BinaryEdge.io
Be Ready. Be Safe. Be Secure.

INTERNET SECURITY EXPOSURE 2016

Acknowledgments

BinaryEdge would like to thank the following security experts for advisory and review of the report:

- Bruno Morisson - @morisson
- Chris John Riley - @chrisjohnriley
- João Poupino - @poupas

Table of Contents

Executive Summary

Everyday, more and more companies expose their servers and services to the internet, increasing the attack surface that can be used by hackers. For this report, we wanted to understand the current state of exposure of the internet connected devices. This will allow us to create a baseline for future reports and follow ups. We identified the most used ports, scanned the ports, identified the services running on those ports and extracted specific security metadata. This allowed us to measure the state of security of different services.

For this analysis, we used BinaryEdge's 40fy platform, which is used daily by our clients to monitor their perimeters, obtain security research data and understand their exposure to attacks.

Our key findings are as follows:

- Security exposure isn't just affecting one specific industry. When creating this report we managed to associate the IP address ownership to companies in different industries such as: **Pharmaceutical, Energy, Information Technology, Robotics, Banks, Media, E-Commerce**, among others.
- The **exposure of companies doesn't have a direct correlation to revenue**, we found data about companies ranging from small startups all the way up to Fortune 500 companies.
- One positive aspect is that **SSH has overtaken Telnet** as the most used service for remote administration on terminal based services. This is a good change, as Telnet is an extremely outdated protocol that was developed without making security a priority.
- An extremely high number of IP addresses are exposing **outdated and vulnerable versions of software**, making them prone to attacks and therefore exposing the organisations.
- **Multiple Terabytes of data are exposed** due to misconfigured databases that have no authentication.
- Critical systems such as electricity grids, PLC controllers, **water tanks and flow control systems, medical devices, petrol stations**, are exposed directly to the internet with no authentication via protocols such as X11 and VNC, therefore granting direct control to their attackers.
- **Smart houses, prisons, hospitals, radiation meters** and others were also found exposed via the MQTT protocol, again with no authentication or any type of security.

Considering everything, there is still a lot of work to be done by organizations to fix the services they expose to the internet. Updating and patching are two important processes that can easily fix a lot of issues currently existing in many organizations. It's crucial that these organisations monitor and control the level of exposition of their services.

A high number of mobiles are also exposed with no authentication using the X11 service. We found this case to be more prominent in India in a specific provider.

IoT and IPv6 are bringing more functionality and power to companies that develop products, which also creates new privacy and security issues. If in the current IPv4 space with 4 billion IP addresses we're already seeing bad results, one can imagine what will happen when IPv6 is widely used and many more devices, services and systems are directly connected to the internet.

Concepts and Terminology

- **Internet Protocol Address:** or IP Address for short, is a set of numbers that is assigned to each device connected to a network that uses Internet Protocol for communication. In very layman terms it's your address on the internet or on the network you're connected to. An IP address usually acts as an identifier of either a client or server.

- **IPv4, IPv6:** An IP address can usually be found in one of two formats: IPv4 or IPv6. Due to the limit of IP addresses that we can have from IPv4 (4 Billion) an exhaustion is happening on that IP address space and IPv6 was created which can have up to 2^{128} (approximately 3.403×10^{38}) IP addresses. An IP address v4 has 4 octets of decimal integer numbers from 0 to 255 and looks like: **10.10.10.1**. An IP address v6 has 8 octets of hexadecimal integers from 0000 to ffff and looks like: **2001:cdba:0000:0000:0000:0000:3257:9652.**

- **Service:** 65,535 Ports are available at each IP Address to run services on. These services are essentially applications that are listening and serving some content that can range from Web servers, FTP servers, Email servers to IoT devices with their web management pages.

- **Probes:** Probes are sets of network packets that BinaryEdge uses to communicate with the different services. We use these to identify which ports are open and which services are running on them.

- **Portscanning:** Portscanning can be defined as the act of asking a port in an IP address if it is open and then following by sending a probe that asks which service is running on it.

- **Vulnerability, exploit and bug:** A software can have bugs. Bugs are flaws in the code that might result in undesired behavior. These bugs can be something simple that just makes a program output a wrong value or they can be a lot more complex including a security vulnerability. Exploits are pieces of code that abuse those security vulnerabilities to make the program do other things that it was not meant to.

- **Blacklist:** List of IP addresses that we do not scan. At BinaryEdge we scan the world regularly, we do not attempt to do any logins and only connect to services that are publicly exposed (no authentication). However there are companies that do not want to be scanned and they communicate this to us. Although this might seem like a good idea, its merely creating a false sense of security as there are plenty of hackers and other entities that will portscan and not respect any blacklist.

- **Encrypted vs Cleartext:** During transmission, data can be in one of two states: either encrypted or in cleartext. Encrypted transmissions are the preferred choice by security as usually encrypted communications can't be deciphered by threat actors that were not meant to view that data in case they have a way to intercept it.

Introduction

For over a year, BinaryEdge has been scanning the IPv4 space of the internet daily. Using the gathered data, we have written two blogposts about the state of security in two different countries:

- **Switzerland**
  https://blog.binaryedge.io/2016/02/16/security-of-a-state-switzerland

- **Portugal**
  https://blog.binaryedge.io/2016/03/31/security-of-a-country-portugal

This time, we decided that a report about the entire world had to be written. Looking at countries individually is great, as microscopic view, but we started to wonder about the results we could find if we took a macroscopic view at all the data we collect.

Being a data company and since we scan the world daily, all our focus goes towards delivering the highest quality of data to our clients, in the simplest and cleanest of ways. This is the reason why in our company DNA, we not only have cybersecurity specialists, but also have an entire team dedicated just to data science.

Using our platform 40fy (https://www.40fy.io/), we acquired all the data necessary to produce this study.

On this report, we will not only look at a different set of ports and services running on those ports, but we will also explore the data acquired from interacting with those services. Some examples of this would be our analysis of all the SSH keys found on port 22 or the SSL certificates on SSL-enabled services.

In order to enrich the data presented, in this report we will also look at CVE's and vulnerabilities. For this, we created a scoring mechanism based on the CVE's that affect different services.

Further work and expansion of this report will be done on our blog: **https://blog.binaryedge.io**.

For information about our data and products, please contact us on: info@binaryedge.io

Who is BinaryEdge?

BinaryEdge (https://binaryedge.io) is a company located in Zürich, Switzerland with a multifunctional team that focuses on acquiring, analyzing and classifying internet wide data by combining efforts in the areas of Cybersecurity, Data Science and Machine Learning.

By combining our engineering knowledge with Data Science methodology, algorithms and tools, we are able to acquire, correlate and transform data to find outputs, patterns and outliers that help people understand their cybersecurity exposure.

As a company, we have created our platform called 40fy (https://40fy.io) and launched a mobile application called Cyberfables (https://cyberfables.io), available for iOS and tvOS.

Besides working on the development of our products, we write a blog (https://blog.binaryedge.io) where we talk about studies we have done focused on security of countries, technologies and services.

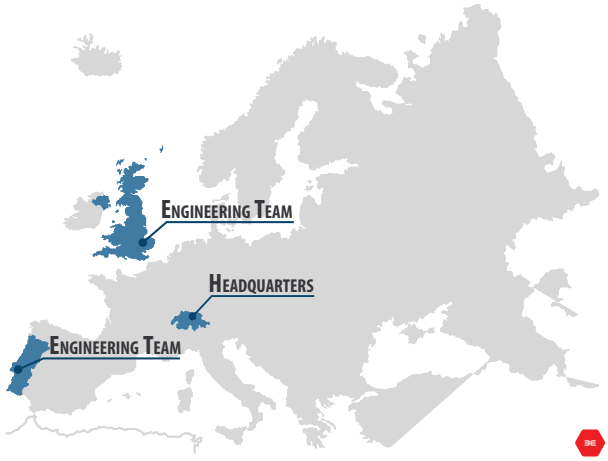


Figure 1: BinaryEdge's presence in Europe

If you're looking for data about the entire internet space or if interested in having real time perimeter monitoring, please contact us. BinaryEdge is always looking for interesting partnerships and people to join its ranks.

Team

Meet the team working on this report :

- **Tiago Henriques -** CEO
  Twitter: @balgan
  Linkedin: www.linkedin.com/in/balgan
- **Marco Silva -** Technology and Infrastructure
  Twitter: @igama
  Linkedin: www.linkedin.com/in/marcodasilva
- **Filipa Rodrigues -** Data Scientist - Machine learning
  Twitter: @filipacsr
  Linkedin: www.linkedin.com/in/filipacrodrigues
- **Florentino Bexiga -** Data Scientist - Data engineering
  Twitter: @fbexiga
  Linkedin: www.linkedin.com/in/fbexiga
- **Tiago Martins -** CTO
  Twitter: @gank_101
  Linkedin: www.linkedin.com/in/tiagojcmartins
- **João Veiga -** Software Engineer
  Twitter: @jcsvveiga
  Linkedin: www.linkedin.com/in/jvveiga
- **Ana Barbosa -** Data Scientist - Data visualization
  Twitter: @ana_barbosa90
  Linkedin: www.linkedin.com/in/anacbarbosa90
- **Inês Barros -** Designer
  Twitter: @IBillustration
  Linkedin: www.linkedin.com/in/inescbarros

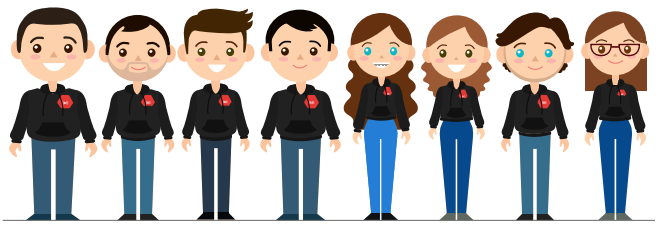


Figure 2: BinaryEdge's team

What is the 40fy platform?

The 40fy platform is a product developed by BinaryEdge that allows you **to get access to real time data about the entire internet.**

We scan the entire internet space and acquire data about all types of internet services and protocols. This data can show the exposure of the customer's perimeter to the world, statistical data about internet security or even assist him in security research.

40fy delivers data in multiple ways, according to each client's preference:

- Real-time firehose containing scanning, security, torrents and data of the entire internet
- Custom made stream that delivers the data requested via our on-demand API
- Access a repository with a complete history of the client requested data
- Access our web portal to see the history of all IP addresses

This platform is composed by different modules that return a great amount of information about the services running on the internet.

Although all these modules have been developed by BinaryEdge and are under constant improvement, we also give our customers the possibility to submit their own custom modules to 40fy, for their own use.

Customers can also request scans via our API in order to create their own personalized threat intelligence streams. By doing this, they can decide whether to have a microscopic view (by looking at their specific IP addresses or their organizations') or a macroscopic view (where they look at data belonging to an entire country or even the world).

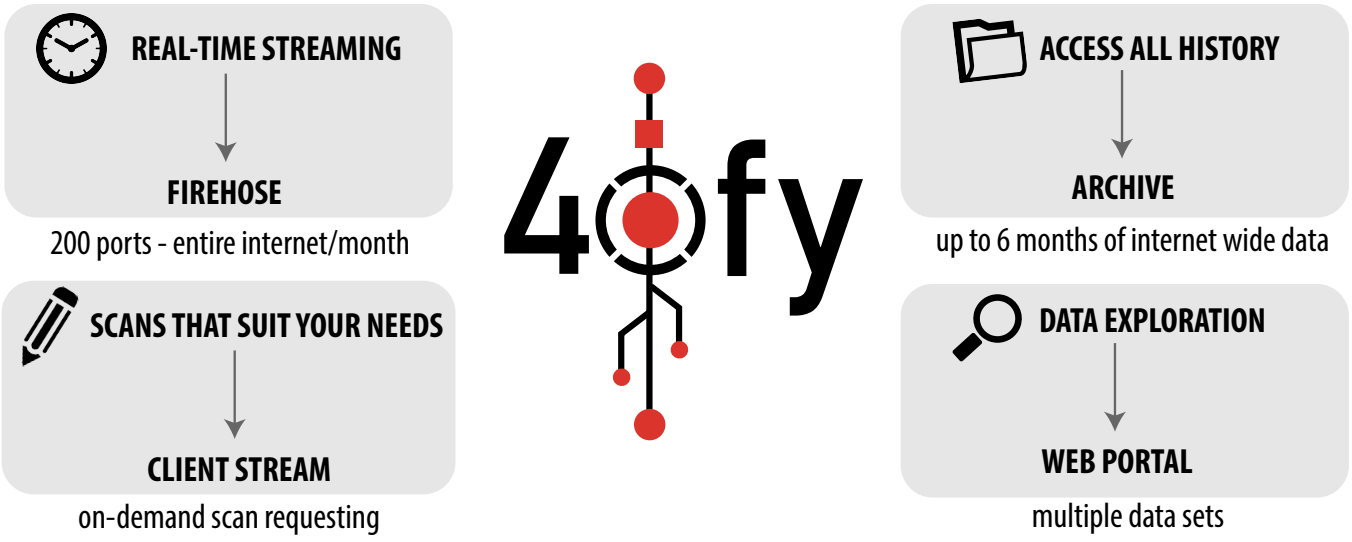


Figure 3: Platform 40fy's main features

Scanning the Internet - Why?

A question we, BinaryEdge, get asked often is **"why would anyone want to ever scan the entire internet?".**

The internet is composed by **4 billion IP addresses**, in the IPv4 space. This is an extremely large number, and, if you take into account the use of Network Address Translation (NAT) that number goes up a lot.

Network Address Translation is the process where a public IP address is assigned to a computer (or more) inside a private network. The main purpose of NAT is to limit the number of public IP addresses an organization can use, not only for security purposes but also for economic reasons.

IP Addresses are assigned to different entities (RIRs - Regional Internet Registry) which are geodistributed around the world. RIRs are organizations that are responsible for the allocation and registration of internet number resources (IP addresses and Autonomous System numbers) for the different regions of the world (as seen on the world map on the right).

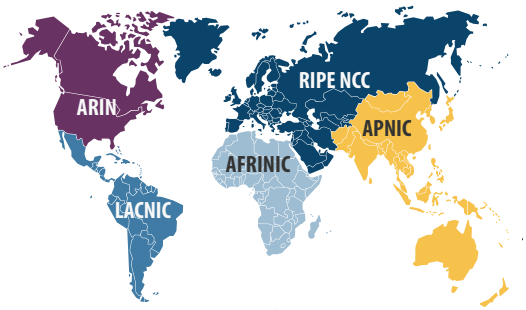


Figure 4: Regional Internet Registry

In the map below, one can see the 10 countries that have most publicly addressable IPs.

Figure 5: Top10 countries with most public IP addresses

In addition, each IP address can have **65,535** open ports with different services running on them.

Taking these numbers into perspective, when scanning the internet, allows us to answer some questions:

- **A new vulnerability is released, who is exposed?**
- **How many computers on the internet are running that specific service?**
- **Is my perimeter exposed to any of these vulnerabilities?**
- **How fast are people patching and updating their services?**

Some of these questions are interesting from a security research perspective, others from an organizational perspective and sometimes even from a user perspective.

For example, we certainly would like to know if the **power station** near our office is exposed to hackers, who could attack it and cause some damage.

...or if our **bank** has not updated its OpenSSL configuration to defend against the latest vulnerability.

...or maybe if a **baby camera** is connected to the internet and exposed for anyone to see.

Methodology

The internet

The internet consists of a network of computers connected amongst them, sharing information, providing services, and being used for many different purposes.

In this report, BinaryEdge analysed multiple services across the entire internet space. We focused on targetting some of the most known and used technologies and services, however other studies will be published in the future, either on our blog or as independent reports.

The image below shows the ports analysed in this report, according to the section of this report they're included in.

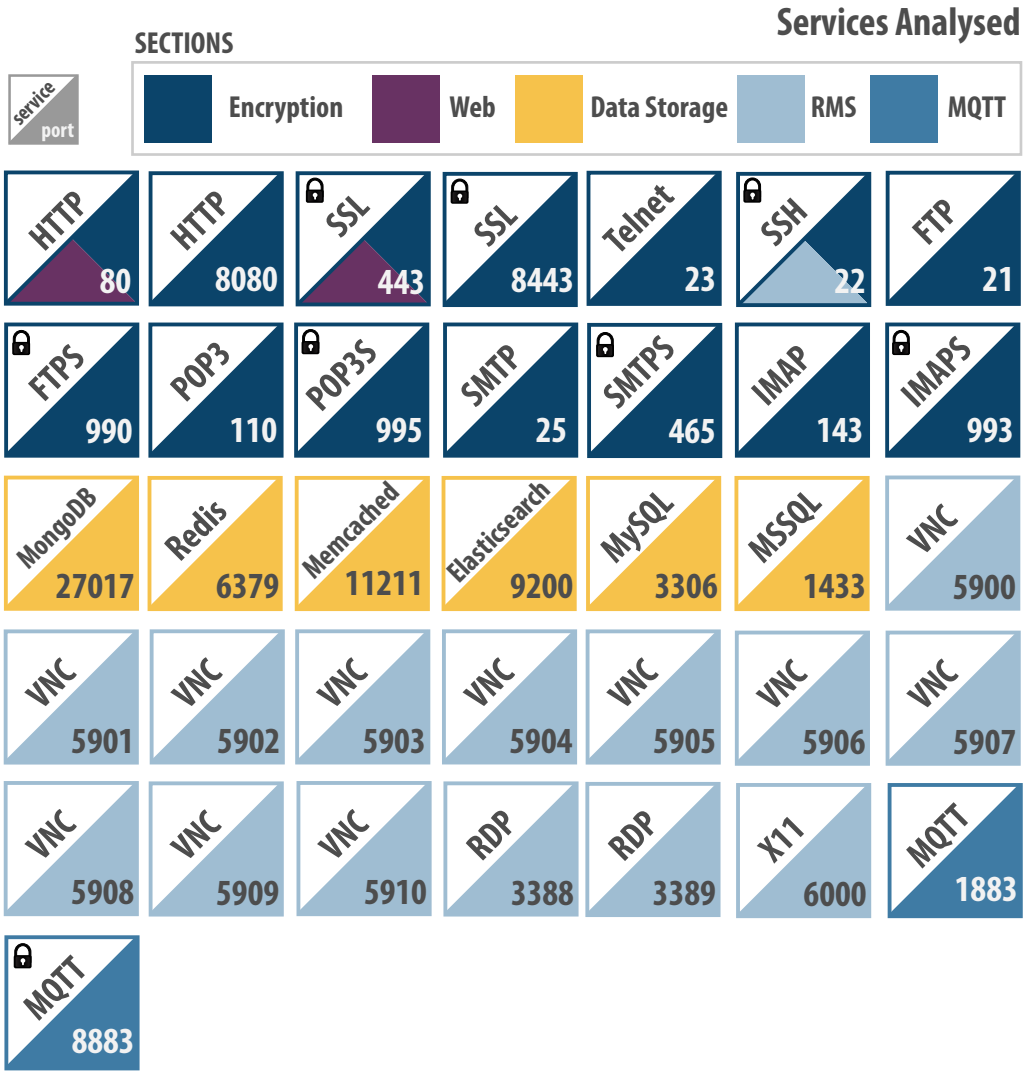


Figure 6: Services analysed in each section of this report

In our table of services, we added a lock symbol to the protocols that use some type of encryption. On this report we analysed a total of **36 ports.**

On those 36 ports, we found **114,030,477 unique IP addresses**. One interesting observation that we found was that **61%** of the IP addresses only exposed services on 1 port, followed by **22%** exposing 2 ports and in third place 3 ports (6%). We found **19,139 IP addresses exposing services on all 36 ports** - this is not uncommon since you have IP addresses that have firewalls and load balancers that respond on all ports.

The following map represents the geo-distribution of nodes that we found. Having the geolocation for all the IP addresses scanned, we were able to plot the IP addresses for which we found services running on a map, allowing us to comprehend how they are distributed.

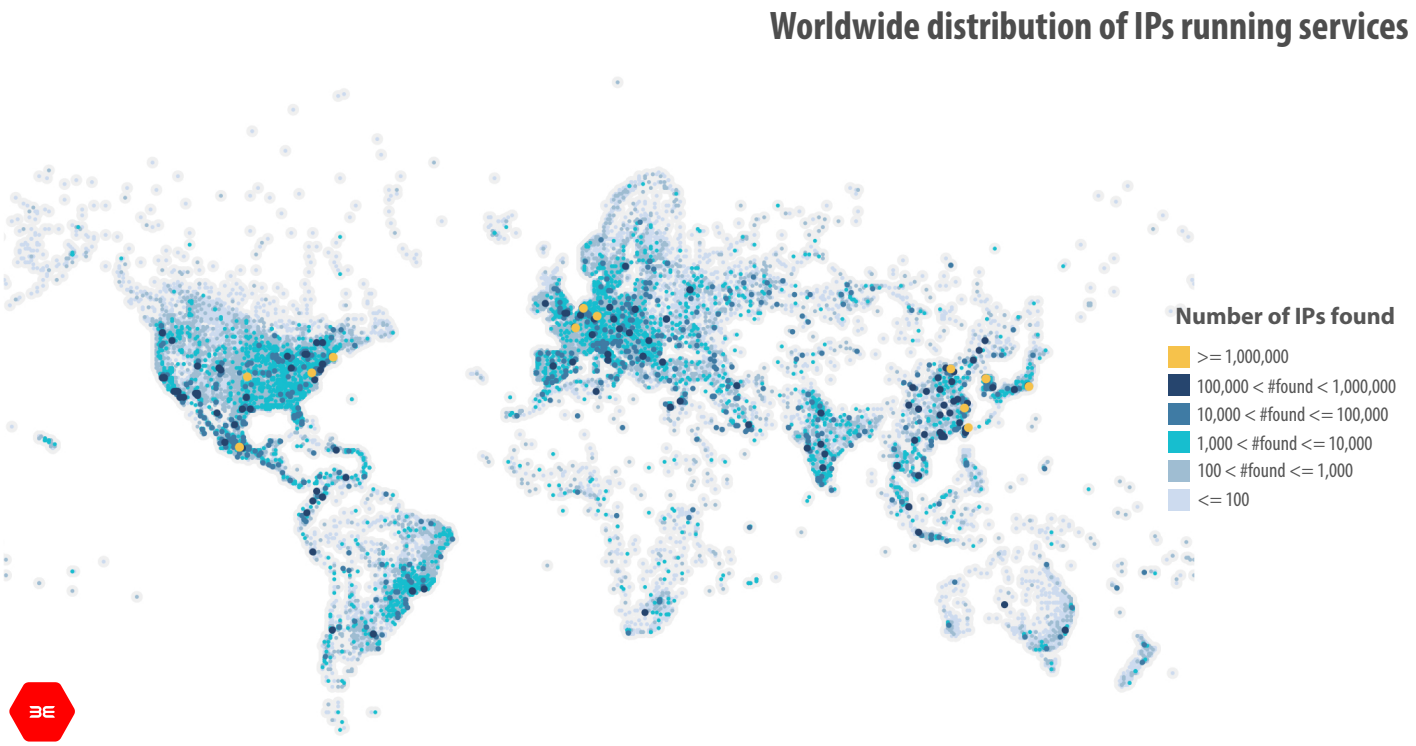


Figure 7: Worldwide distribution of IP addresses running services

The geo-distribution follows, to a certain extent, the map in figure 5, "countries with most assigned IP addresses". It's important to understand that some countries might have ranges that present issues to scanning. Many times we see that we lose visibility into Africa if we don't scan from certain geolocations or visibility into Russia if we try to scan from the USA, for example.

To give an idea of how vast the internet is, even by exploring all the ports presented in this study, we only touched a part of the internet.

The image below is a Hilbert Map that represents a view of the "internet" split by different "/24" networks. Each "/24" (which consists of 254 usable addresses per network) is represented in a pixel. The Hilbert Map includes annotations of which Regional Internet Registry (RIR) those networks are allocated to.

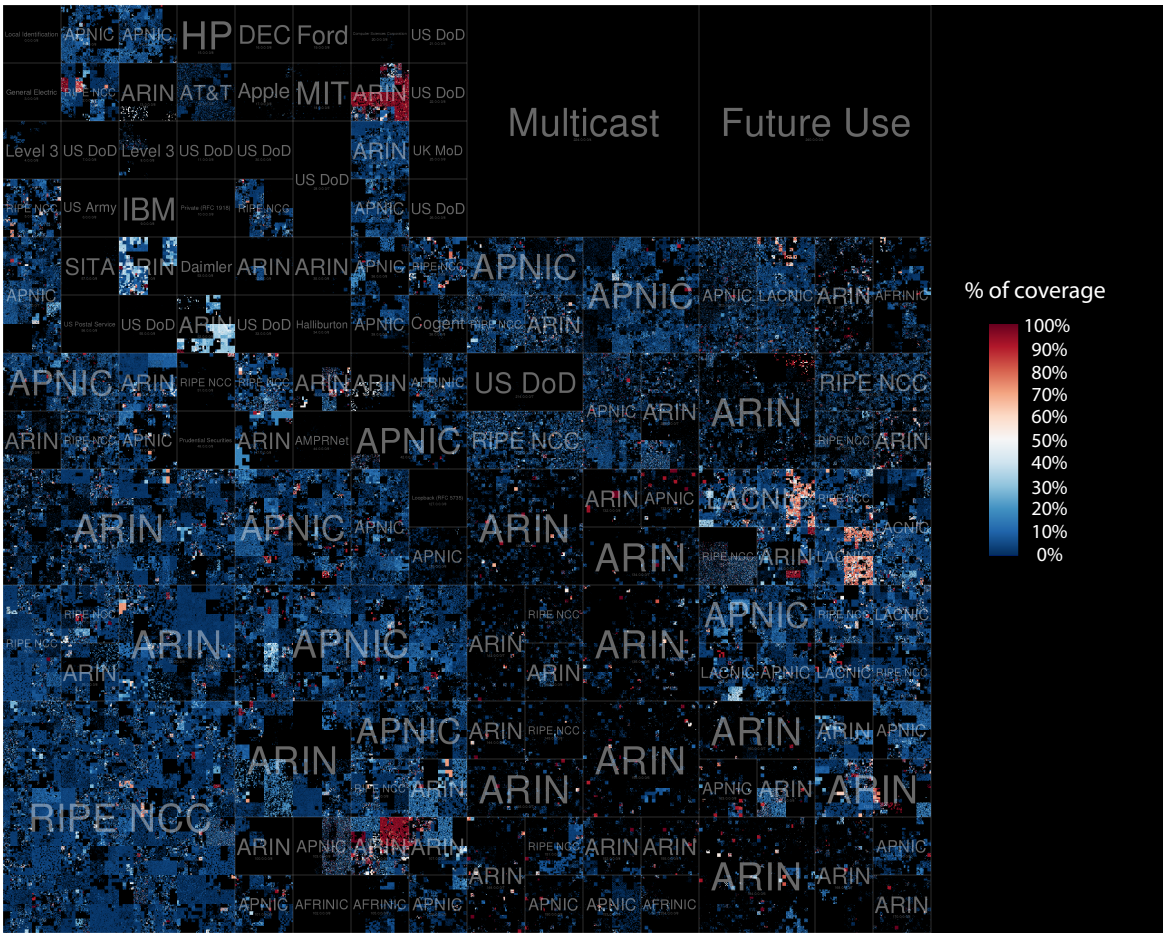


Figure 8: Map IPv4 addresses to Hilbert curves

There are multiple areas where no nodes were found, which could have been for multiple reasons such as:

- Nodes in those areas were not running services on the ports we scanned;
- Multicast IP addresses;
- **IP addresses in those areas are on our blacklist and we did not scan them.**

Note: This hilbert map was generated using the "ipv4-heatmap" tool which can be found on the following website: https://github.com/hrbrmstr/ipv4-heatmap

Scoring Mechanism

To estimate the vulnerability of an IP address, we retrieve information from the NVD (National Vulnerability Database) repository[1]. NVD is the U.S. government repository of standards based vulnerability management data represented using the Security Content Automation Protocol (SCAP). NVD includes databases of security checklists, security related software flaws, misconfigurations, product names, and impact metrics. For our analysis, we used the CVE and CVSS data available on the NVD repository.

CVE[2] stands for Common Vulnerabilities and Exposures and it aims to provide a common "enumeration" for known cyber security issues. It consists of a list containing information regarding security vulnerabilities and exposures. Each CVE entry is composed by a standard identifier number, a brief description, references to related vulnerability reports and advisories.

The CVE description is a free-text entry where the vulnerability consequences are described as well as the products and versions affected by it. Since only from this free-text entry description is possible to know the exact products and versions affected by the vulnerability, we applied NLP (Natural Language Processing) techniques in order to retrieve this information.

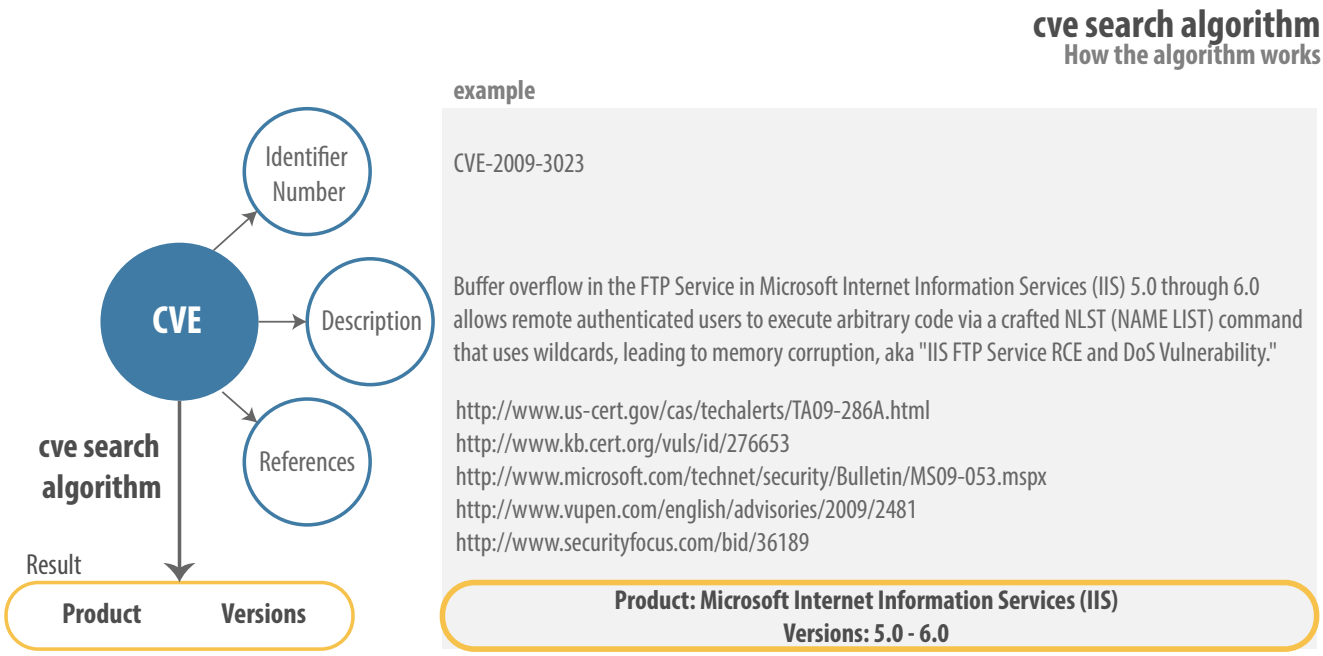


Figure 9: CVE search algorithm

Hence, for each IP/port that we found running a service, we searched for all CVE that affected that product and version.

In order to quantify the vulnerabilities, we then used the CVSS (Common Vulnerability Scoring System)[3]. NVD provides a CVSS version 2 (although version 3 was launched in 2015, NVD still uses version 2) for all CVE. CVSS provides standardized vulnerability scores and are calculated based on three metric groups:

- **Base:** Intrinsic qualities of a vulnerability
- **Temporal:** characteristics of a vulnerability that changes over time
- **Environmental:** aspects of the vulnerability that are unique to a user's environment

[1] "National Vulnerability Database", retrieved from https://nvd.nist.gov/
[2] "CVE - Common Vulnerabilities and Exposures (CVE)", retrieved from https://www.cve.mitre.org/

Although both Temporal and Environmental metrics are provided on the CVSS original framework, they are optional and NVD only uses the base scores.

The Base metric group is composed by several factors:

- **Access Vector**
- **Access Complexity**
- **Authentication**
- **Confidentiality Impact**
- **Integrity Impact**
- **Availability Impact**

These six factors can be divided into two different groups:

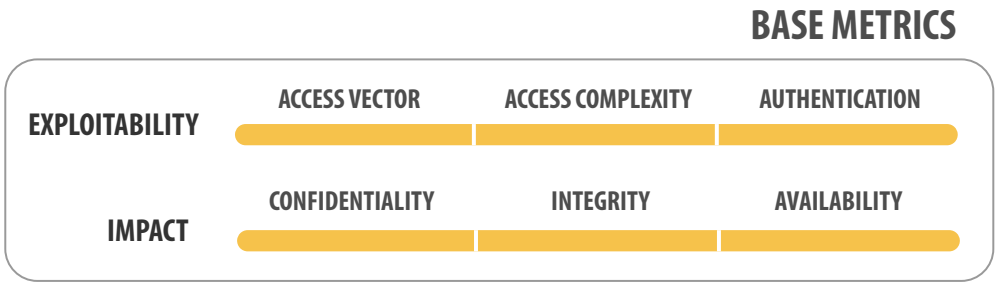


Figure 10: CVSS base metrics

Hence the base score dependends on the exploitability of a vulnerability and the impact of that exploration.

The exploitability, that includes the Access Vector, Access Complexity, and Authentication metrics, capture how the vulnerability is accessed and whether or not extra conditions are required to exploit it.

The impact, that includes the confidentiality, integrity and availability impact metrics, measure how a vulnerability, if exploited, will directly affect the confidentiality, integrity and availability of an IT asset.

The Base metrics produce a score ranging from 0 to 10 and NVD labeled the vulnerabilities based on the CVSS score:

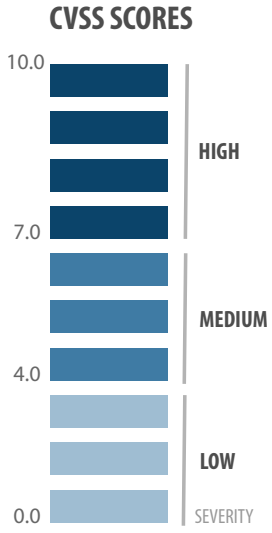


Figure 11: CVSS scores

[3] "Common Vulnerability Scoring System (CVSS-SIG)", retrieved from https://www.first.org/cvss

Below is presented the CVSS score distribution of all CVSS data available on the NVD repository:

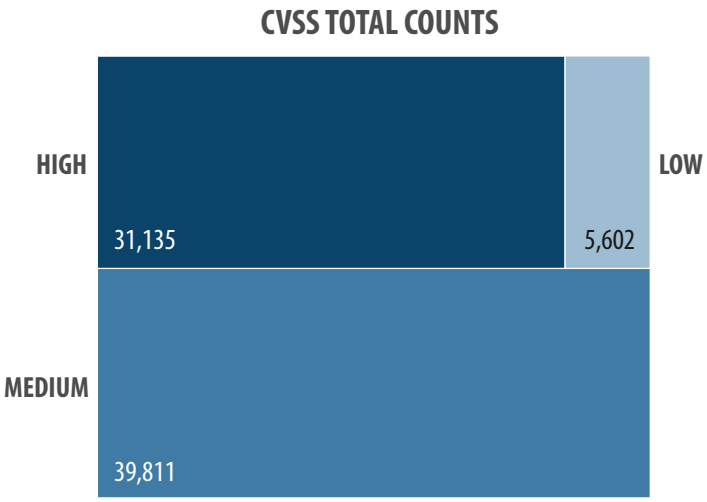


Figure 12: CVSS score distribution of CVSS on the NVD repository

For more details, including the formulas and parameters values, please consult the complete Guide to the Common Vulnerability Scoring System Version 2.0.[4]

In our scoring mechanism, after having all the CVSS's values for a product's version we sum those values. However, it's important to keep in mind that products' versions with the same score can have vulnerabilities of different severities.

It is crucial to mention that we can only use our scoring mechanism if we have the CPE (Common Platform Enumeration) for the service, which is a combination of both the product and its version. Therefore, one can infer that the amount of vulnerabilities is much higher than we can find. It can also happen that we have product and version but they don't have any CVEs associated to them, although they have known vulnerabilities.

It's important for the reader to understand that here we're using a direct banner based comparison and not considering that some of the IP addresses might be using backported fixes of those software versions.

[4] "CVSS v2 Complete Documentation", retrieved from https://www.first.org/cvss/v2/guide

Encryption

One good way of measuring the evolution of the security of the internet is by looking at the number of services that use some type of encryption versus clear/plain text. This chapter focuses exactly on that as we will analyze and compare multiple servers versions (encrypted and unencrypted).

In the early days of the internet, there was no sense of need for security. All that mattered was having the network of machines "talking" to each other. The more people started to "join" the internet, the more important became the information transmitted by those machines that talked to each other. The nature of these communications raised the need for encryption, so that guarantees of security and confidentiality of the data exist.

HTTPS/ HTTP

Web Servers are services that process requests to deliver webpages/applications to users. Web Servers use protocols to communicate: HTTP is a basic unencrypted network protocol while HTTPS has an extra layer of security and has its connection encrypted.

We analysed the main ports for each of these services (port 80 for HTTP and port 443 for HTTPS) as well as their typical alternative ports (ports 8080 and 8443, respectively).

First of all, we analysed how many IP addresses were running HTTP and HTTPS services in their main ports. The image below represents the results found.

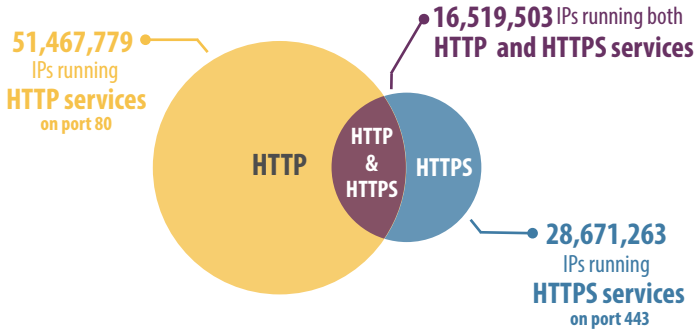


Figure 13: Distribution of IP addresses running encrypted and unencrypted services

It's unfortunate to see that there's such a high number of HTTP servers in use. Hopefully over the next few years with browsers starting to reject HTTP, we will see this number decreasing.

The image below represents the number of IP addresses found on the four ports analysed.

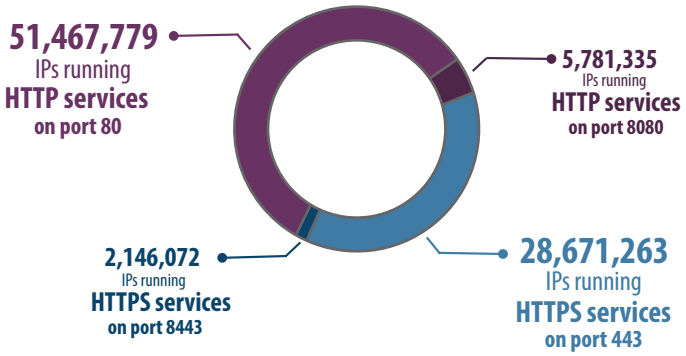


Figure 14: Distribution of IP addresses by the main and alternative ports of HTTP and HTTPS

The number of unencrypted HTTP services still currently overtakes the number of HTTPS services.
One positive note is that over time we've been watching the gap between the hosts running HTTP and HTTPS diminish.

SSH/ Telnet

SSH and Telnet are both network protocols, the primary difference between them being that SSH provides the user with an encrypted connection while the data transmitted through **Telnet is clear-text**. It's important for us to explain that Telnet suffers from the same problem as HTTP: in case you connect with Telnet while being connected to a wireless network with someone else, **all the credentials will be visible** by hackers in that network.
SSH provides a secure channel between a client and a server. Commonly used to manage remote machines and transfer files in a secure form, SSH is usually found on TCP port 22, while Telnet is found on port 23. Our study will analyse not only this service use and exposure but also extract other data such as the encryption algorithms and keys used.

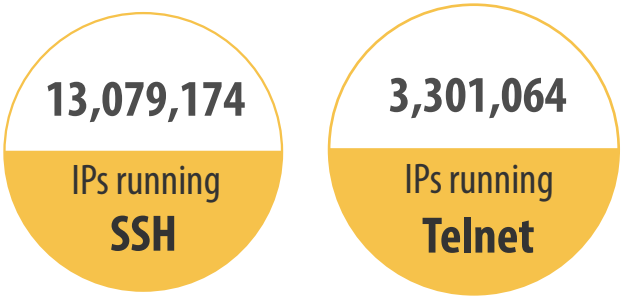


Figure 15: IP addresses running SSH and Telnet protocols

Telnet is really an extremely outdated protocol and even if we're seeing a much higher number of SSH devices, ideally we want to see in the future the number of Telnet going down even further. SSH makes for a much better choice not just at a security level but even at a usability level as it allows for session resumption, has compression and of course on top of all this a huge amount of security features built in.

FTPS/ FTP

The File Transfer Protocol is a network protocol used to transfer files between a client and server. FTPS is the same as FTP but with an SSL tunnel.

FTP is still clearly a widely used service, with the gap between unencrypted and encrypted choices being huge. We also found that 248,261 IP addresses were using both FTP and FTPS.

This huge gap is quite unfortunate as FTPS was built to fix the lack of security design that FTP has since it wasn't initially designed with security concerns in mind. In FTPS there are two SSL modes, implicit on which the use of SSL is implied and any connection established by the client that doesn't use SSL is refused by the server, and the second mode explicit on which the client and server negotiate the level of protection used, this is quite useful as it allows the server to support both unencrypted FTP and encrypted FTPS sessions on the same port.

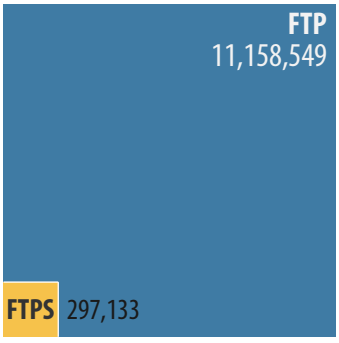


Figure 16: Number of IP addresses running FTP and FTPS

Email Protocols

An email protocol can be defined as a set of rules that are used to transmit information between an email server and the user. There are protocols used for sending emails (SMTP) and protocols used for receiving emails (IMAP and POP3).

- **POP3/ POP3S:** Post Office Protocol is used by email clients to retrieve email messages from a mail server over TCP/IP. However you can only use one computer to check the email, and they get stored on the local computer instead of the remote server, making this a very limited protocol.
- **SMTP/ SMTPS:** Simple Mail Transfer Protocol is used for email transmission. SMTP is typically used to send messages to a mail server for relaying, while for retrieving applications usually POP3 or IMAP are used.
- **IMAP/ IMAPS:** Internet Message Access Protocol is used by email clients to retrieve email messages from a mail server over TCP/IP. IMAP is usually the better and recommended option as it does not delete the emails from the remote server, therefore you can just tap into your synced account and get all updated content.

The image below represents the number of IP addresses found for encrypted and unencrypted email protocols.

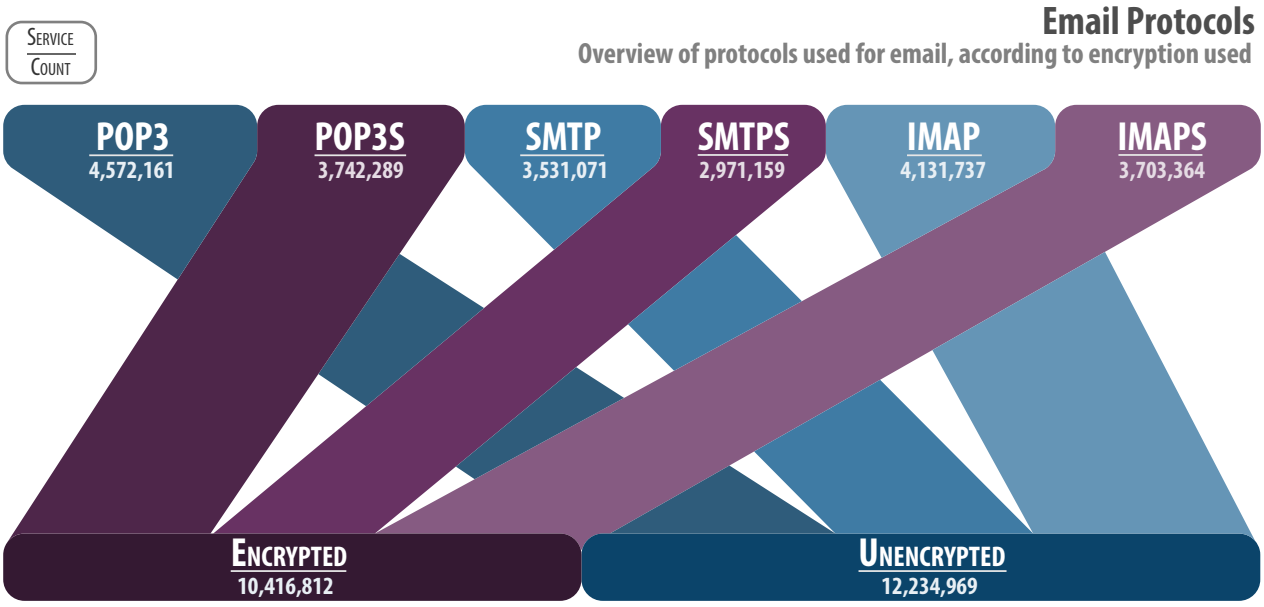


Figure 17: Overview of email protocols usage, according to encryption

Email services were initially designed as cleartext, however across the years providers have upgraded to services that make use of SSL. Here the gap is really close and as with HTTP/HTTPs services, should be getting closer with every passing month.

Web

Port 80: Web servers and Web frameworks

Port 80 is the most used port out of all that we scanned. We decided to split this analysis in half and we will first look at the top web servers and follow with a look at the top web frameworks/ technologies used.

So, starting with the web servers, here is what we found:

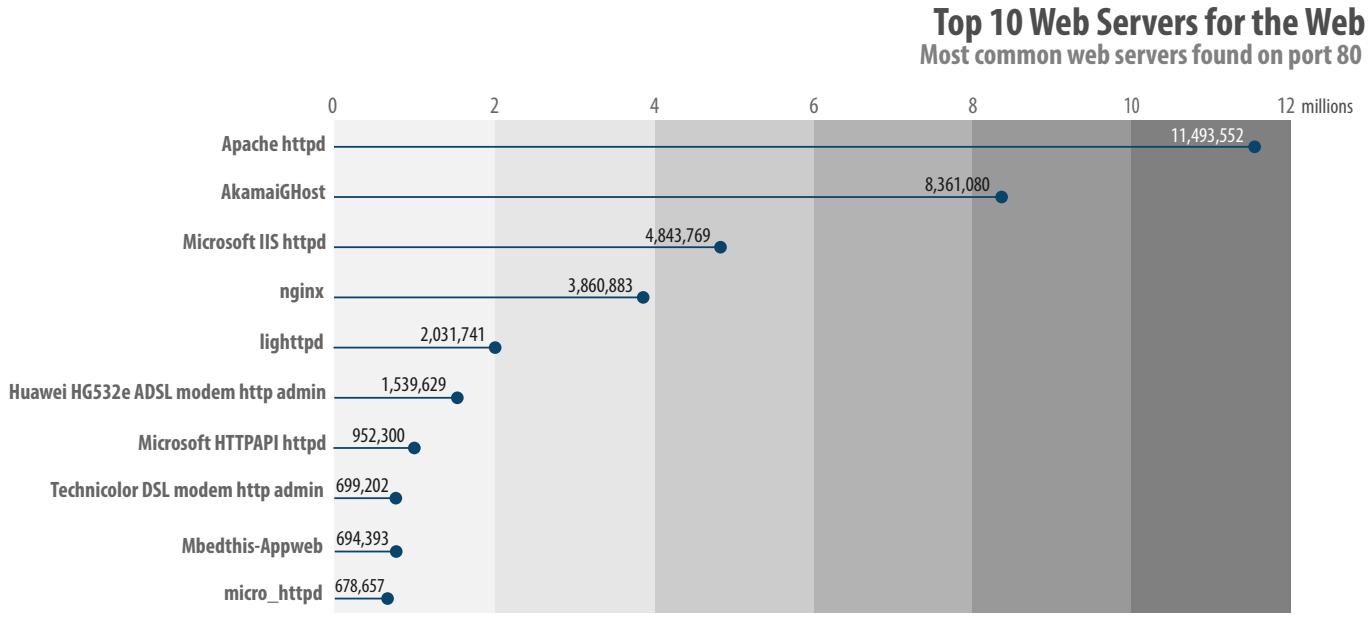


Figure 18: Top10 web servers for the web

Apache leads the run with almost **11.5 million** of web servers. The version most used by this webserver is 2.2.15, which was released on the 15th of March 2010, meaning that this is an extremely outdated version of Apache. This product scored a value of **109.5** in our scoring mechanism, mainly due to a high number of vulnerabilities of medium severity, even though there are a few of high severity.

For Microsoft IIS, the top version found was 7.5, which was released in March 2009 and it's currently being used in 1,687,489 web servers. There were no patches released for this version after 2013, which means that it is actually past the end of life since that year. Since this version has multiple vulnerabilities of high severity, it scored a value of **46.3** in our scoring mechanism. The top of the most used versions of this web server include 6.0 (released in April 2003) and 8.5 (released in October 2015).

Finally, the most common version of Nginx that was found was 1.4.6, which was released on the 4th of March 2014. Once again, this is an old version of a web server and it is actually vulnerable to a code execution (more details can be found on CVE 2014-0133). This version has a score of **22.5** in our scoring mechanism.

As mentioned before, it's important to keep in mind that sometimes it's not possible to retrieve the version of products. Therefore, the amount of vulnerabilities is much higher than we can find.

For the second part of the analysis of port 80, we will look at the top web technologies used on the IP addresses that had an HTTP service on port 80. We can't go without mentioning that we excluded every technology that was a web server (Apache, IIS, NGINX, Tomcat) since we have already covered them in the previous section.

So, for now, we we will focus on the most common web technologies found on port 80.

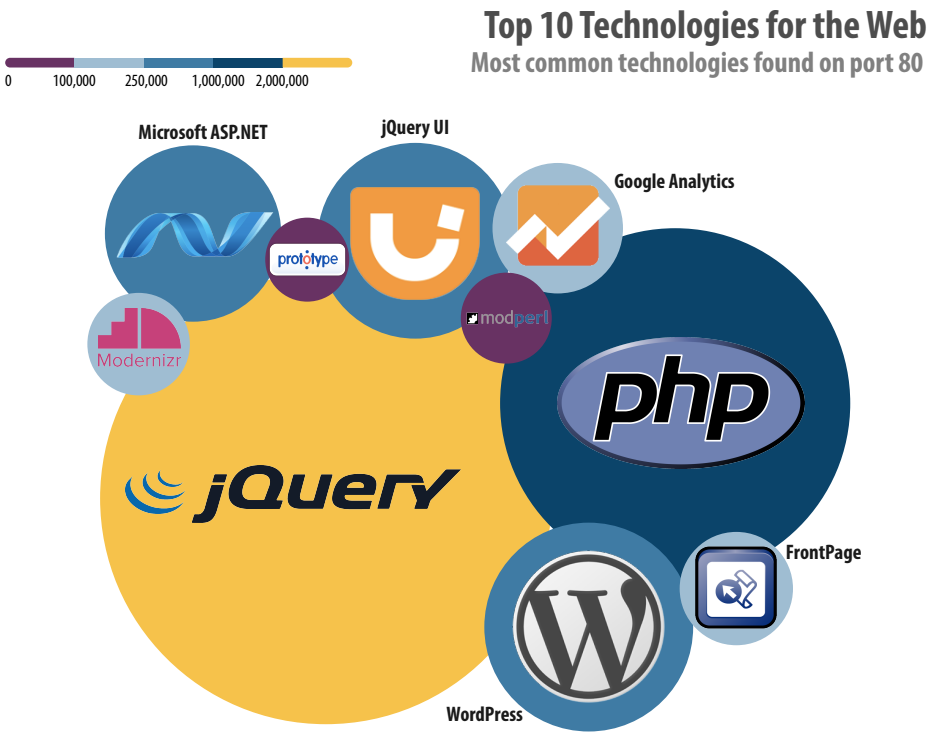


Figure 19: Top10 technologies for the web

It's interesting to still find FrontPage so prominent since the last stable release was in 2007 and it has since been replaced with Microsoft Expression Web. Its most used version (4.0) scored a value of **10.0** on our scoring mechanism, having two vulnerabilities of medium severity.

For jQuery, the most used version is 1.7.1, which was released on the 21st of November 2011. This is an extremely old version of jQuery, which is also extremely prone to vulnerabilities - a simple Google search will show that this version is vulnerable to multiple XSS attacks. This version scored a value of **13.6** in our scoring mechanism, having two vulnerabilities of medium severity.

On the PHP front, things aren't looking much better. The most used version is 5.3.3 which was released on the 22nd July 2010. With PHP being a technology that is already heavily criticized for security issues and slow patches, using old versions is most definitely not recommended. This version of the product scored a value of **1497.8** in our scoring mechanism, due to a high number of vulnerabilities of medium severity and a few of high severity.

Wordpress, a well know blog framework, has 4.5.3 as the top installed version. This was a surprising breath of fresh air, as this version is quite modern and was released on the 18th of June 2016.

For Microsoft ASP.NET, the most found version was 4.0.30319. Once again, this is unfortunate as this version is vulnerable to a critical bug which can be found on CVE-2011-3416 and consists of forms authentication bypass. This version scored a value of **15.0** in our scoring mechanism.

All in all, this paints a grim picture in terms of the updates of software, something that is a huge problem for the information security industry.

Port 443: SSL

Secure Socket Layer (SSL) is a technology for establishing an encrypted link between a web server/service and a browser/client. The link, when correctly configured, ensures that all the data passed between the server and client remains private and integral.

In this section, we are going to analyse SSL, focusing on the most important details of this technology. In reality, we did 2 independent analysis where we separated the root certificates from the leaf certificates and then analyzed both datasets independently. The reason for this is that if we had mixed both, the root certificates would dominate the analysis.

In cryptography, a root certificate identifies the root certificate authority, its usually unsigned or self-signed. A certificate authority can then issue multiple certificates in the form of a tree structure. The term leaf certificate is usually used to indicate the last certificate found on the certificate chain.

Starting with the SSL certificates, we wanted to get a sense of the amount of valid certificates in comparison with expired ones. We considered expired certificates the ones wth expiration date before August 2016. Therefore, we looked at the dates of expiration and found that out of all the **37,970,903** certificates we extracted, only **1,226,042** were expired. Here is the breakdown by root and leaf certificates.

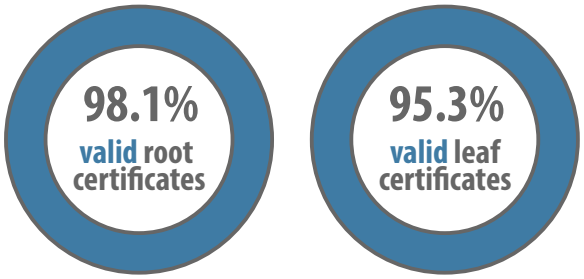


Figure 20: Validity of root/leaf SSL certificates

Next, we analysed the encryption algorithms used in the certificates, considering the key size and the security guidelines for these algorithms.

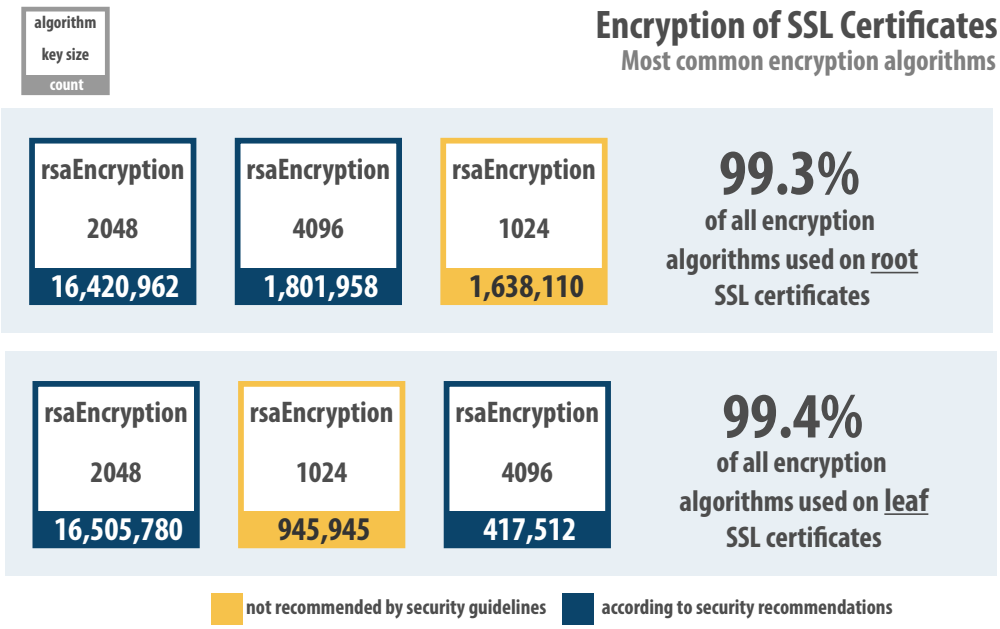


Figure 21: SSL encryption algorithms

Looking at the general picture, one can see that the great majority of the encryption algorithms used follow the security recommendations.

However, we still found a lot of certificates using RSA with 1024 bits, which is not recommended by security guidelines. Furthermore, these certificates have already started to be phased out by multiple browsers (an announcement with details by Mozilla can be found in their blog[5]). Since this phasing off went into process in 2013, it's surprising that we still can find a huge number of certificates using this encryption algorithm.

Still focusing on the SSL certificates, we proceeded to analyse their signatures. Essentially, the certificates' signatures guarantee that the content of the messages sent has not been altered during transmission. Each certificate should have a unique signature.

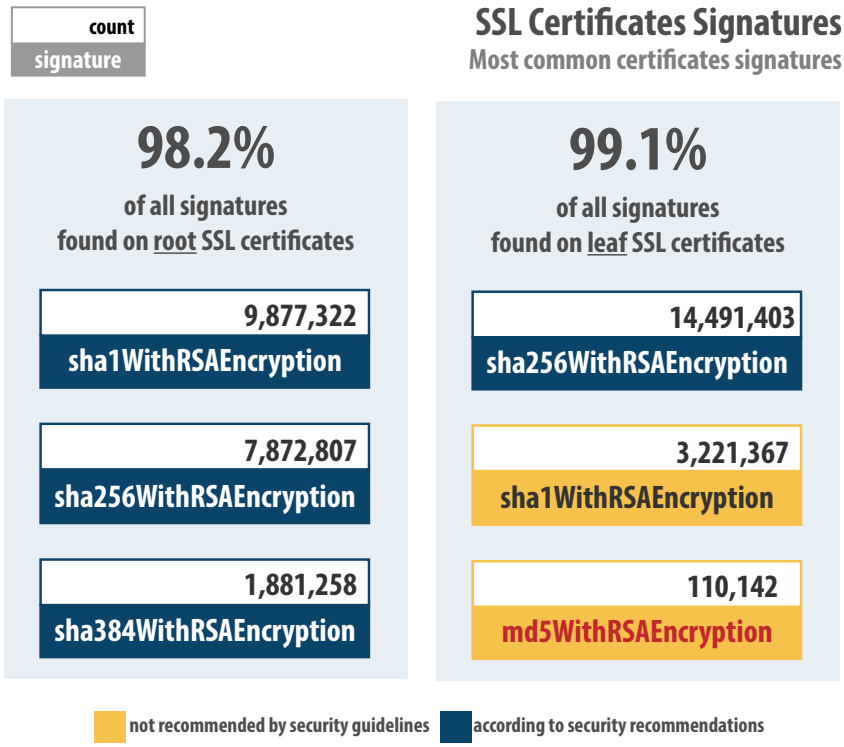
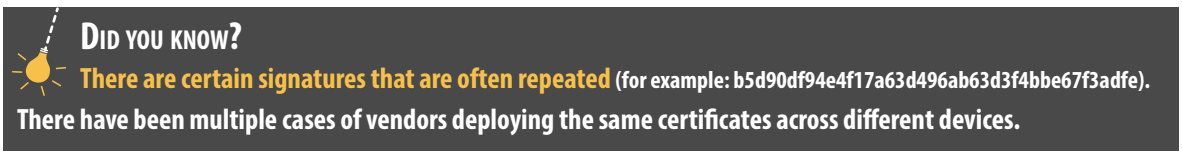


Figure 22: SSL certificates signatures

Much like we observed on the encryption algorithms, certain signature algorithms have also been starting to be phased out. **SHA1 and MD5 are both extremely old algorithms** - attacks for both these algorithms and problems (such as hash collisions) have been found - hence the recommendation for deprecation. As signatures are only validated on intermediate/leaf certificates, having SHA1 encrypted on the root certificates isn't in theory a problem, hence why we marked in yellow the SHA1 only on leaf certificates.

Nevertheless, it's important to make a distinction between the possible attacks. Easy viable attacks against MD5 have been found, therefore making it really insecure, while for SHA1, only state sponsor level attacks have been found.



[5] "Phasing out Certificates with 1024-bit RSA Keys, retrieved from https://blog.mozilla.org/security/2014/09/08/phasing-out-certificates-with-1024-bit-rsa-keys/

Online Certificate Status Protocol (OCSP) allows a web server to query the OCSP responder and to cache the response. By doing this, the web server can check the validity of its certi ficates and the client doesn't have to contact the certi ficate authority.

Out of all the IPs running HTTPS services on port 443, only 7.4% supported OCSP Stappling.

On the next part of this section, we are going to focus on some of the vulnerabilities that affect SSL, in particular, the Heartbleed bug and CCS injection.

Heartbleed is a vulnerability of OpenSSL that allows the attackers to access the memory of data servers and retrieve critical and sensitive information from servers affected by this vulnerability.

We found 101,346 IPs vulnerable to Heartbleed in the world, which means that the confidentiality of the communications made from these IP addresses might be compromised and sensitive data might be stolen.

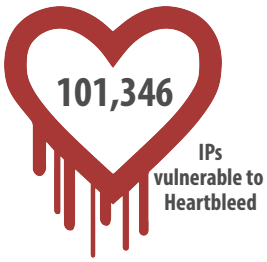


Figure 23: Heartbleed

In the map below, it is possible to see the worldwide distribution of the IPs vulnerable to Heartbleed. One can see that the United States is, by far, the country with a higher number of IPs vulnerable to this bug, followed by China.

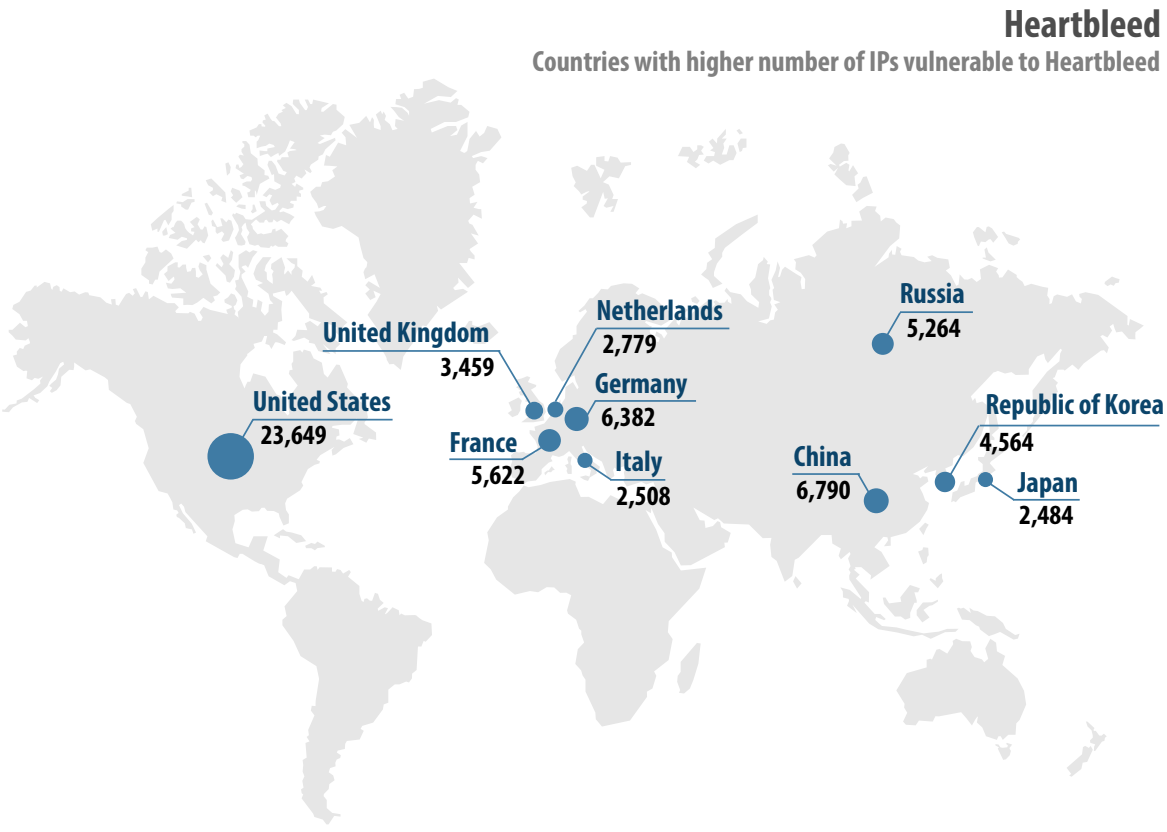


Figure 24: Countries with higher number of IPs vulnerable to Heartbleed

CCS (ChangeCipherSpec) Injection is an attack that works because SSL accepts the CCS inappropriately during a handshake. In the right conditions, it allows for an attacker to man-in-the-middle encrypted connections.

When analysing the data from the entire world, we found 7,630,687 IP addresses vulnerable to CCS injection. The following image shows the countries with higher number of IPs vulnerable to CCS Injection.

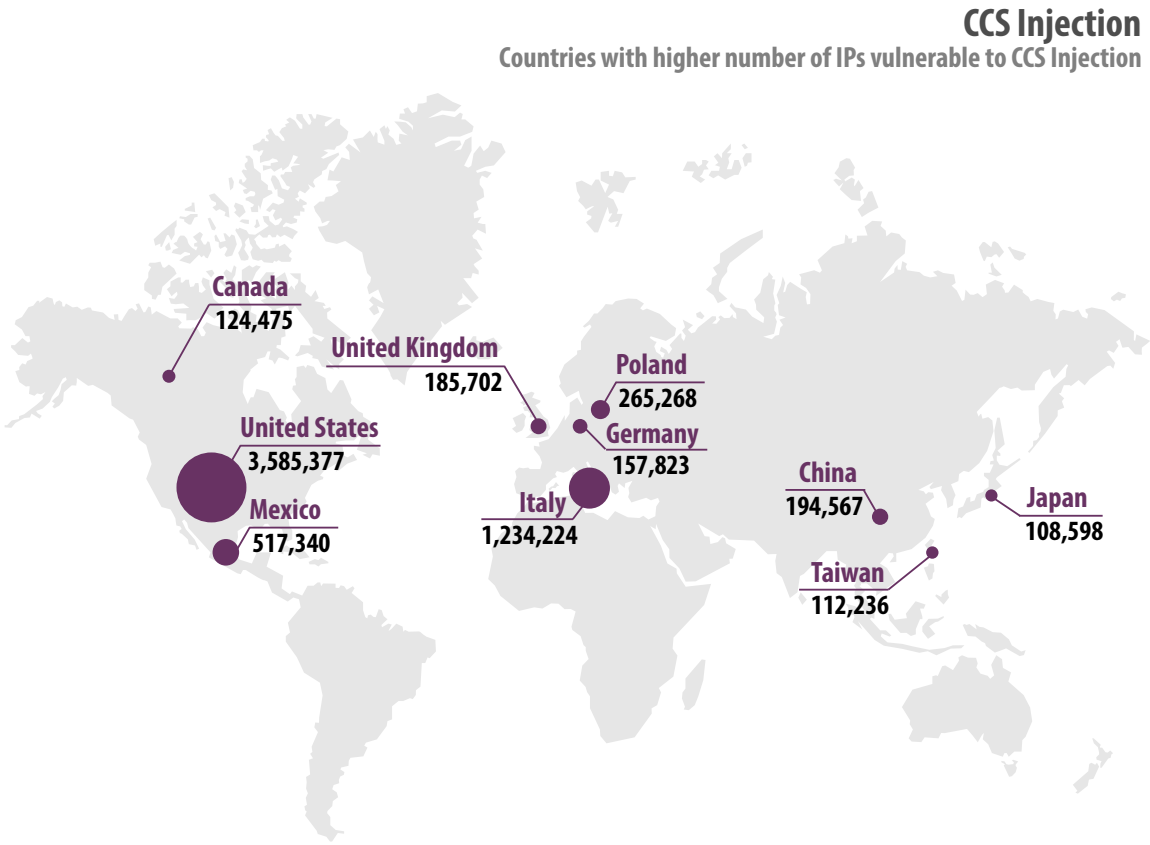


Figure 25: Countries with higher number of IPs vulnerable to CCS Injection

For more information, check the blog[6] created by the original bug founder. Another interesting blogpost[7] can be found on the ImperialViolet blog which goes into further detail about this vulnerability.

On the overall analysis of SSL, these were the results that really caught our attention:

- The amount of IP addresses that were still vulnerable to Heartbleed was far too high, this being a critical, easy to exploit vulnerability makes it a real danger to organizations that still haven't fixed this problem across their networks.
- A positive result was found on the validity of certificates.

[6] "How I discovered CCS Injection Vulnerability", retrieved from http://ccsinjection.lepidum.co.jp/blog/2014-06-05/CCS-Injection-en/index.html
[7] "Early ChangeCipherSpec Attack", retrieved from https://www.imperialviolet.org/2014/06/05/earlyccs.html

Data Storage

Big data is one of those buzzwords used by marketing people to sell a product or a technology. From a security perspective, we believe big data should have that name merely for the big amounts of data that is available on the internet due to the lack of security.

In this section, we will look at different technologies that are used for big data:

- **MongoDB** is a NoSQL database that sells itself as highly scalable, performant and agile.
- **Redis** is a key-value cache and store. It is a very well known and used technology.
- **Memcached** is a general-purpose distributed memory caching system. It is often used to speed up dynamic database-driven websites by caching data and objects in RAM to reduce the number of times an external data source (such as a database or API) must be read. You can also find it along side with Couchbase installations.
- **Elasticsearch** is a distributed and scalable system that allows for search and data analysis in real time. It is schema-free, which means that the user has full control on how the data is indexed.
  Almost every action can be done using a RESTful API (using JSON over HTTP).
- **MySQL** is an open source relational database. It is one of the most used RDBMS (Relational DataBase Management System) in the world.
- **MSSQL** is a relational database management system developed by Microsoft.

We started by looking at the number of instances online of the first 4 technologies mentioned and these are the results we found:

- **MongoDB: 58,942** instances online
- **Redis: 22,757** instances online
- **Memcached: 149,324** instances online
- **ElasticSearch: 67,430** instances online

Knowing the number of instances, we also wanted to know how it was reflected in the amount of data exposed. Since we had already published two blogposts[8][9] where we analysed these technologies, we thought it should be interesting to see the changes in the amount of data exposed in the period of a year, since the first blogpost.

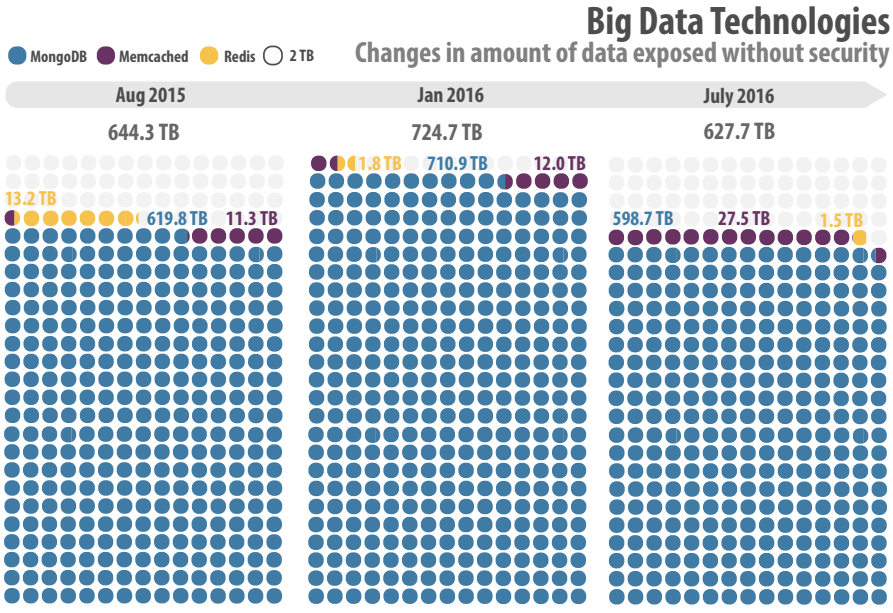


Figure 26: Data exposed from Big Data Technologies

[8] "Data, Technologies and Security - part 1", retrieved from https://blog.binaryedge.io/2015/08/10/data-technologies-and-security-part-1/
[9] "Data, Technologies and Security - part 2", retrieved from https://blog.binaryedge.io/2016/01/19/data-technologies-and-security-part-1-2/

Overall, from January to July of the current year, we found a decrease on the amount of data exposed. However, an increase in the data exposed by memcached happened - in January we had found 12.0 TB exposed and in July there's more than double of data exposed: 27.5 TB.

A big reduction was seen on Redis with the amount of data exposed going from 13.2 TB in August 2015 to 1.5 TB in July 2016. It was found a ransomware that deletes Redis data[10], which can be the reason for this reduction.

Another interesting point we looked at was the versions of ElasticSearch that were used and here are the results found.

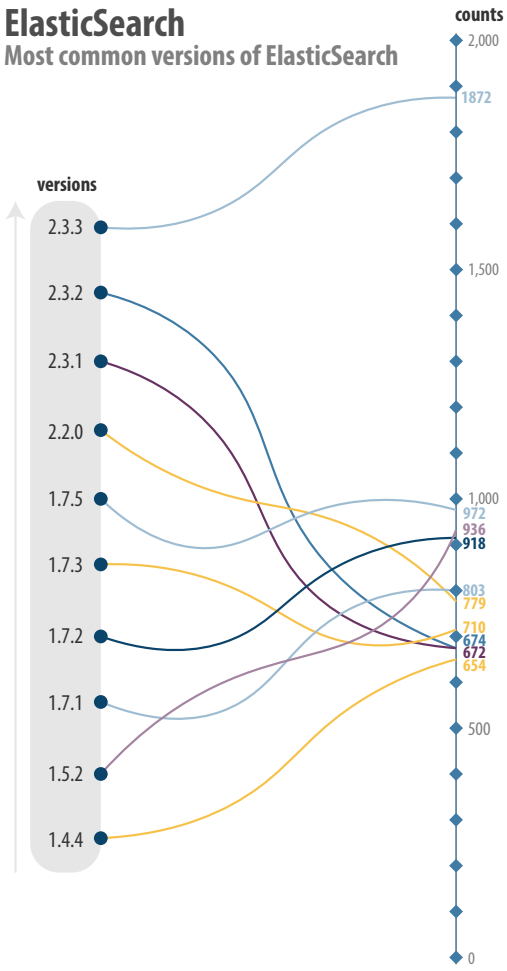
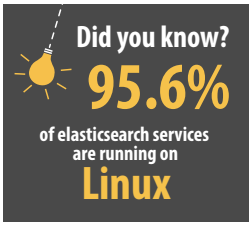


Figure 27: Most common versions of Elasticsearch

We can see that for ElasticSearch a relatively new version is used most widely, version 2.3.3 was released on May 17th of the current year, with the following most used version being 1.7.5 which was released on February 2nd of the same year. At a first look, versions like 1.7.5 and 1.7.3 might appear quite old. This is due to the release cycle of Elasticsearch and the different builds they have.



Out of this list only version 1.4.4 is past the end of life date, with 1.5.2 reaching the end of life by 23rd September 2016.

[10] "Duo Labs: Over 18,000 Redis instances targeted by fake ransomware", retrieved from https://community.duo.com/t/duo-labs-over-18-000-redis-instances-targeted-by-fake-ransomware/319

For ElasticSearch we're also able to observe which versions of the Java virtual machine were running on it and what we found was that many of the elasticsearch instances are running versions of java that are still affected by multiple vulnerabilities, some of them quite critical.

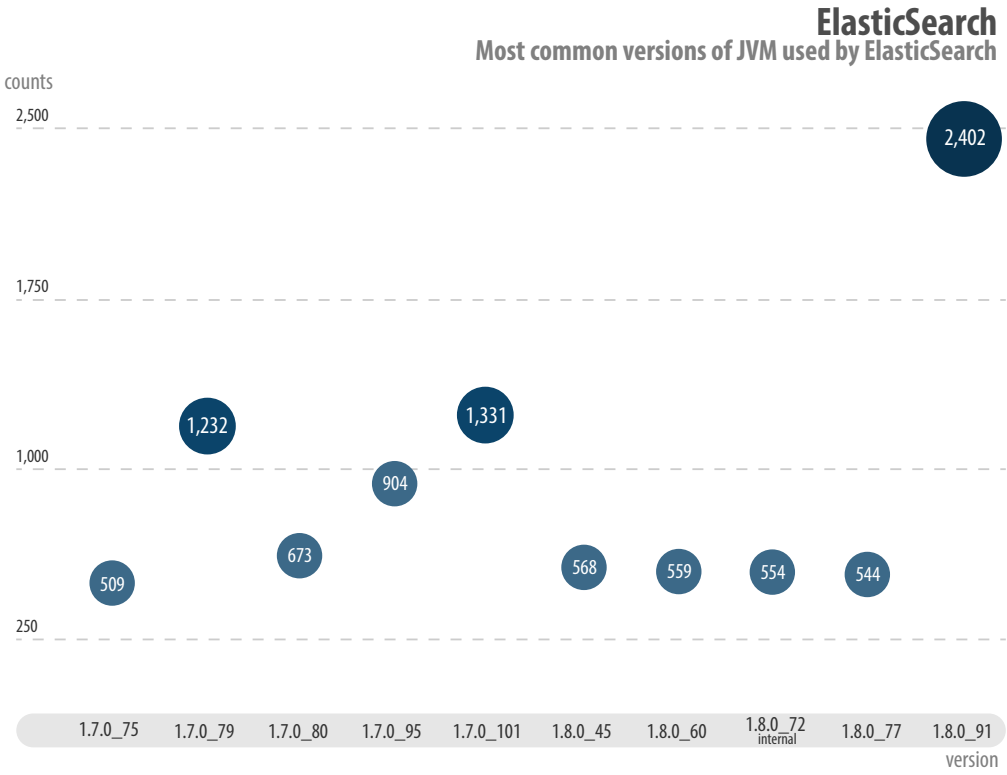


Figure 28: Most common versions of JVM used by ElasticSearch

We also looked at MySQL and MSSQL as these are some of the most used databases in the world. In terms of number of instances we found the following:

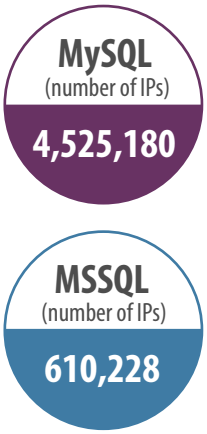


Figure 29: Number of IP addresses using MySQL and MSSQL

One interesting observation is that the versions used of both these softwares are quite old.

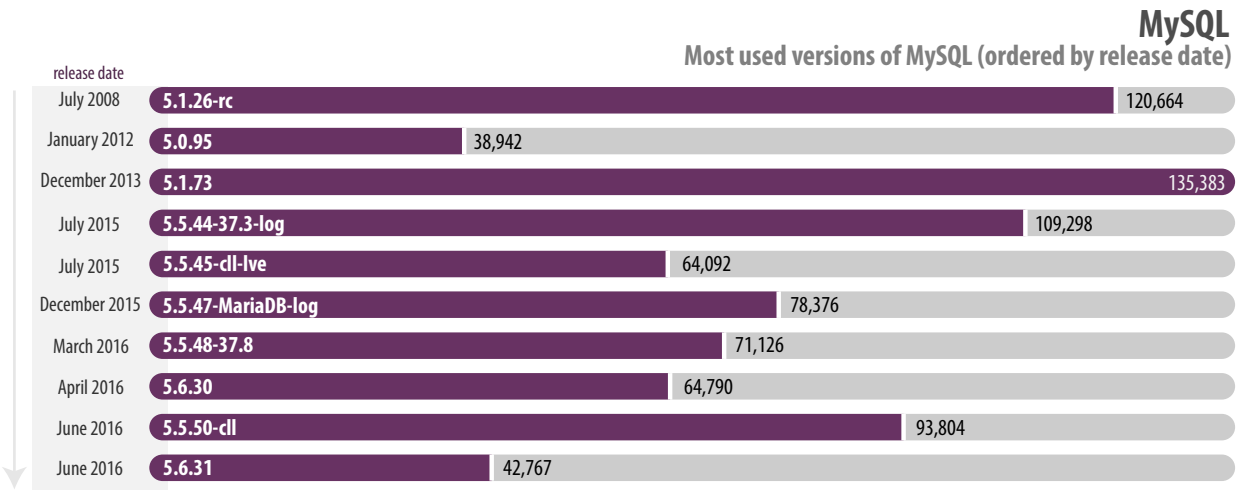


Figure 30: Most used versions of MySQL

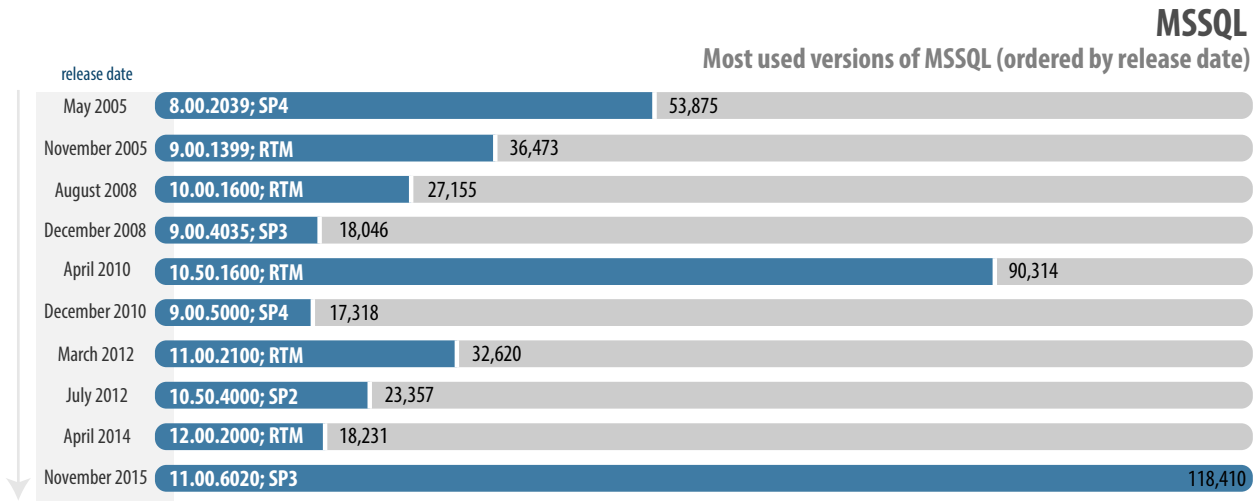


Figure 31: Most used versions of MSSQL

Looking at MySQL we tried to understand what could be the reason for still finding so many old versions. The first thing we did was look at the CentOS package repo as it is used quite often in the enterprise world. There we noticed that for CentOS 6 the default MySQL version is 5.1.73-7, and for CentOS 5 is MySQL 5.0.95-5, what this means is that most likely companies install base versions of MySQL and then don't update them.

Remote Management Services

Remote Management Services allow you to access/ manage remote machines and are typically used by system administrators and devops.

For this chapter, we chose the following services:

- **VNC (Virtual Network Computing)** is a service that allows you to access your machine from anywhere in the world. As one might understand the benefits of this service, one can as easily understand that it's necessary to protect this access with some type of authentication.
- **RDP (Remote Desktop Protocol)** is a proprietary protocol developed by Microsoft, that like VNC allows remote administration of systems. RDP is focused on Windows systems.
- **X11** is a windowing system that can also be used to remotely access Unix systems.
- **SSH** is a network protocol that allows an encrypted transmission of information.

VNC, RDP and X11

If you have ever managed a fleet of remote machines you have probably used one of these.

We've previously touched upon the dangers of VNC on our blogpost entitled "VNC image analysis and Data Science"[11]. On this blogpost, we showed some of the images of critical systems that were connected to the internet without any type of authentication (also seen on the following image).

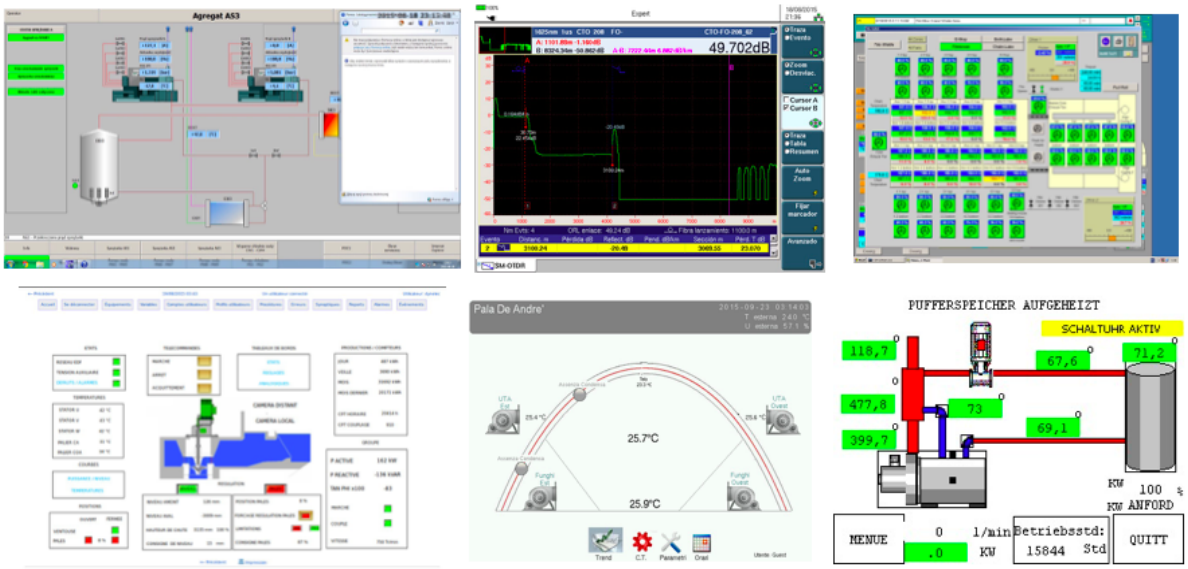


Figure 32: Examples of critical systems connected to the internet without authentication

Analysing the data gathered for these three services, we obtained a total of 2,951,950 images. It's fundamental to mention that we found even more instances of this services, however, we could not extract images of all of them.

[11] "VNC image analysis and Data Science", retrieved from https://blog.binaryedge.io/2015/09/30/vnc-image-analysis-and-data-science/

Here is the breakdown by service:

- **13,172** VNC images
- **2,936,393** RDP images
- **2,385** X11 images

Having this amount of images, we had to come up with a way to extract useful, usable information that is easily accessible, sortable and searcheable out of this dataset quickly. So we prioritized our needs and defined four problems related with these images that we wanted to solve using Data Science and Machine Learning techniques:

- **Huge amount of images:** As mentioned, there are a lot of IP addresses out there that expose these services;
- **Similar images:** Lots of the devices are locked or just have similar screensavers services;
- **"Boring" images:** Quite a big number of "locked" Microsoft Windows systems or Linux consoles requesting login were found. We consider these as "boring" images since there is not much information to gather;
- **Ability to extract data and metadata** that makes this an interesting dataset, such as emails, logos, faces and text.

In order to address these problems, we came up with a workflow that all of the images our platform captures go through.

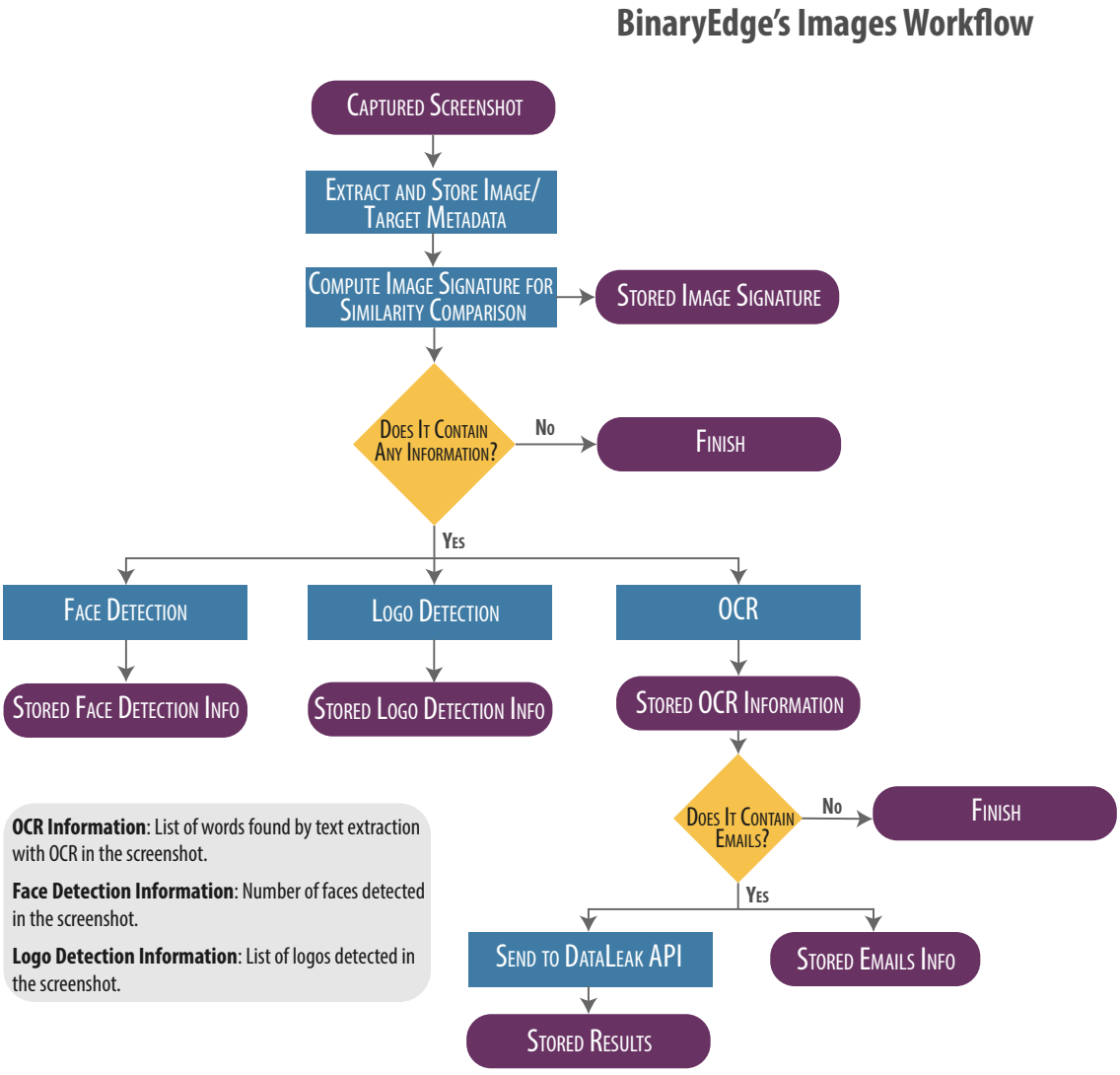


Figure 33: BinaryEdge's images workflow

Additionally, having the results stored in this way allows both us and our clients to quickly search, sort, enrich image related data and metadata on our Web Platform instead of having to manually search through millions of images.

Considering the images collected by our platform, we extracted their metadata, following the process explained in the image workflow. We then generated these wordclouds for each service, containing the 20 words most found in each one.

Figure 34: Most common words found in VNC screenshots

Figure 35: Most common words found in RDP screenshots

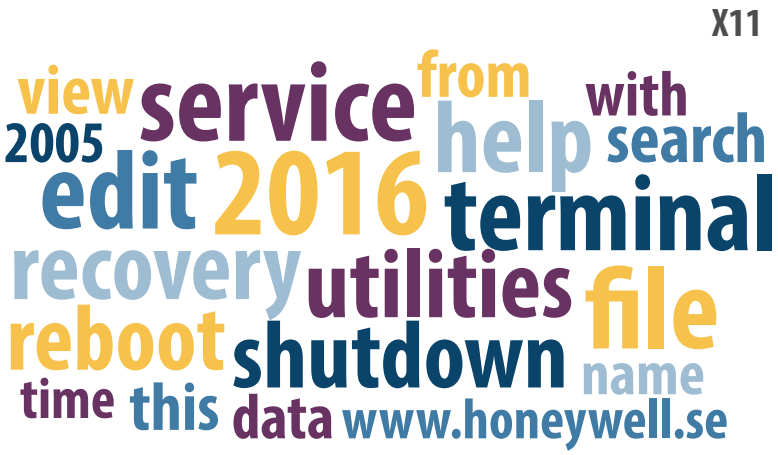


Figure 36: Most common words found in X11 screenshots

SSH

We have previously conducted a study on SSH[12] and, for the purpose of this report, we will do some comparisons with the results found at the time of writing of that blogpost.

As mentioned in the first chapter of this report, we found **13,079,174** IP addresses running SSH. We further explored this service by looking at different parts of this protocol.

We started by looking at SSH banners, which are the part of the data that tell us which version of SSH is running on the IP Address.

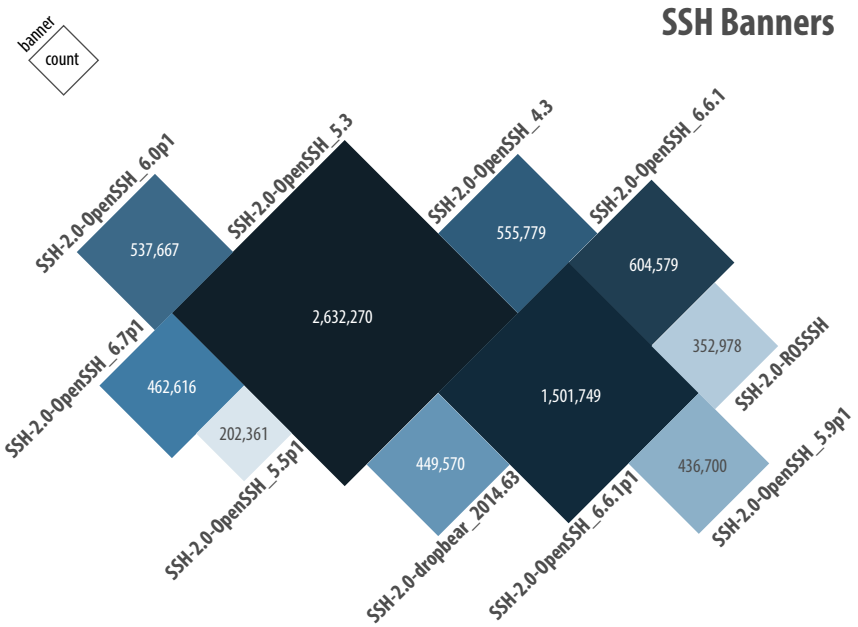


Figure 37: SSH banners

Looking at the previous image, one can see that the version most commonly found of OpenSSH is 5.3 with a staggering number of **2,632,270** IP addresses running it, followed by OpenSSH 6.6.1p1 with **1,501,749** IP addresses.

When we initially wrote the SSH blogpost in November of last year, we had a different top 10 versions of banners. At the time, we found that ssh-2.0-dropbear_2014.66 lead the top with approximately 7 million IP addresses running it, most of them were found on a specific AS (AS number 7922), which belongs to Comcast. This time, we only found 449,570 IPs with this banner. One possibility for this difference is that Comcast had cable modems for home customers with port 22 open to the internet and then deployed a patch which no longer made this port accessible by default.

Our next analysis was on key lengths which represent the size (in bits) of an SSH key. Depending on the family of the encryption algorithm, the key length defines the algorithm's security. In the initial study we had found the top key length was 1032 bits followed by 2048. On the current scan this is how our top 10 is looking like:

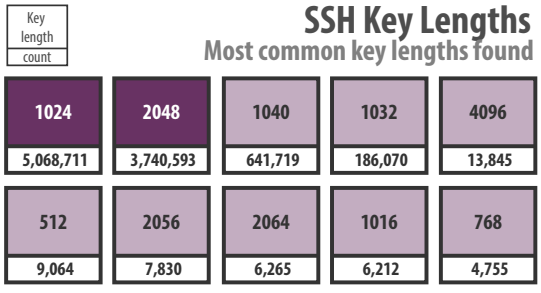


Figure 38: SSH key lengths

[12] "SSH - A brief analysis of the internet", retrieved from https://blog.binaryedge.io/2015/11/10/ssh/

Comparing these results with the ones in a forementioned blogpost, we can see that there was an enormous decrease in the amount of keys with 1032 bits, which we believe is related with the changes made by Comcast.

One huge change we also noticed was on the fingerprints. Before the Comcast change, the fingerprint e7:86:c7:22:b3:08:af:c7:11:fb:a5:ff:9a:ae:38:e4 used to be in the lead of the top 10 with 7.14 million of them being found, followed by d0:db:8a:cb:74:c8:37:e4:9e:71:fc:7a:eb:d6:40:81 in second place with 73.17 thousand. The new scan shows a much more balanced top 10 without the discrepancy caused by the repeated "e7:86..." key.

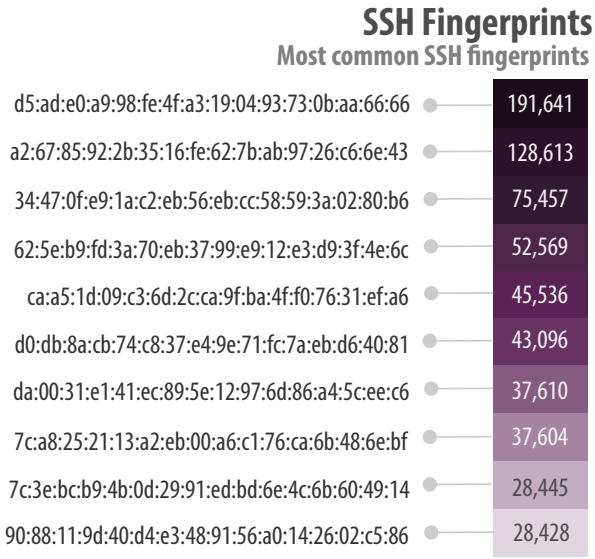


Figure 39: SSH fingerprints

Focusing now on the kex algorithms (key exchange algorithms), here is what we found.

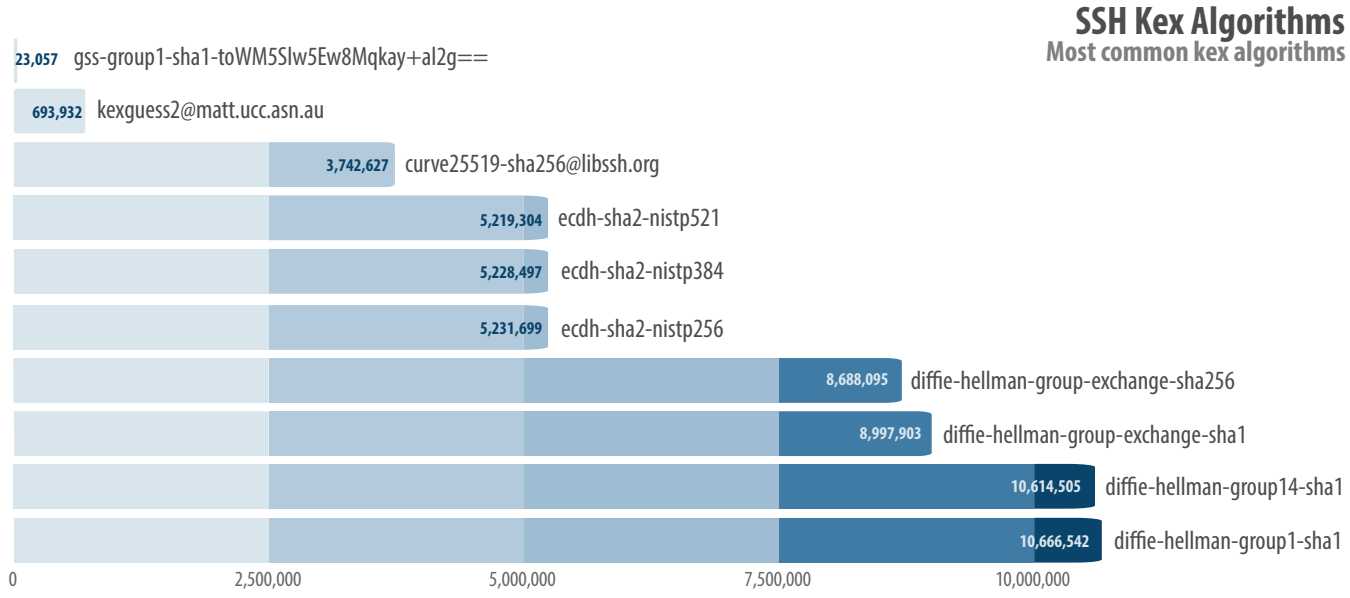


Figure 40: SSH kex algorithms

Both on the blogpost posted November of last year and on the data gathered for the purpose of this report, we found that the most common algorithm for key exchange is still "diffie-hellman-group1-sha1". Much to our dismay, and as we mentioned on that blogpost, we believe state sponsored attackers and government agencies can compromise and attack this algorithm.

MAC (Message Authentication Code) Algorithms are used to verify packet integrity. In the following image, you can see the most common MAC algorithms found.

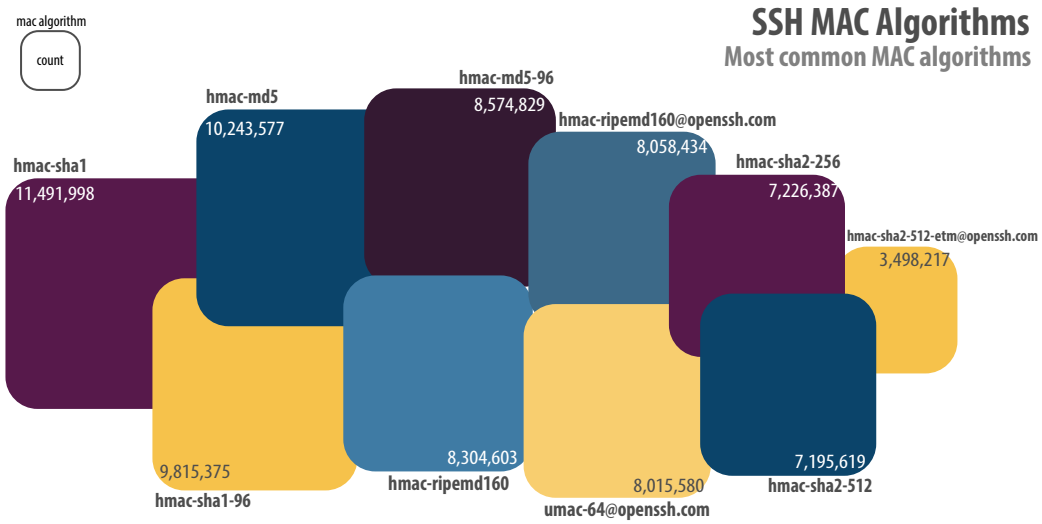


Figure 41: SSH MAC algorithms

Once again, as seen on our blogpost, the "red flag" pointed out at the time is still visible here: lots of SSH servers are still using SHA1 and MD5. Quoting our article: *"One big red flag here is the use of MD5 and SHA1. MD5 and SHA1 should never be used and yet there they are. Even though for HMAC you are most likely still safe , why bother with old, antiquated and bad crypto? Do not risk the security of your servers. Attacks only get better so, unless you have legacy maintenance issues, go for better crypto."*

Our final analysis was on Encryption Algorithms. Encryption algorithms, as the name indicates, are used to encrypt communication. Here is what we found:

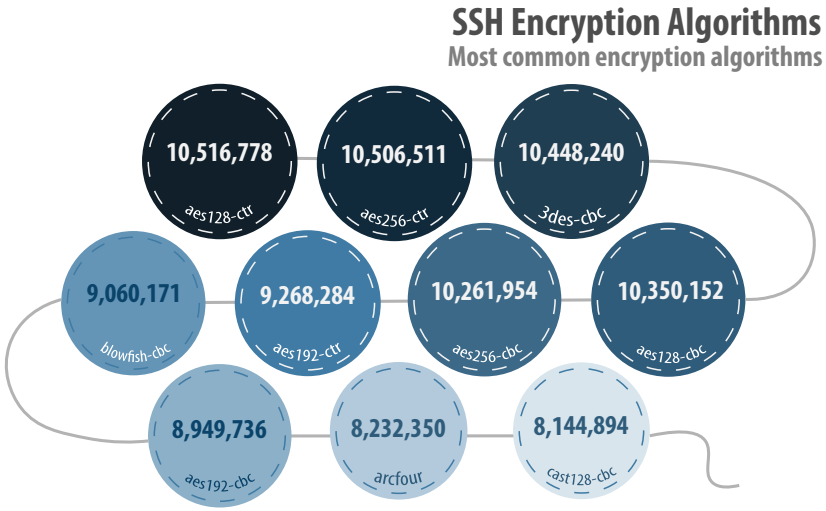


Figure 42: SSH encryption algorithms

Here we see that 3des-cbc is no longer on top but has been dethroned by aes-256-ctr and aes128-ctr.

With all the new information about the NSA being released, and the Shadowbrokers[13] showing that NSA's capabilities in terms of SIGINT and breaking encryption go way beyond what we originally thought, it's important that system administrators and devops keep their SSH configurations updated to the latest and most secure configurations.

All significant changes found during the writing of this report will be the subject of further analysis in the future.

[13] "The Shadow Brokers: Lifting the Shadows of the NSA's Equation Group?", retrieved from https://www.riskbasedsecurity.com/2016/08/the-shadow-brokers-lifting-the-shadows-of-the-nsas-equation-group/

MQTT

MQTT is a protocol invented in 1998 and was created by Andy Standford-Clark and Arlen Nipper. It was supposed to be used as a protocol to combat the unreliable satellite networks. It is a Client/Server pubsub messaging transport protocol. MQTT is a protocol regularly used in IoT, due to the small code footprint and lightweight design of the protocol.

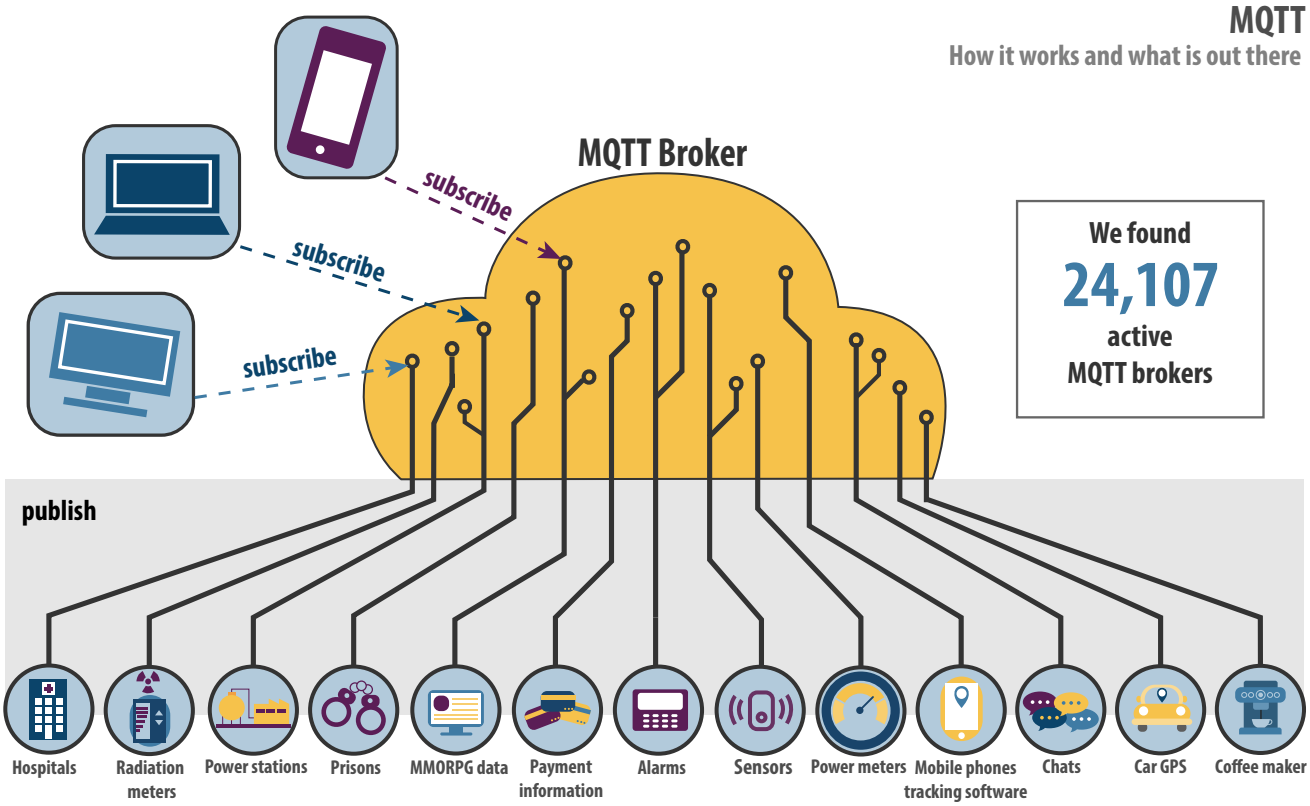


Figure 43: MQTT

The danger of having an MQTT broker open to the internet with no authentication is that anyone can connect to the broker and publish or subscribe to the different channels. If the information one is passing is critical, this presents multiple problems:

- If one is reading the data that comes from these channels on the MQTT Broker and saving it into a database, attacks such as XSS and even SQLi can be performed;
- An attacker can read that data and extract the classified information being transmitted;
- If one has sensors/receivers relying on that data, an attacker can send tainted data to those.

DID YOU KNOW?
During Defcon 24, Lucas Lundgren did a presentation on MQTT brokers he found with no authentication or encryption. Upon looking at those brokers, he found Hospitals, Prisons, Alarms, Cars, Mobile Phones tracking software, Radiation meters, Power meters, MMORPG data, amongst others. BinaryEdge's scans showed the same results.

Conclusion

By analyzing the exposure of these services and going into detail about certain features, we were able to get a global overview about the current state of the security of the internet.

Observing the data there are interesting facts that give us some positive outcomes from a security perspective such as:

- More SSH servers than Telnet servers;
- The gap between HTTP and HTTPS services on the main ports is much smaller than it was years ago.

These two points however are completely shadowed by the negatives:

- Multiple organizations are leaking many terabytes of data due to misconfigured database services;
- Multiple devices get replicated by providers and therefore get same keys and certificates - this was further explored by SEC Consult and you can read about it in their blog[14];
- An alarming number of critical systems are exposed at multiple levels, either by being directly exposed to the internet, by using messaging systems that have no security (MQTT) or by exposing their controlling computers to the internet with no password (VNC or X11).

If the evolution of infosec, IoT and technology as a whole keeps following the same path as it has now, a lot more problems will appear. With the introduction of IPv6, a lot more devices will be exposed and, with the users of these IoT devices not having any type of knowledge about security, it falls on the vendors to start focusing on making sure that the devices they are building are secure or, at least, have an easy way to apply a secure configuration.

Besides IoT, organizations in general need to evolve when it comes to data security. An user can try to have the most secure password in the world or try to follow all the basic security measures but, if the organization that owns and develops the web app where he is submitting all his personal information doesn't really care about security, it will all be for nothing.

Further studies will explore in more detail specific types of devices and protocols, in order to provide updated details and numbers on the protocols and services discussed on this report.

[14] "House of Keys: Industry-Wide HTTPS Certificate and SSH Key Reuse Endangers Millions of Devices Worldwide", retrieved from http://blog.sec-consult.com/2015/11/house-of-keys-industry-wide-https.html