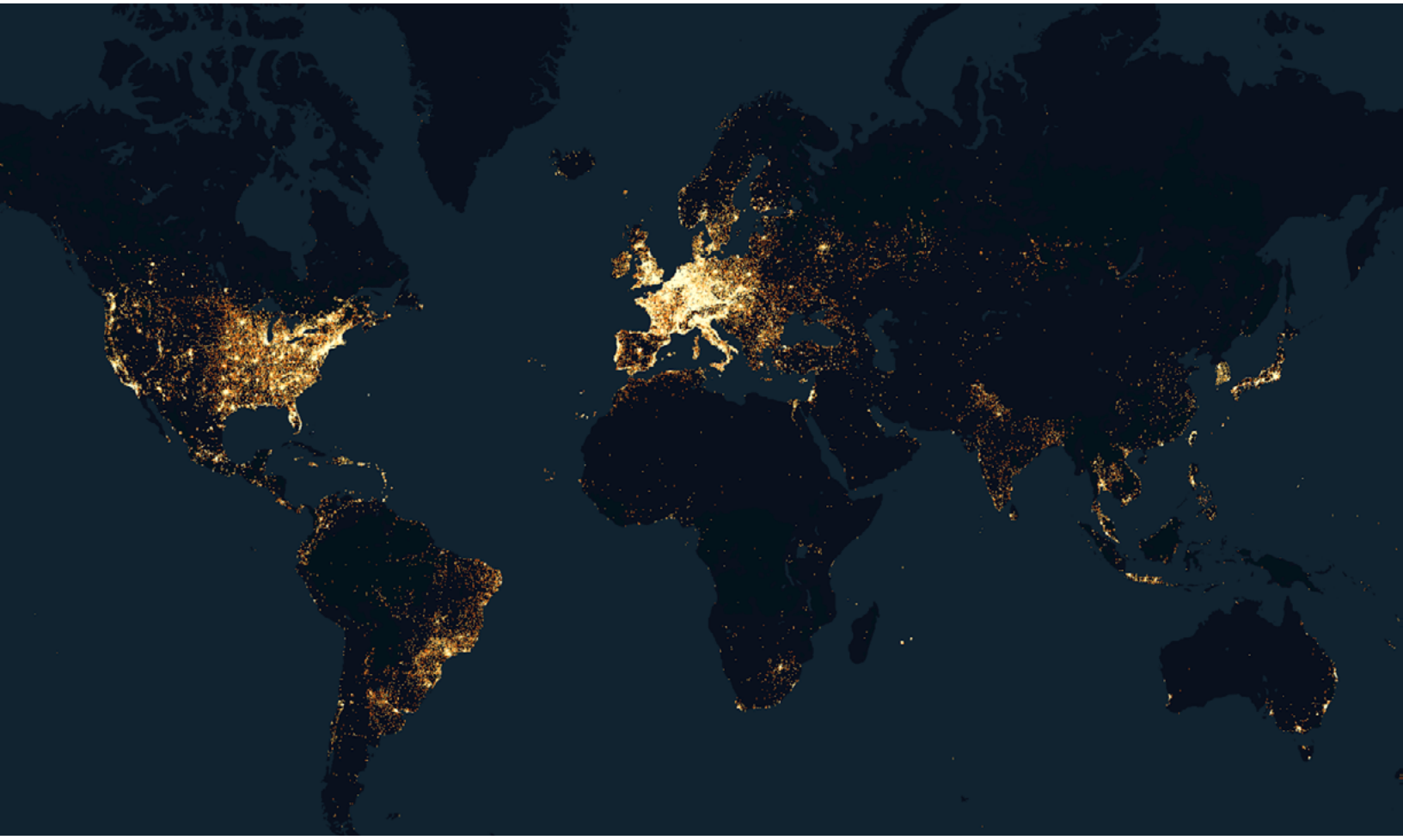


Figure 1: *Snapshot of all hosts running an HTTP service from February 28, 2023 (generated via kepler.gl)*

HTTP IN THE CLOUD

Cloud infrastructure is increasingly being adopted in today's digital landscape, powering the backbone of modern businesses and organizations worldwide. So just how much of the web is on the cloud? While it is difficult to determine the exact percentage of the web that is on the cloud, we can gain a general idea by examining the web infrastructure of the dominant players in the cloud provider industry: Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure, and Oracle Cloud Infrastructure (OCI).

Censys observed over 130 million hosts globally (~18% of all hosts running HTTP on the internet) with nearly 230 million HTTP services in one of these four major cloud providers.

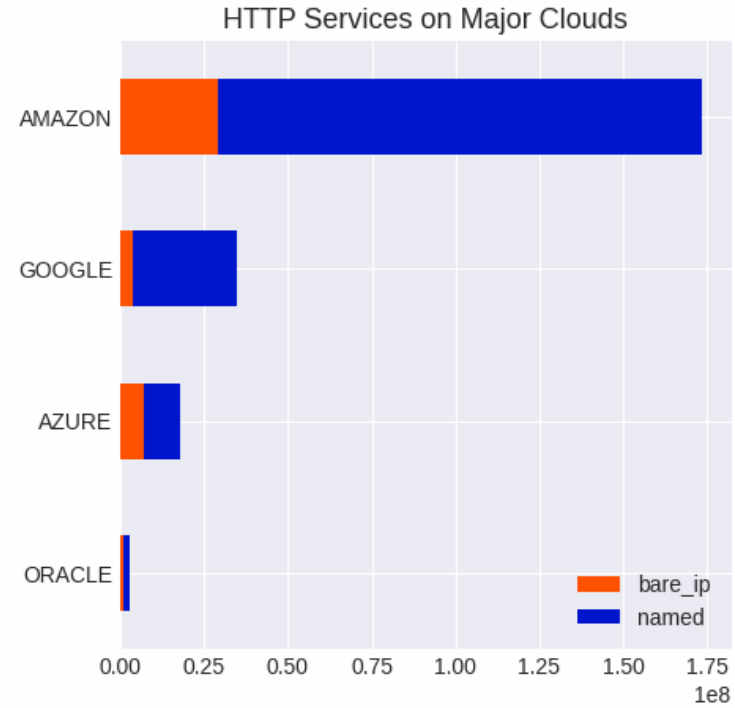


Figure 2: *Breakdown of HTTP services across major cloud providers*

Amazon remains the most popular cloud provider we observe among HTTP services, accounting for 13% of all HTTP services. Zooming out, services running in AWS make up 11% of all the services Censys observes. These include popular AWS services such as EC2 and the web apps and web properties behind its CloudFront CDN.

Amazon also sees the most significant number of HTTP services running on named hosts, with over 80% of all its HTTP services being named.

What Comprises HTTP Services on the Internet?

How many of the HTTP services on the web are actual websites? The answer is more complex than it may initially seem. A "website" – in the context that many internet users are familiar with – is a collection of publicly accessible documents that share a common name. Websites are typically published on web servers that speak HTTP to serve content to web clients.

Defining what constitutes a "website" requires some nuance, as not everything deployed on web servers fits this definition. For instance, we can find many web servers that appear to have no function - they may return zero-byte documents, blank pages, or simply a 404 status code indicating no data can be found. Although these are web servers serving web content, they do not necessarily qualify as websites. The same applies to web servers that serve data designed to interact with other computers, such as API endpoints and content forwarders.

To better understand the distinction between a web server and a website, it may be helpful to examine the various products and technologies associated with web servers. Our data provides visibility into over 4,800 products, technologies, and applications running over HTTP, including not only the web servers but also load balancers, APIs, proxies, and more.

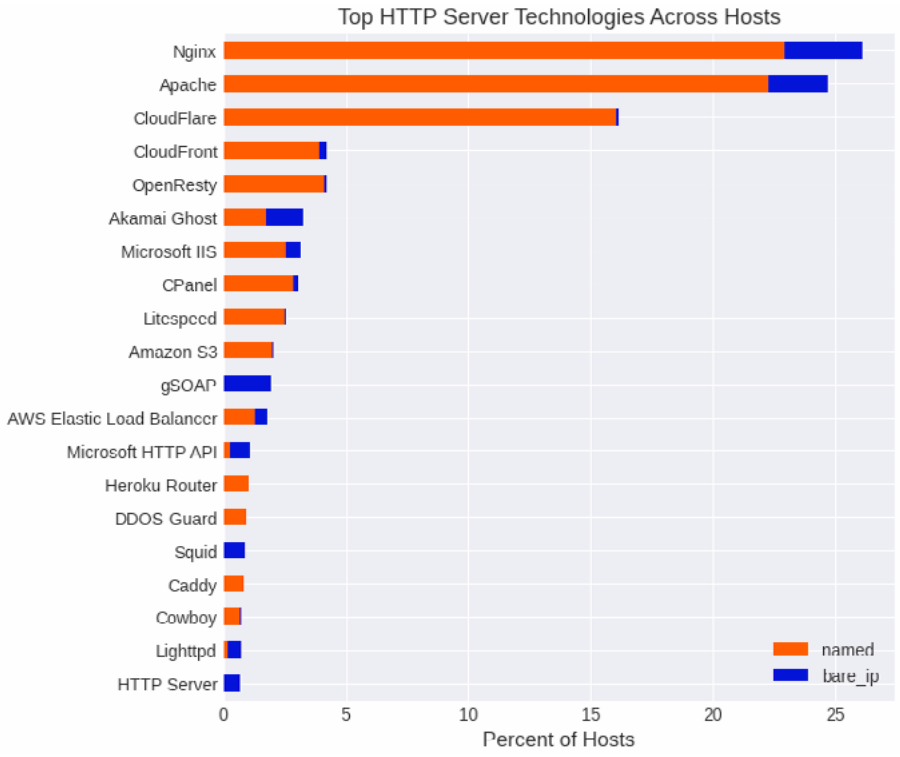


Figure 3: *Top 20 HTTP server technologies observed on hosts*

By examining a higher-level breakdown of the HTTP server technologies across all hosts, we can start to understand the scope of the technologies working alongside HTTP to serve web content.

Of the top 25 technologies we observed, web servers are the most common resources that we see. Comparing the specific web server products running on named versus unnamed hosts uncovers interesting potential differences in security practices.

In total, Apache's HTTPD and Nginx are the most common web server technologies on the internet as a whole, with unnamed hosts accounting for 3% of these while named hosts account for over 20%

Other popular web server technologies include Microsoft IIS (Internet Information Services), LiteSpeed, and two well-known web servers based on Nginx: OpenResty and Tengine.

When we dig deeper into other web server products beyond the top 20, particularly those running on unnamed hosts, we note some technologies that have concerning security implications: specifically, Hikvision and Boa web servers.

Hikvision web servers are web interfaces used to manage Hikvision video surveillance products, including network cameras and video recorders. In 2021 a critical unauthenticated command injection vulnerability was patched in Hikvision firmware. This vulnerability, CVE-2021-36260, allowed threat actors to gain administrative-level access to affected devices. Censys sees 1.5 million hosts running Hikvision web applications, with nearly 80% of these observed as unnamed hosts. There are *potentially* tens of thousands of Hikvision devices worldwide that are still vulnerable.

In 2021 a critical unauthenticated command injection vulnerability was patched in Hikvision firmware. This vulnerability, CVE-2021-36260, allowed threat actors to gain administrative-level access to affected devices. **There are potentially tens of thousands of Hikvision devices worldwide that are still vulnerable.**

Boa is an open-source web server primarily designed for embedded applications but has since been discontinued. However, Boa servers remain in use, and unfortunately, they suffer from various known vulnerabilities. Additionally, the absence of fundamental built-in security measures, including access controls, requires the underlying application to handle all security features. Censys detected one million of these servers running exposed to the Internet.An ongoing attack on critical Indian electrical grid assets detailed by Recorded Future was found by Microsoft researchers to be targeting exposed IoT devices running Boa web servers.

Boa is an open-source web server that was primarily designed for embedded applications but has since been discontinued. Boa servers remain in use, and unfortunately, they suffer from various known vulnerabilities. **There are over a million of these exposed servers currently running on unnamed hosts.** *An* **ongoing attack on critical Indian electrical grid assets** *was found to be targeting exposed IoT devices running Boa web servers.*

WHAT'S OUT THERE BEYOND WEB SERVERS?

Many other technologies we see running on HTTP services, while not directly serving web content, work to support content delivery.

One example is load balancers, another significant technology we see running on HTTP. These technologies help distribute incoming web traffic across multiple backend servers to prevent any single system from becoming overwhelmed. Load balancers are often used to help applications run smoothly amidst spikes in traffic and server failures. They can also provide protection against various types of cyber attacks, including distributed denial of service (DDoS) attacks, by effectively blocking potentially malicious traffic.

The most commonly used load balancers on HTTP services we see in our data include CloudFlare, CloudFront, and AWS Elastic Load Balancer, with the latter two being Amazon products.

Besides web servers and load balancers, there are other types of software products commonly used on HTTP services, such as web hosting admin panels, CDN servers, and web proxies. In this overview, we will highlight the most popular software products in each of these categories found in our data:

Web Hosting Admin Panels: Web hosting admin panels, also known as control panels, provide a web-based interface for users and administrators for web hosting providers to manage their servers. In other words, they are specialized web-based software for managing other web servers but don't directly serve website content. Several types of admin panels are available, but some of the most popular ones we see in our data are cPanel and Plesk. These are both more commonly found on named hosts running HTTP.

Web Proxy Servers: Web proxy servers act as intermediaries between end users and the web services they access on the internet. Caching proxies store popular web content to improve delivery speed and reduce bandwidth, while reverse proxies function as gateways that handle incoming requests to a web server. The most popular web proxies we see in our data is Squid, which is primarily used as a caching proxy. Squid is much more prevalent on unnamed hosts.Note that Nginx can also be used as a reverse proxy.

CDN Servers: We also see specialized proxies in the form of Content Delivery Network (CDN) servers, also known as edge servers. Edge servers are globally distributed caching proxies that CDN providers deploy to improve website load times and reduce costs. They can also handle other tasks like load balancing, SSL termination, and security filtering for the web. CDNs serve a large portion of the internet content today, particularly for applications that include video streaming and software downloads. The most common CDN edge server we see running on HTTP is Ghost, developed by Akamai, one of the leading CDN providers.

Cloud Storage and Bucket Services: Cloud storage and bucket services are remote data storage solutions from cloud providers. In our data, this mainly comprises Amazon S3 buckets, which are found more commonly on named hosts. S3 is commonly used for big data analytics and storing application data such as images and videos for websites, user data, and database backups.

Web-Based APIs: Web-based APIs are widely used in web and mobile applications to allow different systems to communicate with each other over HTTP and integrate popular services. The top web-based API we observed in our data was Microsoft's HTTP API, which allows developers to build applications that can speak to various

Microsoft software products such as Microsoft Office 365 and Azure. We mostly observed this API on unnamed hosts running HTTP.

In addition to examining software products and technologies, another perspective on what runs on an HTTP service is the content it returns to client requests, commonly referred to as the response body. Our analysis of the types of content found on HTTP services reveals that the overwhelming majority of responses are in HTML. This suggests that most HTTP services, whether on unnamed or named hosts, are websites or website-adjacent.

Beyond HTML, however, we discover some intriguing examples of more specialized content on HTTP services:

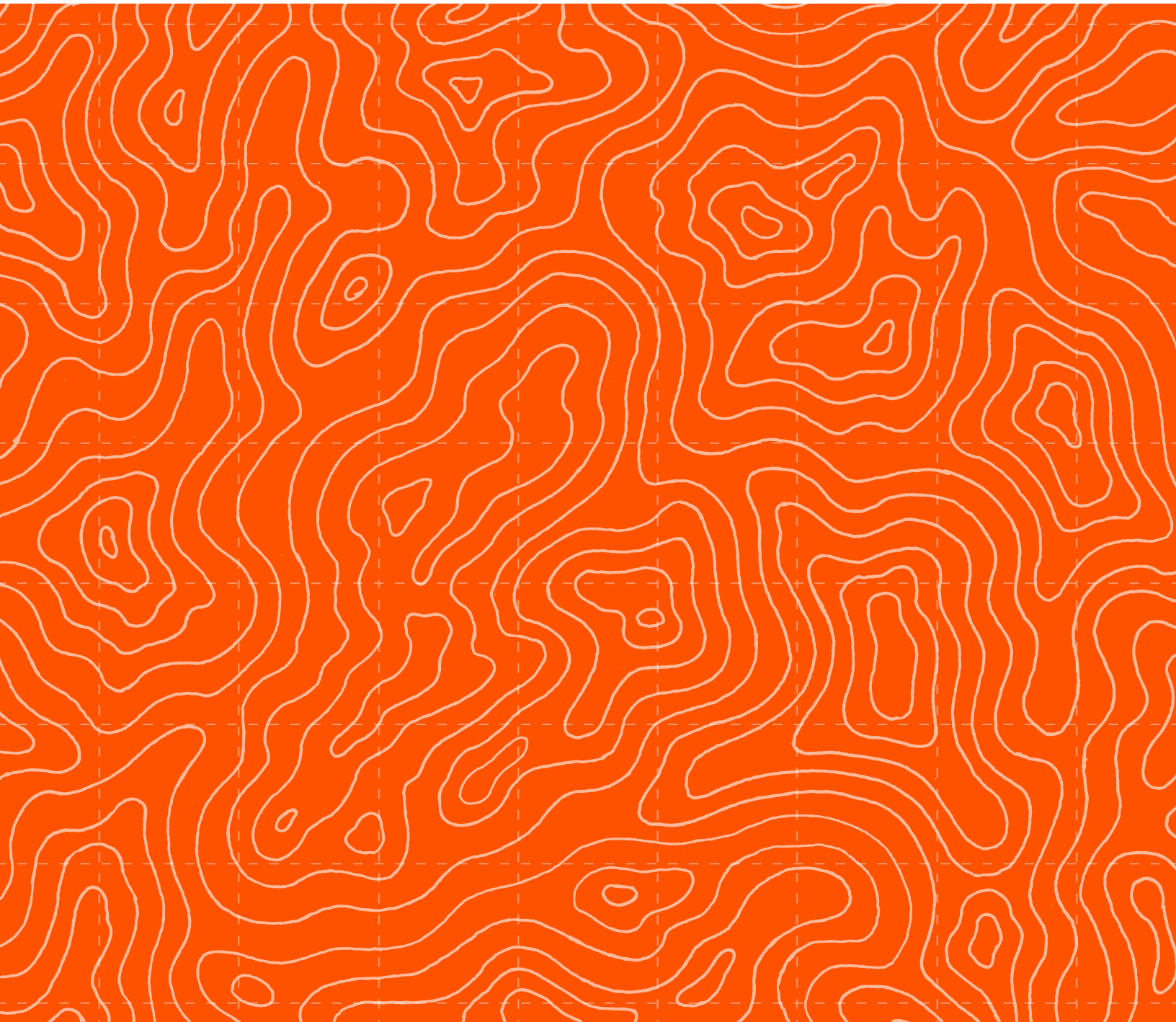
Content Type	Unnamed Service Count	Named Service Count
ASCII Text	10.6M	44.4M
Unicode Document	1.1M	543.2K
XML 1.0 Document	729.5K	81.5K
JSON Text Data	246.8K	119.2K
Exported SGML Document	49.6K	117.6K
GIF Image Data	10.4K	13.3K
Smile Binary Data	1.2K	2.6K
PDF Document	412	13981
MS-DOS Executable	164	2159
SVG (Scalable Vector Graphics) Image	123	5652

We observe other types of markup languages here, such as XML and SGML, which are similar to HTML but designed for different purposes. XML was written to transport data with less emphasis on how that data is presented. SGML is a language for defining other languages. Further, we encounter other types of machine-readable data, such as JSON data and Smile, which is a binary format based on JSON. These are likely reflections of various services that exchange data over HTTP but may not necessarily qualify as websites or be designed for human consumption.

Although not exhaustive, this provides a glimpse into the wide range of technologies and datatypes that can be found on HTTP. **While a significant portion of HTTP involves websites or related technologies, it's apparent that maintaining the web requires more than just web servers running websites.**

Part Four

X.509 Certificates and the Quest for a More Secure Internet



X.509 Certificates and the Quest for a More Secure Internet

We interact with certificates daily while using the modern internet. They're now such an integral part of the secure internet infrastructure that we rarely think about them unless they expire or are missing from a website. X.509 certificates are often known as SSL or TLS certificates; X.509 refers to the standard that defines the format for public key certificates.

X.509 certificates (we'll just call them "certificates") serve two primary purposes:

1. **Certificates enable encryption for web traffic** – This means threat actors can't easily intercept and read the data passed between a client and server. There was a time when one could be on the same broadcast domain as another user, fire up a network sniffer, and effortlessly monitor unencrypted web traffic. The ubiquity of certificates has made this an increasingly rare and largely unsuccessful practice.
2. **Certificates act as identity verification** – When an entity obtains a certificate from a trusted certificate authority (CA), it must provide evidence of its identity. In some cases, this means providing proof of ownership of a domain (DV) or a meeting with an employee from the organization requesting the cert (EV). When a site presents a certificate from a CA, it's evidence that they have successfully verified their identity and are who they say they are.

While certificates have provided an additional layer of security on the web, the presence of a certificate on a website should not be confused with its legitimacy. Browsers display a padlock icon to indicate that a site is served over HTTPS, which could lead users to believe that the site is secure. However, threat actors have exploited this common misconception by creating certificates for phishing sites to help them appear more legitimate. While free certificate services like Let's Encrypt aim to democratize the use of certificates, they have also enabled bad actors to obtain legitimate-looking certificates for malicious purposes.

Today, much of the web has shifted toward Domain Validation (DV) certificates over Organization Validation (OV) and Extended Validation (EV) as a result of several factors:

- Perhaps most importantly, DV certificates can be issued via an Automatic Certificate Management Environment (ACME), making it much less likely that a

domain's certificate protection will lapse sans human intervention. Chromium reports that over 50% of certificates issued by web PKI rely on ACME, while 95% of web PKI-issued certs come from CAs with some form of ACME available.

- In the past, web browsers would display an organization's name in the URL bar alongside a padlock icon to provide an extra layer of validation and security. However, this practice has been largely discontinued, and currently, unless a user explicitly checks a website's certificate, all types of certificates, including DV, OV, and EV, are presented identically.
- DV certificates are freely available from providers like Let's Encrypt, providing a minimal barrier to adoption, even for small personal or hobby sites.

As of March 2023, our certificates dataset has archived over ***6.7 billion certificates****. We use Mozilla's NSS root store to find that over* ***514 million are browser-trusted****, 4.4 billion were previously trusted, and 1.8 billion are not and never have been browser trusted (i.e., self-signed certificates).*

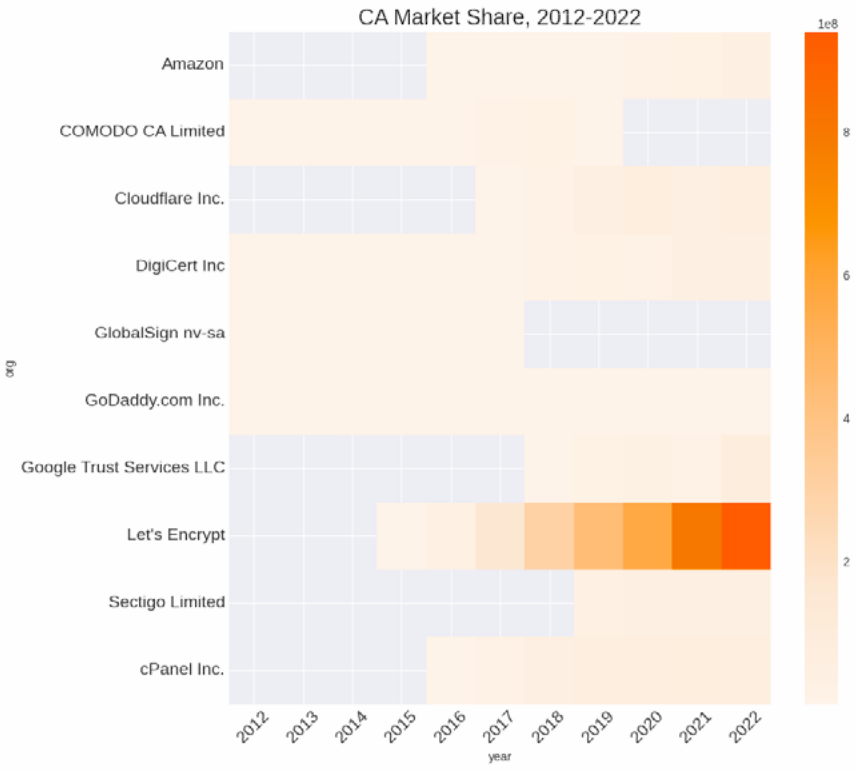


Figure 4: *Time series heatmap of CA market share over time; all other CAs quite literally pale in comparison to Let's Encrypt even only several years after they began issuing certificates. Grey regions on the heatmap represent times when CAs named along the y axis were/are not issuing certificates.*

Examining CA market share over the last 10 years, we can see that it has shifted dramatically. GoDaddy.com and DigiCert are the only major CAs from 2012 who remain in the top 10 in 2023, though with much less market share.

Let's Encrypt has significantly impacted internet security and HTTPS adoption since its founding in 2014. By offering free and easy access to X.509 certificates, Let's Encrypt has made a major contribution to the democratization of certificates for individuals and organizations. Let's Encrypt is the largest CA, representing over 49% of all browser-trusted certificates in our data.

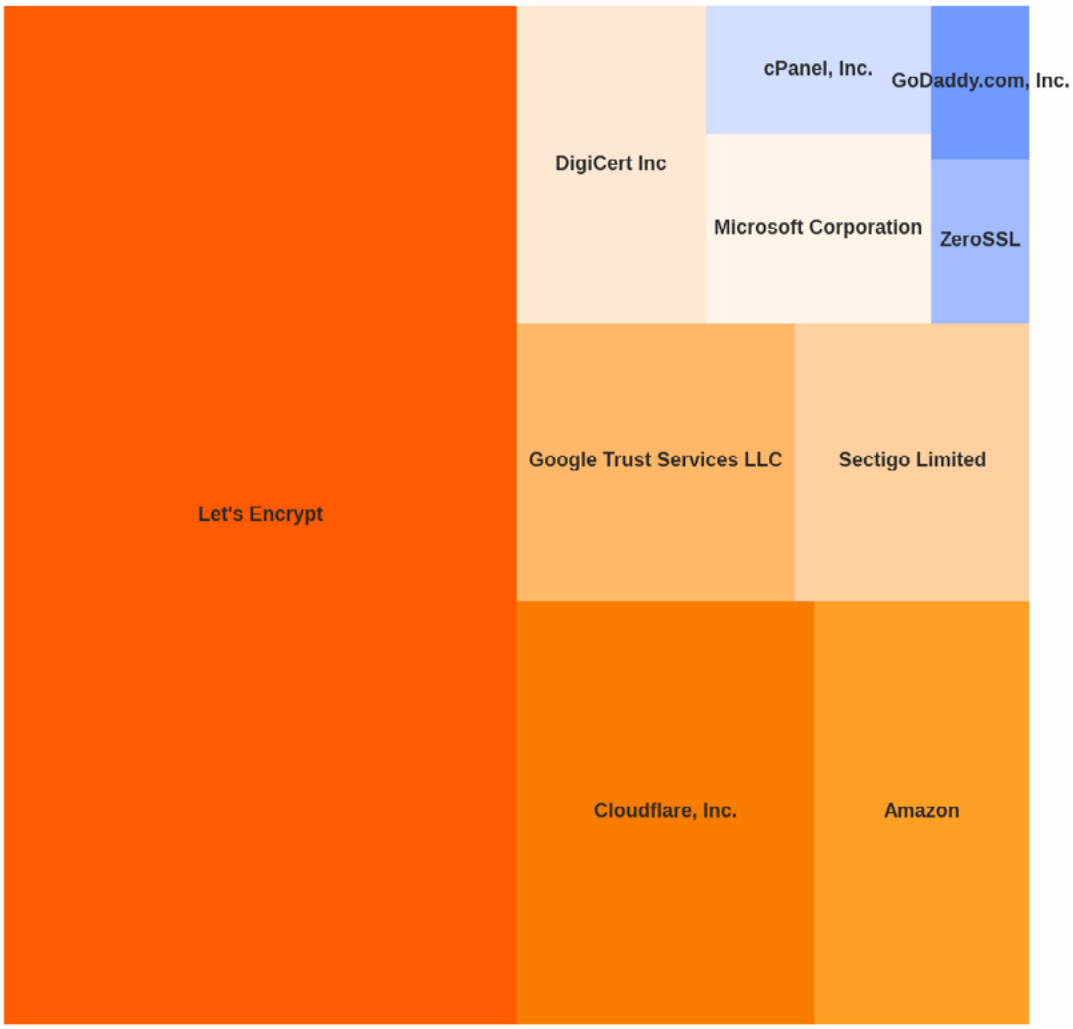


Figure 5: *Treemap illustrating market share of the top 10 CAs who have issued certificates that are currently browser-trusted, March 2023.*

Encryption of Web Entities on the Internet

Before HTTPS, most web entities communicated over plain text HTTP, which meant that any data exchanged between a web client and web server – including users' sensitive information and passwords – was transmitted in unencrypted transports and could be intercepted by a third party via man-in-the-middle attacks. Certificates are instrumental in the widespread adoption of HTTPS and provide the basis for establishing trust between the web client and the web property and protecting sensitive information exchanged between the two.

Although HTTPS has become the norm for web entities that handle data, sensitive or not, many unencrypted HTTP services still exist online. Furthermore, even though TLS version 1.3 was ratified in 2018, its predecessors are still used to serve content. As with anything on the internet, enhancing security and privacy requires time and effort.

Comparing HTTPS adoption on web entities running on named versus unnamed from a snapshot in early 2023, we see a more significant proportion of TLS services on named hosts. Of all the HTTPS services we see, over 85% of them are running on named hosts.

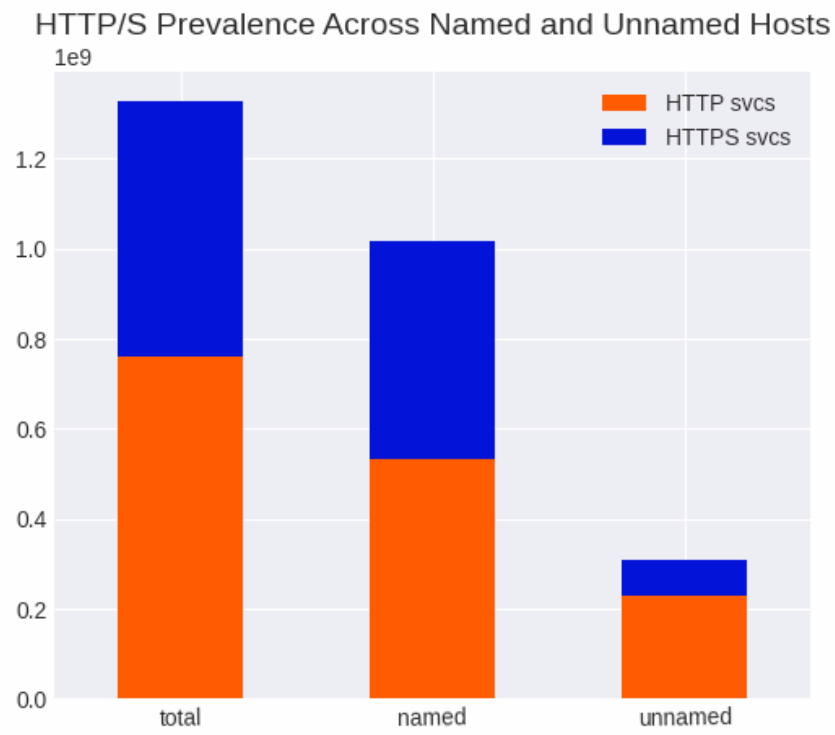


Figure 6: *Encryption (HTTPS) adoption across named and unnamed hosts*

Encrypted Services

Of all HTTPS services we observed on the internet, the vast majority were running TLS 1.3 and 1.2, which are widely accepted as the two strongest versions currently available.

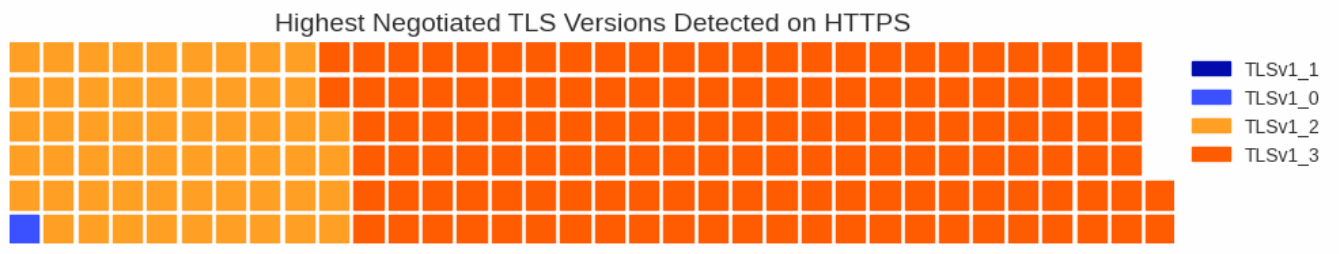


Figure 7: *Percentage of HTTPS services by highest negotiated TLS version detected*

Of these, less than 1%, which accounts for over 3 million services, negotiated at TLS versions 1.0 or 1.1. These versions have been considered deprecated since 2021. It should be noted that TLS 1.1 is now so rare that it doesn't even appear on our graph above (**Figure 7**). Below are the top ASes where these services are still negotiating TLS versions older than 1.2:

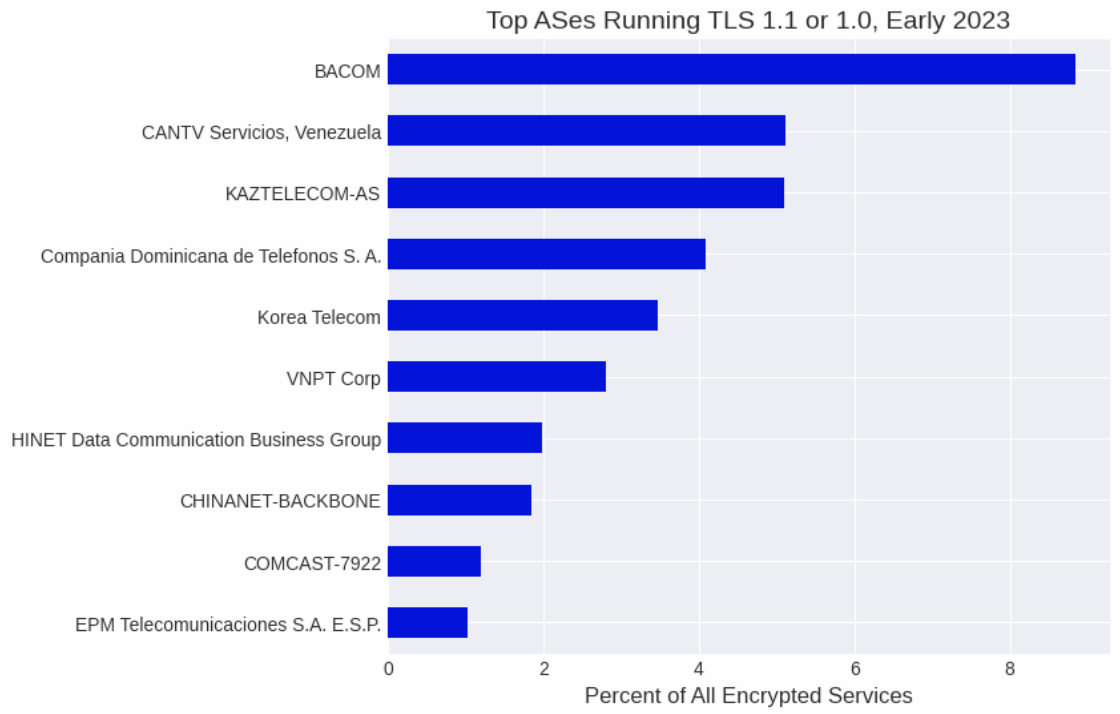


Figure 8: *Top autonomous systems by the count of how many HTTP services they host with TLS 1.1 or 1.0 as the highest negotiated TLS version*

There are various reasons why some organizations may still use older versions of TLS for encrypting their services, including the need to support legacy systems that are not compatible with newer TLS versions and cost and resource constraints associated with upgrading their infrastructure. However, it's important to note that using outdated TLS versions can pose security risks. To ensure the highest level of security for HTTP services, it is generally recommended to use the latest TLS version available and to regularly update and patch systems to mitigate potential vulnerabilities.

Since early 2022, Censys has observed just over a 20% increase in the proportion of HTTP services capable of negotiating with the highest version of TLS, TLS 1.3, and a 30% decrease in negotiating with TLS 1.0.

TLS Version	Proportion of all HTTPS Services in 2022	Proportion of all HTTPS Services in 2023	Delta
TLSv1_0	3.89%	2.70%	-31%
TLSv1_1	0.69%	0.62%	-10%
TLSv1_2	62.64%	57.18%	-9%
TLSv1_3	32.79%	39.71%	21%

HTTP Services by TLS Version

Unencrypted Services

Censys observed that nearly 60% of web entities were running unencrypted HTTP, meaning we did not detect the use of SSL/TLS. While this statistic may be concerning at first, it's important to note that this is a view of all HTTP services we see on the internet, not all of which are user-facing websites.

In many cases, browser requests to an unencrypted HTTP service will be configured to redirect to a corresponding HTTPS service. This occurs for many reasons, but a common one is to provide backward compatibility for legacy applications.

We can get an initial picture of how these unencrypted services are behaving by taking a look at the status codes they return to client requests.

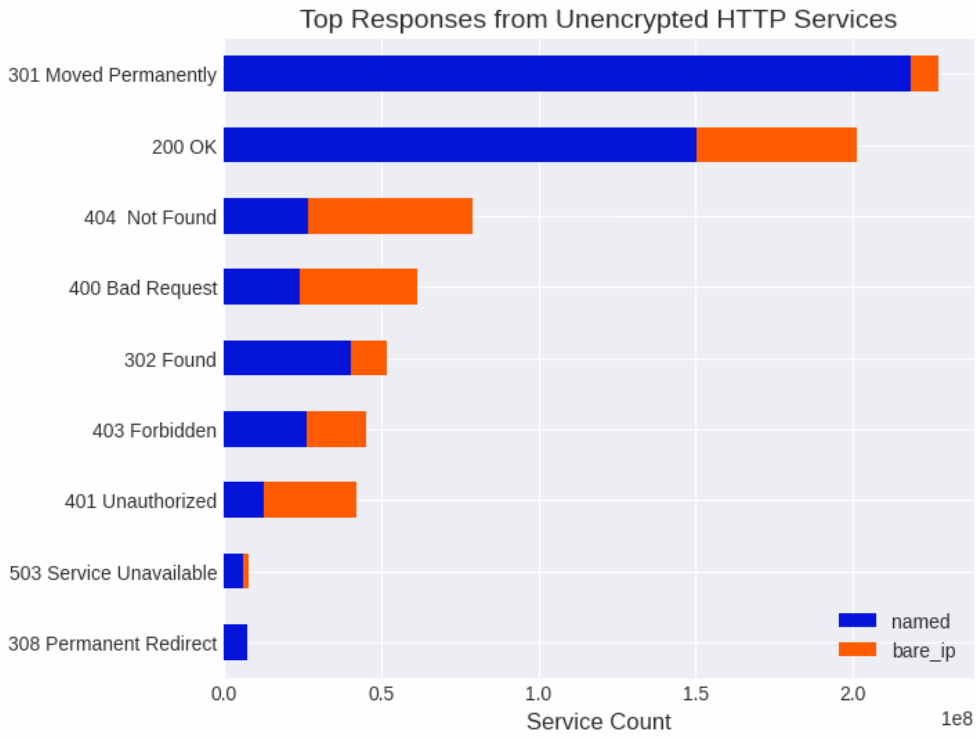


Figure 9: *Top HTTP response status codes observed on unencrypted services*

Some of the most common response codes we see here indicate that many unencrypted services redirect to other services. In particular, redirection messages returning statuses of 301, 302, and 308 make up nearly 40% of all responses from unencrypted services.

It's important to note that the data presented comes from Censys's extensive scan of the entire IPv4 space, providing an internet-wide perspective on web entities, but

not how user traffic is distributed among those web entities. To gain a more holistic understanding of web encryption, it is important to consider web traffic. The Google Transparency Report, which tracks the prevalence of HTTPS connections to Google Chrome, reveals over 90% of web traffic is encrypted. This is a reassuring indication that top websites are widely adopting HTTPS for secure browsing.

Part Five

The State of Web Entity Security

The State of Web Entity Security

Now that we've started to unravel the nebulous concept of web entities on the Internet, let's delve into their associated security risks. In the following section, we will take a closer look at some of the most prevalent types of security incidents we observed on the Internet, including data breaches and exposed assets. It's important to note that Censys is not a vulnerability scanner. We conduct passive scanning across public internet-facing hosts and do not attempt to gain access to authenticated services. However, we can leverage our data to better understand the spread of security vulnerabilities and identify specific vulnerable devices and networks.

Data Leaks on the Web

In today's digital world data leaks and breaches are a common concern, and organizations and individuals must remain vigilant in protecting sensitive information. Generally, a data leak refers to the unintentional exposure of sensitive data, while a data breach refers to intentional unauthorized access of sensitive data for malicious purposes. For example, a data leak could occur when a cloud storage service is misconfigured and allows unauthorized access to sensitive data. Both data leaks and data breaches can have serious consequences, including financial loss, reputational damage, and regulatory penalties.

In the past decade, some of the most significant data breaches were not caused by advanced nation-state-developed zero-day attacks. Rather, many of them occurred due to ***human error***, where a mistake led to the exposure of large amounts of data on a server without any security measures in place, such as authentication, authorization, or filtering.

For example, in 2017 a significant data leak occurred when River City Media leaked 1.37 billion detailed contact records online due to a poorly configured system with no authentication.

> *"Someone had forgotten to put a password on this repository," Vickery claimed. The data was found in a backup…"*

And in 2022, confidential and personal data on 1.8 million Texas residents was available online for what turned out to be three years. The information contained real names, social security numbers, addresses, phone numbers, and birthdays for anyone who filed for workers' compensation with the Texas Department of Insurance.

According to the 2022 Verizon Data Breach Investigations Report, 41% of hacking-related breaches involve stolen credentials. This highlights the significance of using previously-compromised data to gain a foothold into an organization as the starting point for a full-scale attack. Sometimes the targets of a breach were not targets until attackers found something that made them a target.

This raises concerns about the potential for small initial mistakes to lead to significant consequences. For instance, when migrating a database to another server, an administrator might take shortcuts and dump the database to a plaintext SQL file in a directory accessible via HTTP. After the migration is complete, the files may not be properly cleaned up, or the administrator might think that using a non-standard HTTP port would hide the exposed data. Nevertheless, the data remains accessible on the internet, and with the right tools and techniques, the files could be discovered and used for malicious purposes.

Censys maintains the most complete, accurate, and up-to-date view into HTTP/S services across the internet, even those on non-standard ports that are not referenced by external websites, making it an exceptional tool for detecting information that may not be intended for public consumption. In our investigation, we discovered several instances of hosts exposing certain types of data to the public internet without safeguards against unauthorized access.

We identified 8,336 servers on the internet hosting various database dumps, backup files, passwords, excel worksheets, environment variables, and even some SSL and SSH private keys.

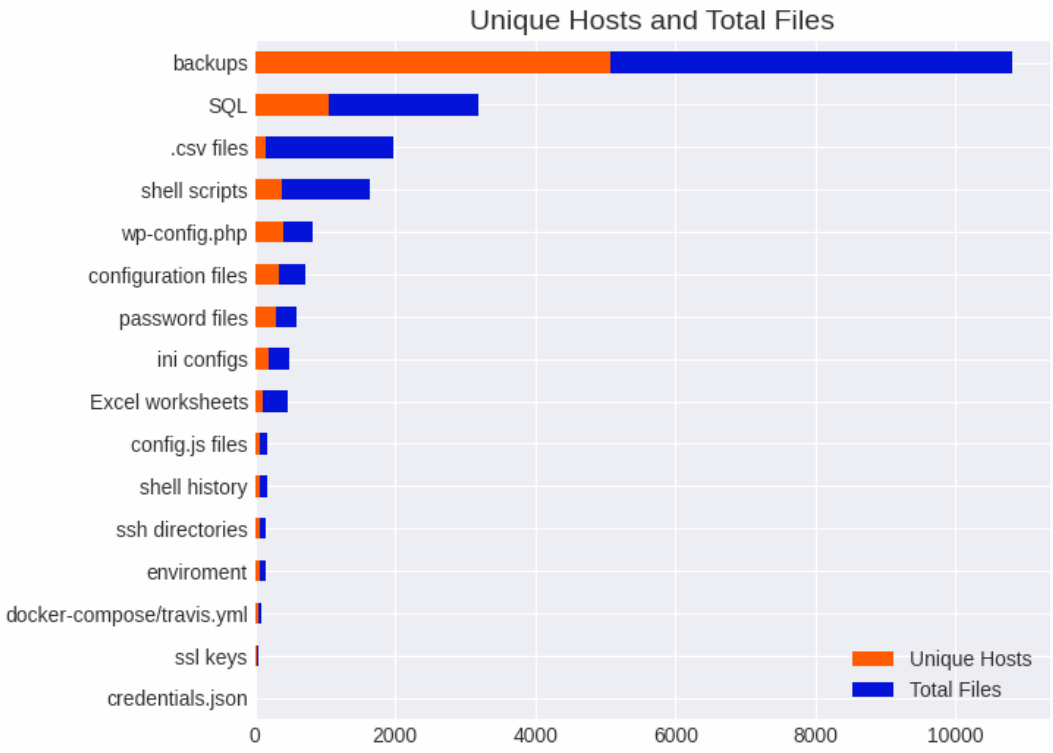


Figure 10: *Potentially sensitive file and host counts*

Over a thousand hosts had exposed over two thousand SQL database files with no authentication requirements on the HTTP services themselves. While Censys does not have insight into the contents of these files, their mere existence on a publicly accessible web server should be enough to raise alarms.

In the same vein as SQL data, CSV (Comma Separated Values) files are commonly used for storing and exchanging tabular data between different applications. Some everyday use cases for CSV files include financial and accounting data, scientific and research data, customer and contact lists, inventory and product lists, and website data such as blog posts or product descriptions. **We found over eighteen thousand CSV files publicly exposed on just 147 hosts.**

If that wasn't enough, **we found that over five thousand hosts exposed over five thousand files and directories, indicating they are related to a backup.** These backup files and directories could contain confidential, personal, or credential-related information.

We also observed over four hundred publicly accessible WordPress configuration files ("wp-config.php"). WordPress requires a database configured via the "wp-config.php" file containing all the necessary credentials. Attackers could use the credentials to access the underlying database if this file is exposed to the internet.

Furthermore, the situation could escalate if these credentials are shared with other accounts. Numerous organizations use the WordPress content management system to manage their public-facing website (over 13,000 hosts on the internet use the enterprise version).

We also observed several more risky data types exposed on these hosts. The above chart shows that backup-type files make up the majority, but these exposures also included shell scripts, histories, and various configuration file types.

Censys never attempts any website crawling or indexing; it only knows about the default content served from the base path of the web server, meaning we never traverse past the site's root directory. A threat actor, however, has no limits and will crawl and download the entirety of an HTTP property if the files look enticing.

The data above has demonstrated that **the term "vulnerable" host doesn't only refer to servers with outdated and exploitable software. Vulnerabilities can arise from various sources, including errors in judgment, misconfigurations, and rushed work.** It's important to understand that a quick and easy solution today may result in a severe data breach tomorrow.

Who Monitors the Monitors and Documents the Documenters?

When integrating a new tool into your organization, it's crucial to be aware that the default security measures may not always be sufficient. It's a mistake to assume that a tool is inherently secure right out of the box. In some cases, tool developers may place the responsibility of securely configuring software instances on the end users themselves. One clear illustration of the security challenges that can arise from this is evident in certain widely used monitoring software.

As website architectures and organizations scale up, the task of monitoring and documenting them becomes increasingly complex. To tackle this challenge, developers have created tools that leverage dynamic feeds and connectivity to growing networks and services. These tools help us track the behavior and performance of our systems and enable us to self-document the backend API's functionality. **However, this approach may create security challenges as some systems prioritize flexibility and simplicity over security.**

If the monitoring software and API endpoints are inadequately protected, they could serve as a gateway for malicious intent. If such an attacker gains access to an organization's monitoring data, they could identify other assets on the network owned by the target and craft a further plan of attack. Furthermore, if an attacker can gain direct access to an API endpoint, they could exploit this opportunity to identify potential vulnerabilities or scrape for confidential data.

It's important to note that the issues we'll discuss here aren't necessarily the fault of the tools' developers. They have deliberately chosen to streamline their development process by pushing the responsibility of ensuring general security best practices down to the software users.

In the next section we will explore these security implications, starting with the popular monitoring tool Prometheus.

PROMETHEUS

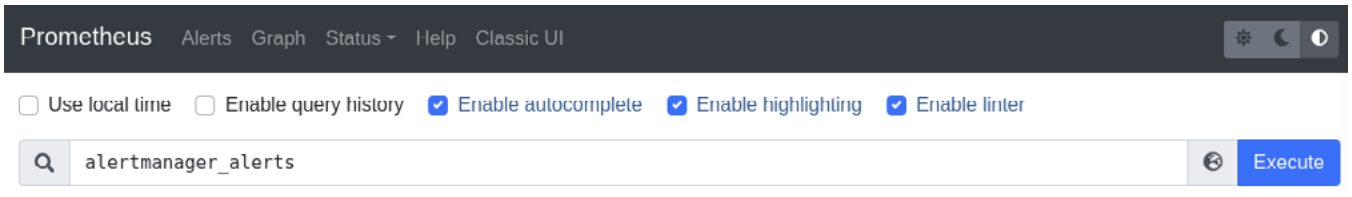


Figure 11: *Screenshot of Prometheus navigation bar*

The more the web evolves into decoupled micro-services – scaled up and down as dynamically as traffic increases or decreases – the more flexibility our monitoring tools need to have in finding systems to monitor.

Prometheus fits this new model for the web with its ability to auto-discover and monitor an organization's assets running in dynamic environments like Kubernetes or AWS with zero configuration.

Prometheus pulls data from an HTTP endpoint in a Prometheus-specific format describing different metric types, such as counters, gauges, and histograms. There are two primary ways Prometheus finds endpoints to monitor: either statically configured or auto-discovered using various cloud connectors and/or DNS Service Discovery (DNS-SD).

One issue arises from Prometheus's security model, which presumes that all untrusted users have access to the HTTP endpoints and the data contained within, including the entirety of the database. **This means that by default, anyone in the world can view the activity in a Prometheus installation – including potential attackers.**

We can see the Prometheus servers themselves, and an attacker can also gain deep insights into the systems they monitor. For example, over 41,800 Prometheus servers are exposed to the internet and are monitoring over 219,400 endpoints. Amazon alone has 6,981 Prometheus servers monitoring 35,175 systems.

Below are the Prometheus servers exposed to the internet broken down by the autonomous system. Several cloud providers are in the top ten: Amazon, Alibaba (which provides cloud services), Digital Ocean, OVH, and Google Cloud Platform.

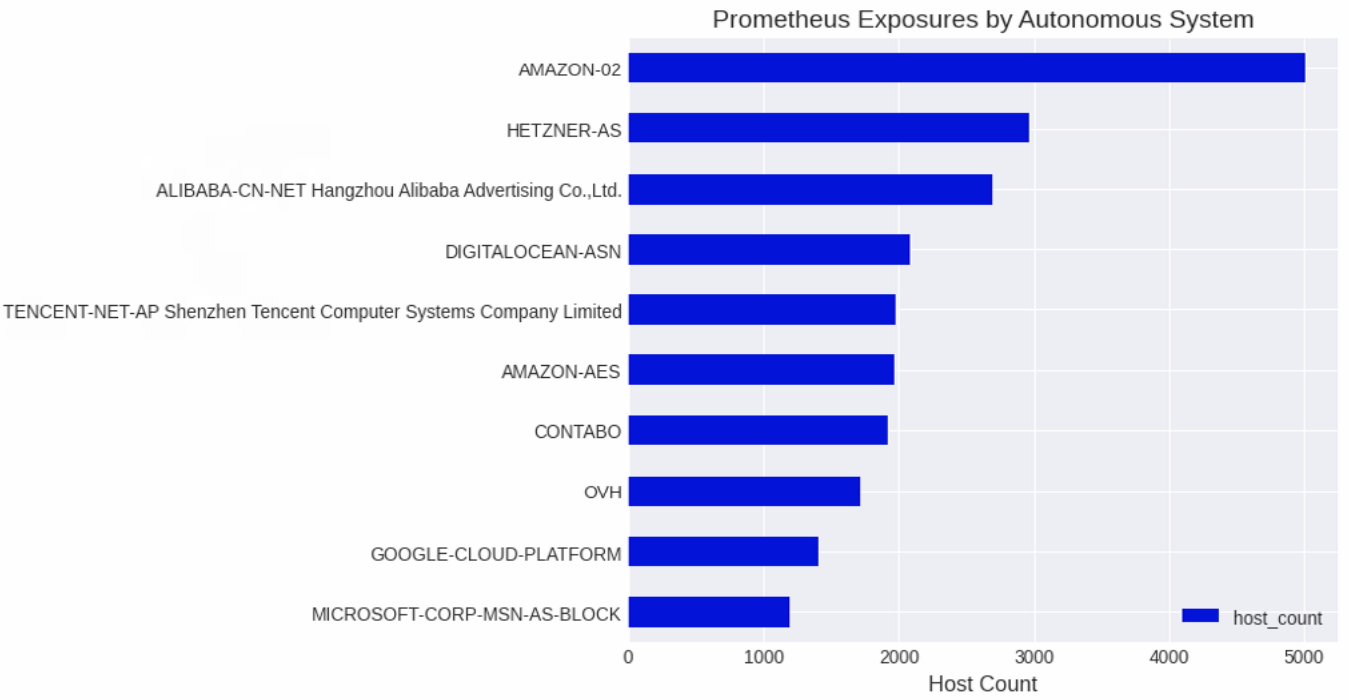


Figure 12: *Top autonomous systems where we observe Prometheus exposures*

We can break this down even further to show which hosts each Prometheus server monitors and whether those hosts exist in private (RFC1918) IP networks or private DNS namespaces. In other words, how many monitored endpoints can attackers gain insight into which are living in an organization's private networks?

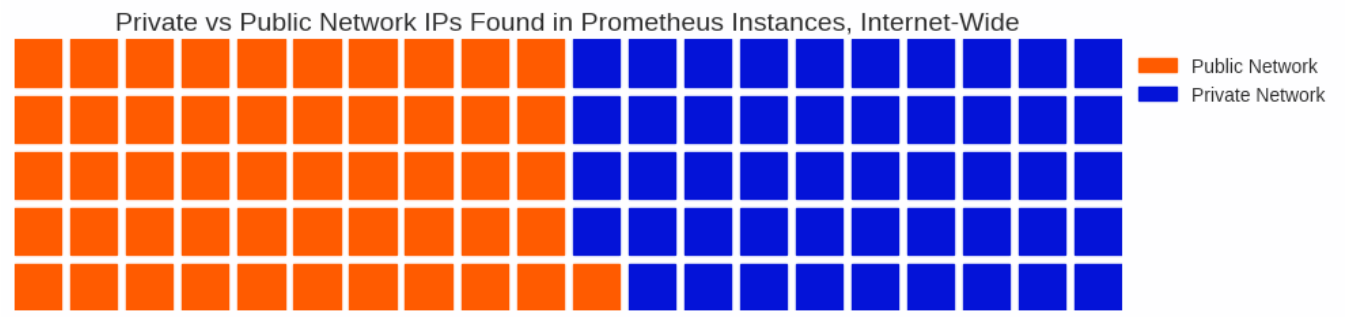


Figure 13: *Percentage of private and public network IP exposure observed in internet-exposed Prometheus instances*

Over 48% of the total monitored endpoints with active metrics exist exclusively in private IP and DNS space, which would typically not be visible to global internet users.

Figure 14: *Percentage of private and public network IP exposure observed in internet-exposed Prometheus instances on Amazon networks*

The Prometheus servers located only in Amazon networks contain internet-accessible metrics on over 18,600 private (53%) and 16,600 (47%) public networks.

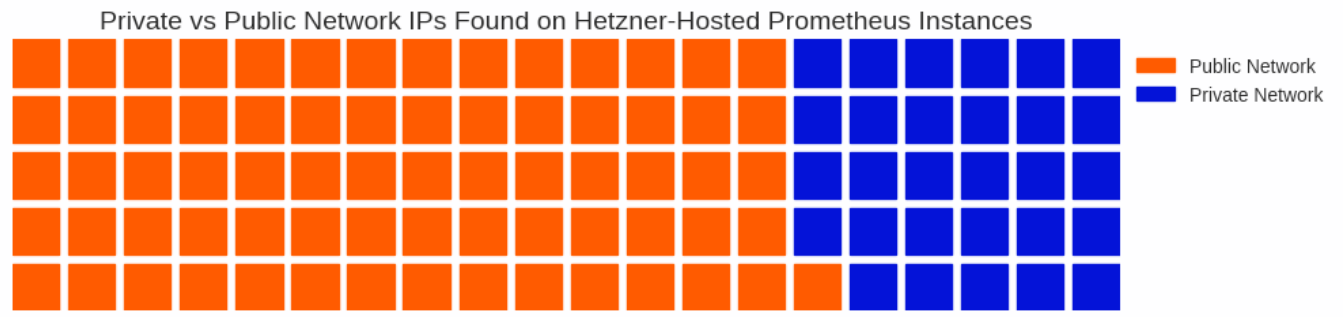


Figure 15: *Percentage of private and public network IP exposure observed in internet-exposed Prometheus instances on Hetzner networks*

On the other hand, Hetzner networks, which provides all types of internet hosting (Managed, Private, Virtual, Dedicated, and some cloud services), have Prometheus servers that primarily monitor public IP and DNS zones. Only 5,392 Prometheus-monitored targets are on a private network, while over 13,300 monitoring targets are publicly available.

Monitoring systems like these not only allow attackers to perform reconnaissance, but they allow them to create detailed schematics of the running cogs of a network, both public and private. Integrating a new tool into an organization requires careful consideration of the default security measures. If these systems don't give the security we need, it's left up to the individual to ensure that only the trusted users can look around.

Next, let's examine another prevalent instance of incomplete security practices often observed in web entities: unprotected web-based APIs and API endpoints.

WEB APIS & SWAGGERUI

In 2018, India's national ID database, a system with fingerprint data, iris scans, and other biometric information that 1.1 billion citizens of India use to register for various government services, was found to be fraught with security issues. In March of that year, news hit that this system had been leaking private data via an unsecured API endpoint without limits on the number of queries a client could make. Attackers then used this endpoint to iterate over every possible ID permutation to pull down citizens' private information.

With the increasing availability of data on the web, protecting the most sensitive components of our web applications has become paramount. Since the 2018 Indian national ID database leak, sufficient measures have not been taken to prevent similar breaches. As we continue to strive toward automation, we must remain vigilant in our security practices to protect those automated processes.

Web APIs and API endpoints serve as gateways for frontend applications to access and manipulate potentially sensitive data, providing a layer of abstraction to backend systems. However, without proper security measures, these APIs can be vulnerable to exploitation by threat actors, leading to unauthorized access to the underlying data, theft of sensitive information, or even complete system takeover.

The first step for exploiting an API is to identify its existence. Once the API has been pinpointed, the attacker must understand how that API functions. Unfortunately, if this information is readily available to the attacker, it significantly simplifies their job.

Let's explore a widely-used system that has become an industry standard for the creation, maintenance, documentation, testing, and development of web-based APIs, and is known for its user-friendly interface. A system that does not come out of the box with any security controls at all.

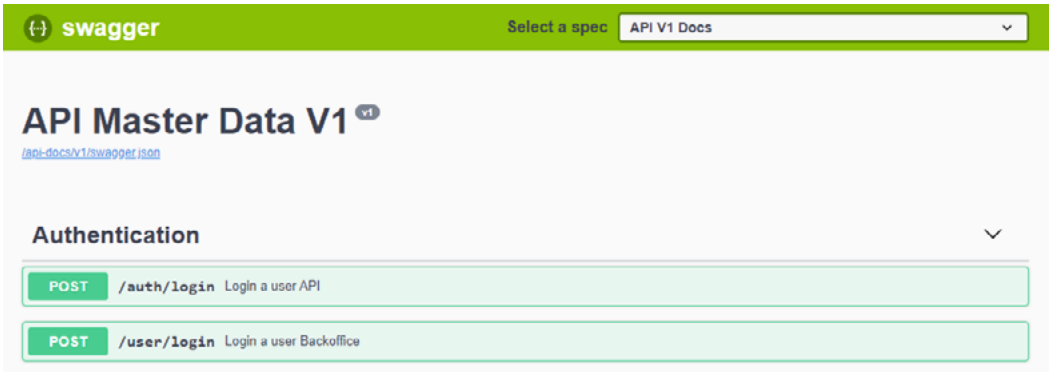


Figure 16: *Screenshot of SwaggerUI docs with an example "Authentication" API request*

SwaggerUI is an open-source tool using the OpenAPI standard that allows developers to generate interactive documentation for their web APIs. It provides a graphical interface that enables developers to visualize and interact with the API endpoints, explore their functionality, and test specific API calls directly from the dashboard.

There are legitimate use cases for SwaggerUI being made publicly available; for example, many companies have a public API for their products, and Swagger is an exceptional way to document it.

However, if unintentionally exposed to the internet due to misconfigurations, human error, or other oversights, SwaggerUI can reveal sensitive information about a private API, such as endpoints, request/response schemas, and authentication methods. Malicious actors can abuse these API endpoints to misuse and discover vulnerabilities, such as data scraping or injection attacks.

In total, Censys has observed over 42,800 hosts running the SwaggerUI dashboard. Over 46% (16,923 hosts) of the exposed SwaggerUI endpoints can be found in the United States.

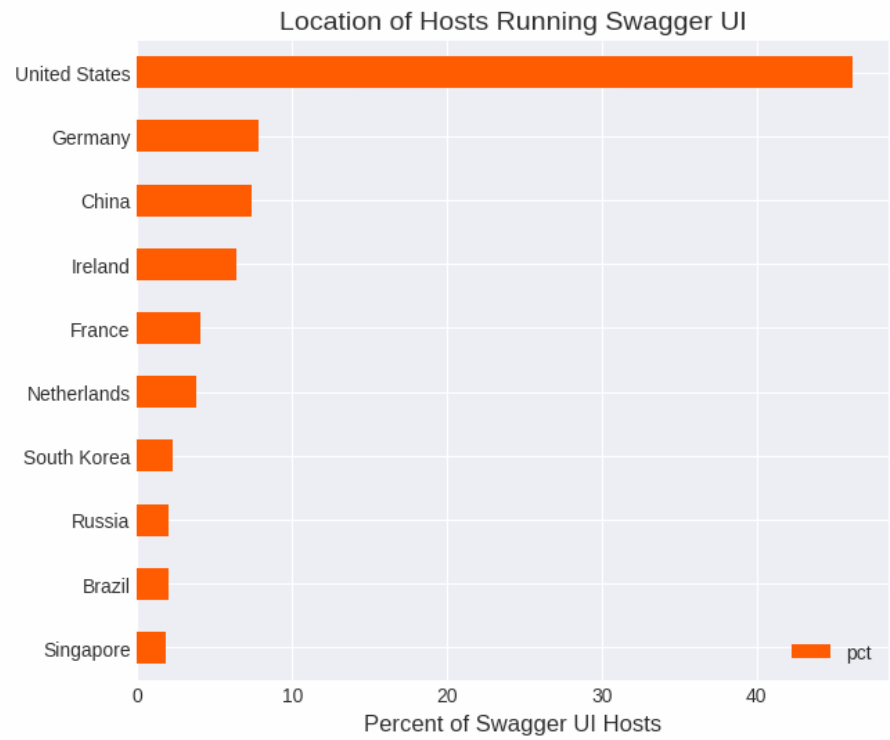


Figure 17: *Percent of SwaggerUI instances among the top countries where hosts running SwaggerUI were observed*

By default, SwaggerUI will run on HTTP port 8080. If an administrator exposes or moves this to a standard HTTP port (80 or 443), we can assume that the administrator intended this service to be public (although we can not determine intent). If we see that the service is kept on port 8080, this may or may not indicate that the service was intentionally public. On the other hand, if the service is moved from port 8080 to another non-standard HTTP port, we may consider this an unintentional exposure.

Let's look at how things change when we break out SwaggerUI exposures by standard and non-standard HTTP ports.

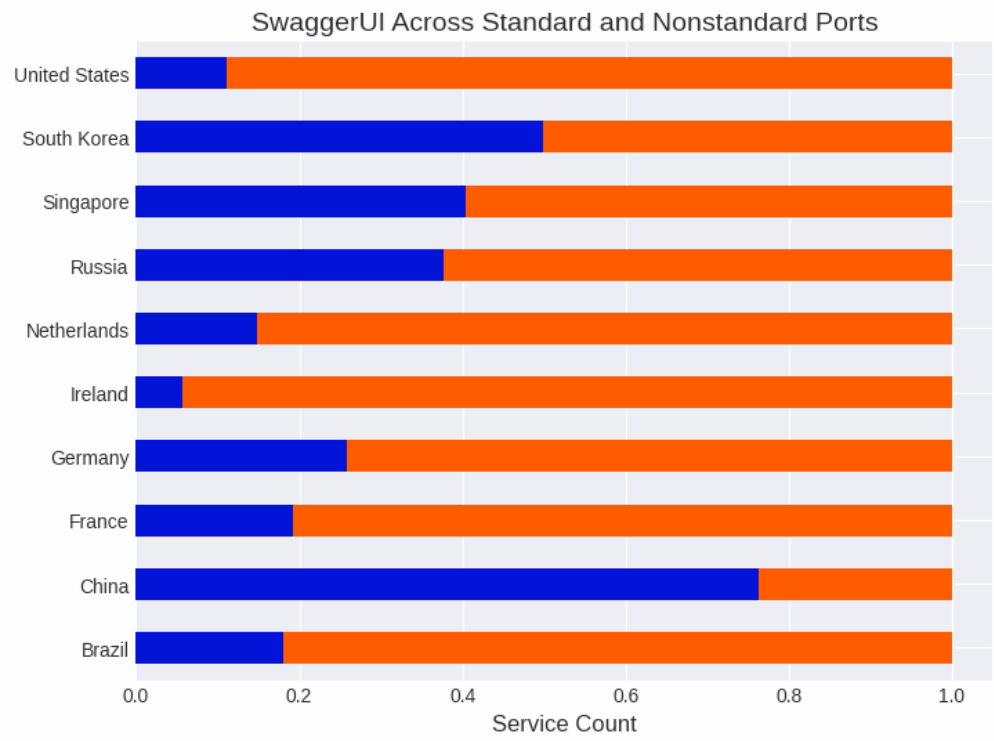


Figure 18: *SwaggerUI instances on standard (orange) and non-standard (blue) ports, with counts normalized from 0-1*

We see here that the view changes drastically. While the United States still has the most number of SwaggerUI services, the majority run on standard-HTTP ports, and China is the reverse; almost all of their exposed SwaggerUI services are running on non-standard HTTP ports. Over 35,000 hosts are running SwaggerUI on a standard HTTP port, while over 8,000 hosts are running on a non-standard HTTP port.

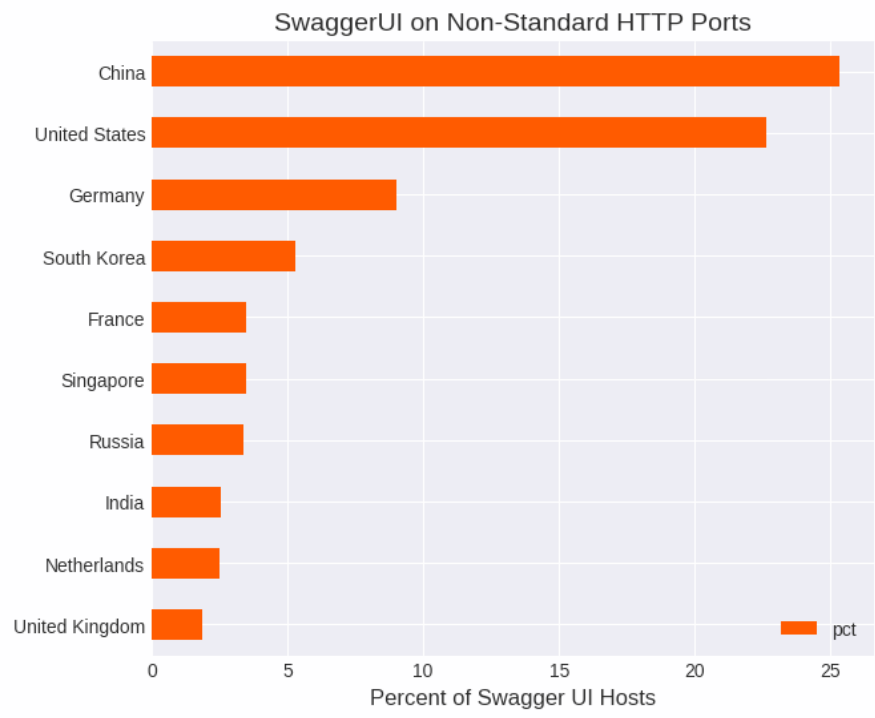


Figure 19: *Percent of hosts running SwaggerUI on HTTP ports other than 80 and 443 by country*

If we further reduce this by excluding the default SwaggerUI port 8080, we see the United States decrease in exposures by 4% and China increase by 3%.

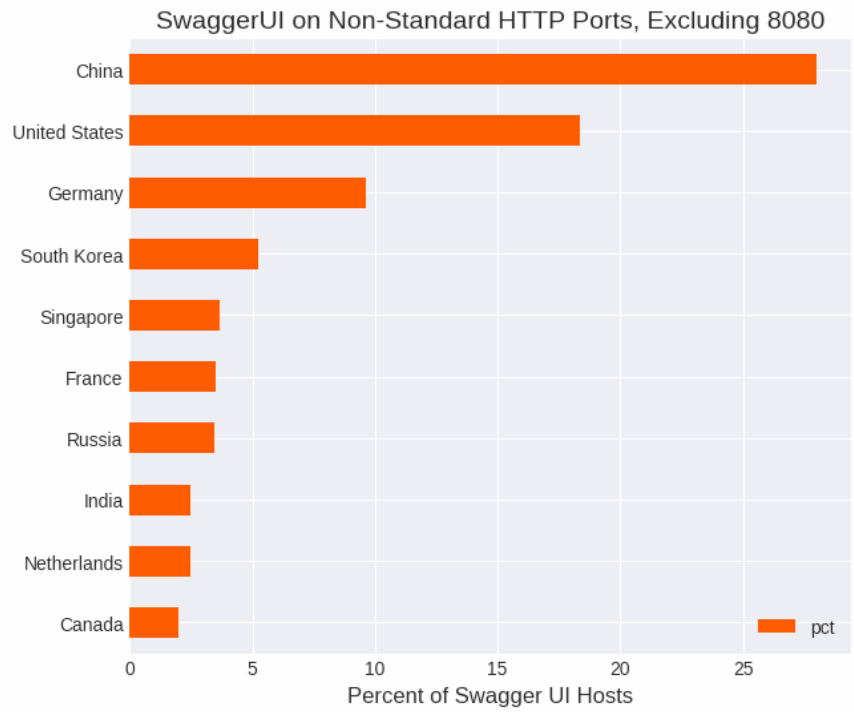
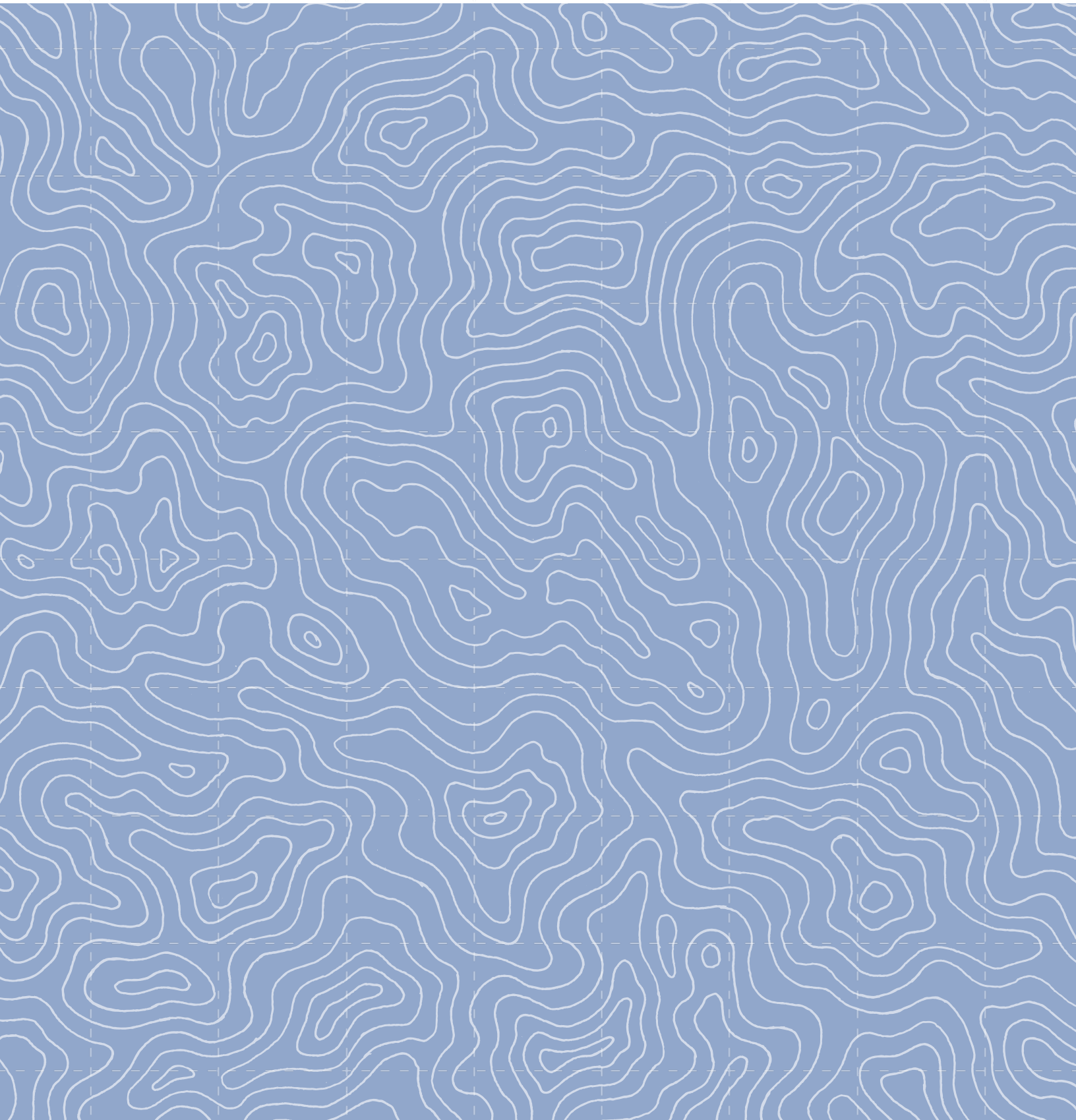


Figure 20: *Percent of hosts running SwaggerUI on non-standard HTTP ports by country, excluding default SwaggerUI port 8080*

In this section, we have only begun to explore the security issues frequently observed in web entities on the internet. Although misconfigurations, default security settings, and exposures may seem like minor oversights in the short-term, they can serve as a foothold for threat actors to gain unauthorized access to an organization's network, potentially resulting in more severe security incidents with long-term consequences. It is crucial to take a proactive approach in implementing robust security measures in order to shrink your organization's attack surface and enhance the overall security posture of your web entities on the internet.

Part Six

Conclusion



Conclusion

The internet, with all its complexities, has become an integral part of our lives and businesses. It is a vast and dynamic infrastructure, and in this report, we have only scratched the surface of what it encompasses. While web servers are the most common technology that we see running on HTTP, many other technologies operate on web entities, such as load balancers, APIs, and proxies. Beneath the surface, there are countless layers of software that power the web.

There are indicators that the state of internet security is moving in a positive direction, with an increasing percentage of HTTP services encrypted with higher versions of TLS and increased TLS certificate adoption. However, opportunities for threat actors to disrupt the security of our online presence remain, and organizations must be vigilant in their efforts to address these threats. Misconfigurations, outdated and vulnerable software, and improperly exposed API endpoints are just some of the weaknesses threat actors can leverage to exploit organizations' online systems.

However, the good news is that exposure to these types of threats can be significantly reduced by adopting proactive security strategies. The often unglamorous work of asset, vulnerability, and patch management is critical for helping reduce an organization's attack surface. The security issues we've explored in this report aren't a result of zero days or other advanced exploits, but rather misconfiguration and exposure issues that are likely a result of simple mistakes or configuration errors.

We hope that by shining a light on these issues, we can help emphasize the importance of strong foundational security practices, allowing us all to work together toward a safer internet.

Appendix

All data unless otherwise specified is pulled from our Universal Internet Dataset (UIDS) and Certificates 2.0 dataset. Daily snapshots of host data are from February 28, 2023, and certificate data is from March 27, 2023. Year-over-year comparisons of host data were made between aggregated daily snapshots over February 2022 and February 2023.

Our host and certificate data is freely available to search at https://search.censys.io.

Universal Internet Dataset (UIDS): Our internet-wide scan dataset. It is derived from continuous scanning of the entire IPv4 space on over 3,592 ports, as well as:

- **More Frequent Global Scan of Popular Ports**. We scan the whole IPv4 space on 137 ports with IANA-assigned services every day.
- **Cloud Provider Scans**. Since many cloud hosts are ephemeral, we scan the 1,440 most popular ports on Amazon, Google, and Azure hosts every day.
- **Global Scan of Less Popular Ports**. We scan the whole IPv4 space on 3,455 additional ports on a regular basis, completing a walk every 10 days.
- **Global Scan of Every Other Port Number**. We scan the entire IPv4 address space across ALL 65535 ports at a low background rate.

Certificates 2.0 Dataset: the largest X.509 certificate repository in existence. Certificates are collected via TLS handshakes during Censys scans of the public internet and via syncing with a number of CT (Certificate Transparency) logs, including:

- **Daily revocation checks & processing of certificates**. Censys collects certificate trust information and regularly validates unexpired certificates.
- **Fast integration with new CT logs**.
- **Deduplication of pre-certificates**.