
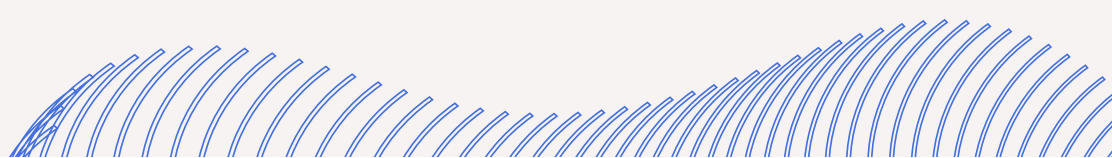


2023

IMPACT ANALYSIS OF MALWARE ON OPERATING SYSTEMS

TABLE OF CONTENTS

- 
- 01** EXECUTIVE SUMMARY
 - 02** GRAPHICAL SUMMARY
 - 03** PURPOSE OF THE REPORT
 - 04** PRELIMINARY CONCEPTS
 - 05** DEVELOPMENT OF THE ANALYSIS
 - 06** RESULTS
 - 07** ANNEX I: TESTING LABORATORY
 - 08** ANNEX II: SAMPLE ANALYSIS
TOOLS AND METHODOLOGIES
- 

EXECUTIVE SUMMARY

Today we live in a society deeply interwoven with the Internet, where it is easy to be exposed to the threat of malicious software, or malware. These programs infiltrate our computers through various means (from email attachments to vulnerabilities in outdated software) and cause numerous problems that directly impact our daily lives. These include identity theft, loss of information, and compromise of banking credentials, among others. Consequently, it is crucial to understand how different operating systems respond to such threats.

The purpose of this study is to examine the impact of these malicious programs on the most widely used operating systems: Windows, MacOS, Linux, and ChromeOS. Based on the infection rate calculated in this study, and the number of samples submitted to VirusTotal for each operating system during 2022, we have estimated the percentage of malicious files that could potentially compromise the security of these operating systems, despite their built-in security measures, as follows:

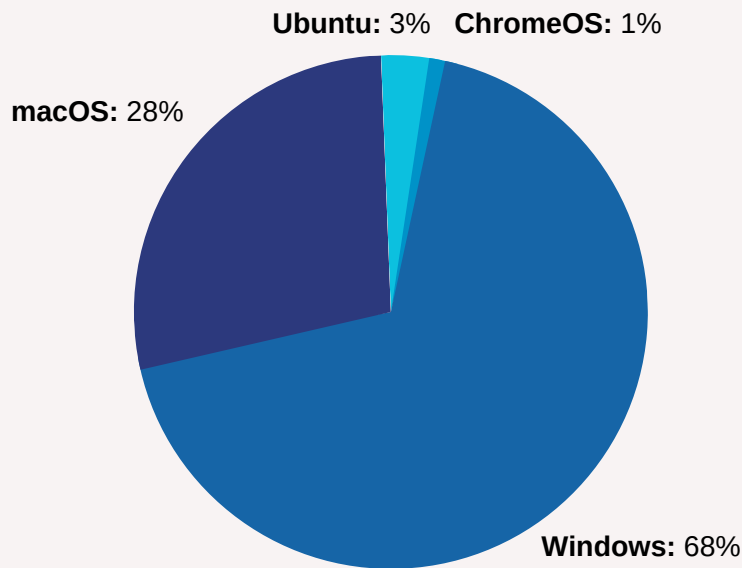
- Executable files:
 - Windows 74.5%
 - Linux 23.8%
 - MacOS: 1.5%
 - ChromeOS: 0%
- Executables and documents:
 - Windows 40.5%
 - Linux 21.6%
 - MacOS: 37.7%
 - ChromeOS: 0%

The main findings of the study are the following:

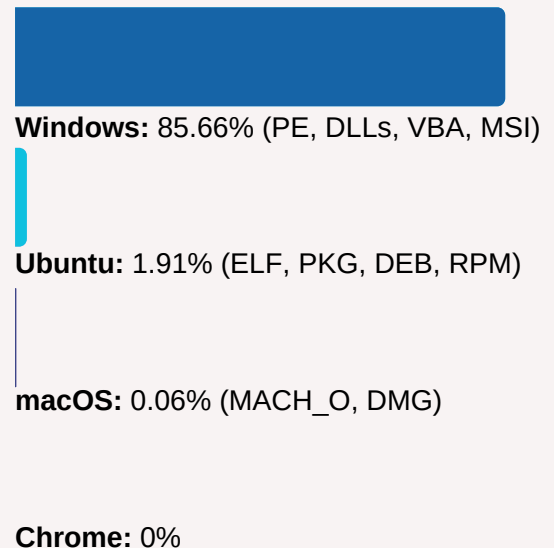
- Among all the tested operating systems, ChromeOS was the only one that successfully blocked all threats. Even using specific malicious apps from PlayStore, the malware resistance rate was 100% for all samples used.
- Microsoft Windows was found to be the preferred operating system for attackers, accounting for over 70% of the malicious files received by VirusTotal in 2022. According to data from this website, for every 44 Windows samples, one was uploaded for Linux, and for every 34,000 Windows samples, one was uploaded for MacOS.
- After ChromeOS, Linux is the operating system that performed best against malicious documents.

GRAPHICAL SUMMARY

CONTEXT

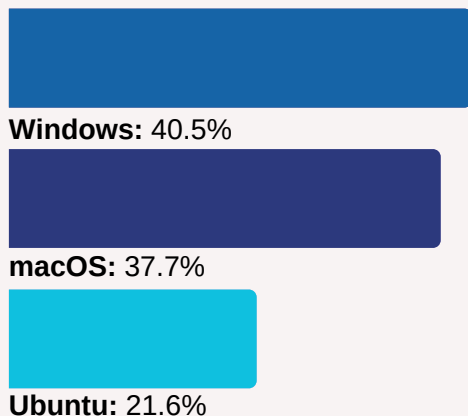


Reported unique vulnerabilities (CVEs) of major operating systems in 2022



Distribution by platform of malware submitted to VirusTotal in 2022

ANALYSIS RESULTS



Chrome: 0%

Estimation of the risk of infection by executables and office documents, considering the total number of malware samples per OS

0%

Ubuntu,
ChromeOS

Ratio of analyzed office documents (doc, pdf, excel...) that succeed in infecting Ubuntu and ChromeOS

1. PURPOSE OF THE REPORT

The purpose of this report is to analyze how today's most widely used operating systems (OS) can be affected by malicious software (malware). To this end, we conducted a comparative study, evaluating the security of default installations of Windows, MacOS, Ubuntu and ChromeOS by exposing them to various malware samples. The results obtained enable us to gauge the extent to which the choice of operating system can influence the risk of malware infection.

For this study, we analyzed malware samples uploaded to VirusTotal within a specific timeframe, without considering individual malware families or specific malware attributes. The idea was to recreate a situation as real as possible, exposing the operating systems to malware received by VirusTotal during that particular timeframe. Furthermore, we analyzed the results by considering the distribution of malicious samples submitted to VirusTotal, allowing us to estimate the genuine risk of infection for each operating system.

The various samples were tested on operating systems installed with default configurations, including both the configuration of the operating system itself and its security mechanisms.

2. PRELIMINARY CONCEPTS

Exposure to computer threats is a reality in today's digital landscape. Consequently, it is crucial to understand which operating systems are more secure and which are potentially more vulnerable to malware threats.

Malicious software, or malware, is software designed to infiltrate and, in many cases, damage a computer system without the knowledge or consent of its users. There are various types of malware, each with its own characteristics and methods of infection. Some of the most common types of malware are:

- **Viruses:** These are malware that replicate and spread through files and programs. Once executed, they can harm the system or alter files.
- **Worms:** This type of malware replicates itself and propagates through networks and emails.
- **Trojans:** These programs masquerade as legitimate software but are designed to infiltrate and potentially harm a system. Once installed, they can open backdoors, enabling unauthorized system access.
- **Phishing:** This type of malware is typically transmitted via email or SMS, tricking the user into visiting fraudulent websites that mimic legitimate ones.

- **Other:** Often, office documents (e.g., PDF, DOC, ODT, XLS files) may contain malicious code that executes when the file is opened, thereby infecting the system. Macros, which are scripts that automate tasks, are commonly used to execute this malicious code. It is crucial to exercise caution when opening files with macros, ensuring they originate from a trusted and legitimate source.

The primary methods of malware infection and propagation include:

- **Social engineering:** This technique involves the use of persuasion and deception techniques. A person is influenced to perform an action that results in system infection, such as clicking on a malicious link or downloading a dubious file.
- **Exploits:** These are software codes or commands utilized by attackers to infiltrate a system. Exploits are commonly found in malware samples and are used to exploit known software vulnerabilities, enabling the execution of a malicious payload.
- **Spread via email:** Many malware types spread via email, either as attachments or links to malicious websites.
- **Network propagation:** Certain malware types spread through networks, either via devices connected to the same network or through the internet.

Once malware is installed on a system, it can employ various techniques to ensure its presence and continuous operation, even after a system reboot. This process is referred to as **persistence**. Some persistence techniques include modifying the system configuration or installing services or scheduled tasks.

Before proceeding further, it is important to clarify the concept of **vulnerability**. One primary reason malware can impact an operating system or software application is due to flaws in its design, implementation, or maintenance. Examples of such flaws include programming errors, misconfiguration, lack of updates, and faulty design. Given the complexity of software creation, the existence of such flaws, and the weaknesses they cause (vulnerabilities) are virtually unavoidable. Those vulnerabilities can then be exploited by an attacker using the previously mentioned exploits.

In fact, one aspect to consider when studying the resilience of various operating systems is the number of reported vulnerabilities over the course of a year. The number of vulnerabilities can be extracted from the MITRE database, a non-profit organization whose goals include identifying, defining, and cataloguing publicly disclosed cybersecurity vulnerabilities. For each reported vulnerability, a CVE (Common Vulnerabilities and Exposures) is assigned. CVEs are unique identifiers given to a security vulnerability to track and document its impact, facilitating the communication and resolution of security issues.

Using the previous year's vulnerabilities as a reference, we derive the graph depicted in **Figure 1**.

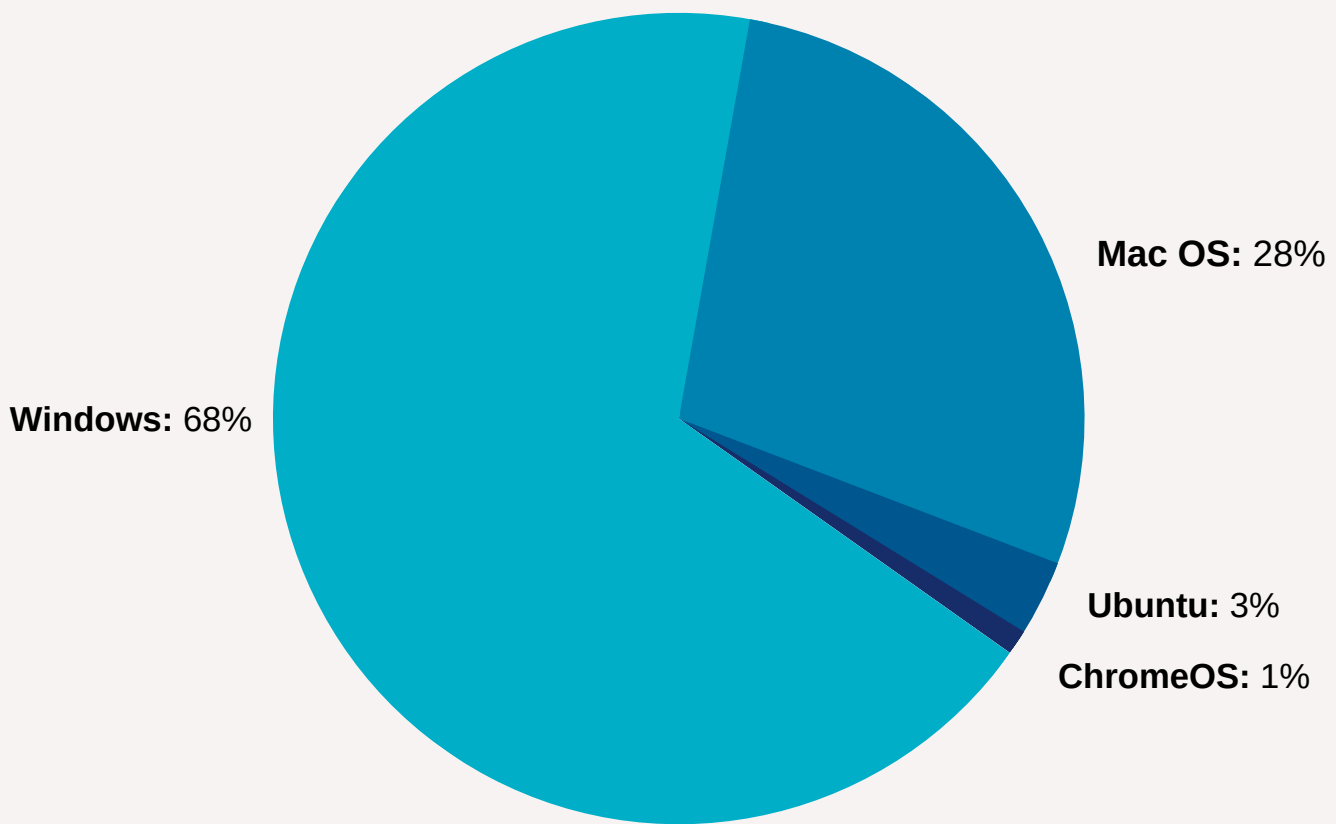


FIGURE 1. Reported CVEs of major operating systems in 2022

3. DEVELOPMENT OF THE ANALYSIS

To assess the impact of malware on each operating system (OS), we utilized a test laboratory composed of five physical computers. Each computer has a specific type of operating system installed, including Windows, macOS, Ubuntu, and ChromeOS. Details of the test lab can be found in **Annex I**.

To analyze the malware samples, we decided to divide them into several batches. That is, for each operating system samples were downloaded at an instant in time ("download day") without any prior filtering. We did not consider specific families or characteristics of the samples, only the submission date. The goal was to simulate a real-world scenario where the malware being spread at a given time is analyzed. Furthermore, given the complexity of analyzing a large quantity of malware in a single day, the samples downloaded at that instant were run over a specific period, referred to as the "analysis timeframe". This information is presented in **Table 1**.

Operating System	Unloading day	Start of the analysis	Completion of the analysis	Analysis timeframe
Windows	9 November	9 November	14 November	6 days
Ubuntu	15 November	15 November	18 November	4 days
macOS	15 November	22 November	25 November	4 days

TABLE 1. *Timeframe of the OS-specific file samples*

It is worth noting that due to the study's focus on evaluating novel samples within a given timeframe, no specific sample was received for ChromeOS within the analysis timeframe.

Another notable aspect is the number of samples used during the analysis. The distribution of samples for each operating system used in our dataset does not reflect the distribution of samples found in the VirusTotal database. This is illustrated in **Figure 2** and **Figure 3**, where the sample distribution in our analysis was chosen to facilitate comparison between the various operating systems.

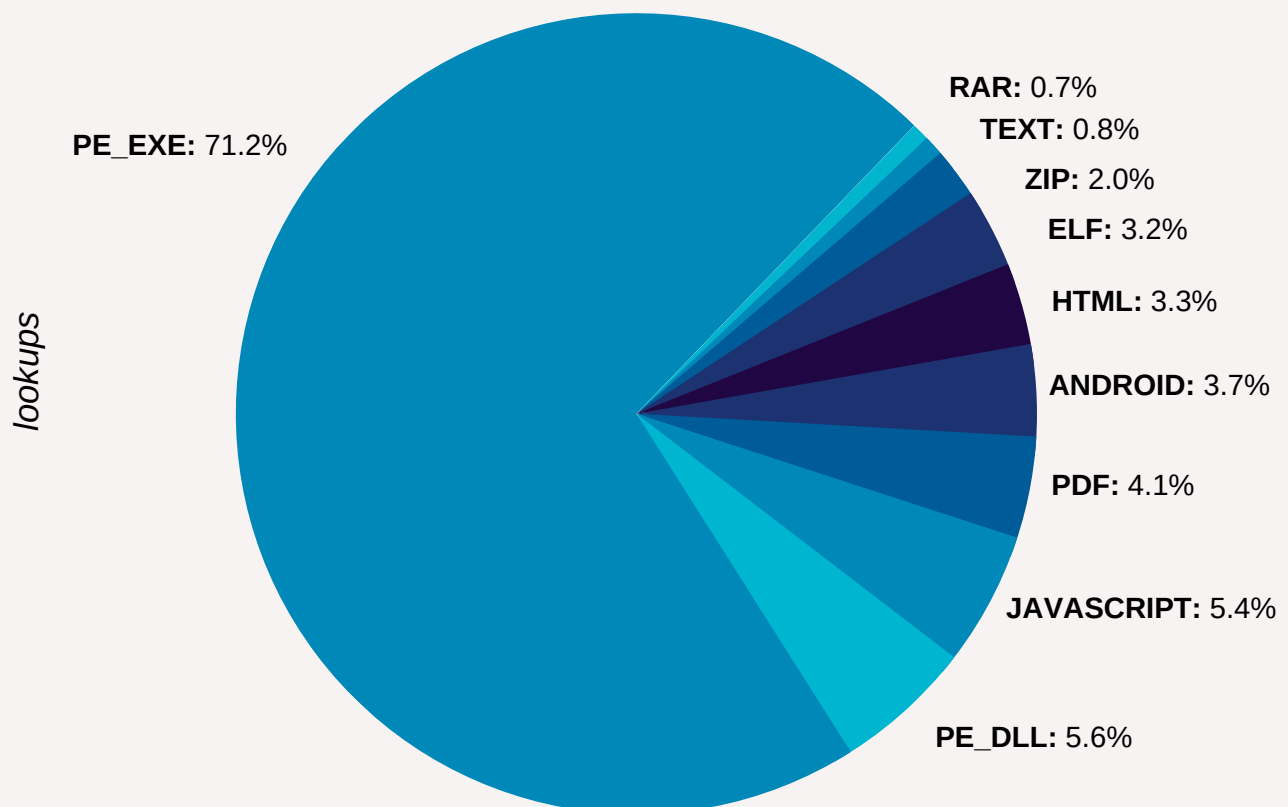


FIGURE 2. *Top 10 malicious file formats uploaded to VirusTotal in 2022*

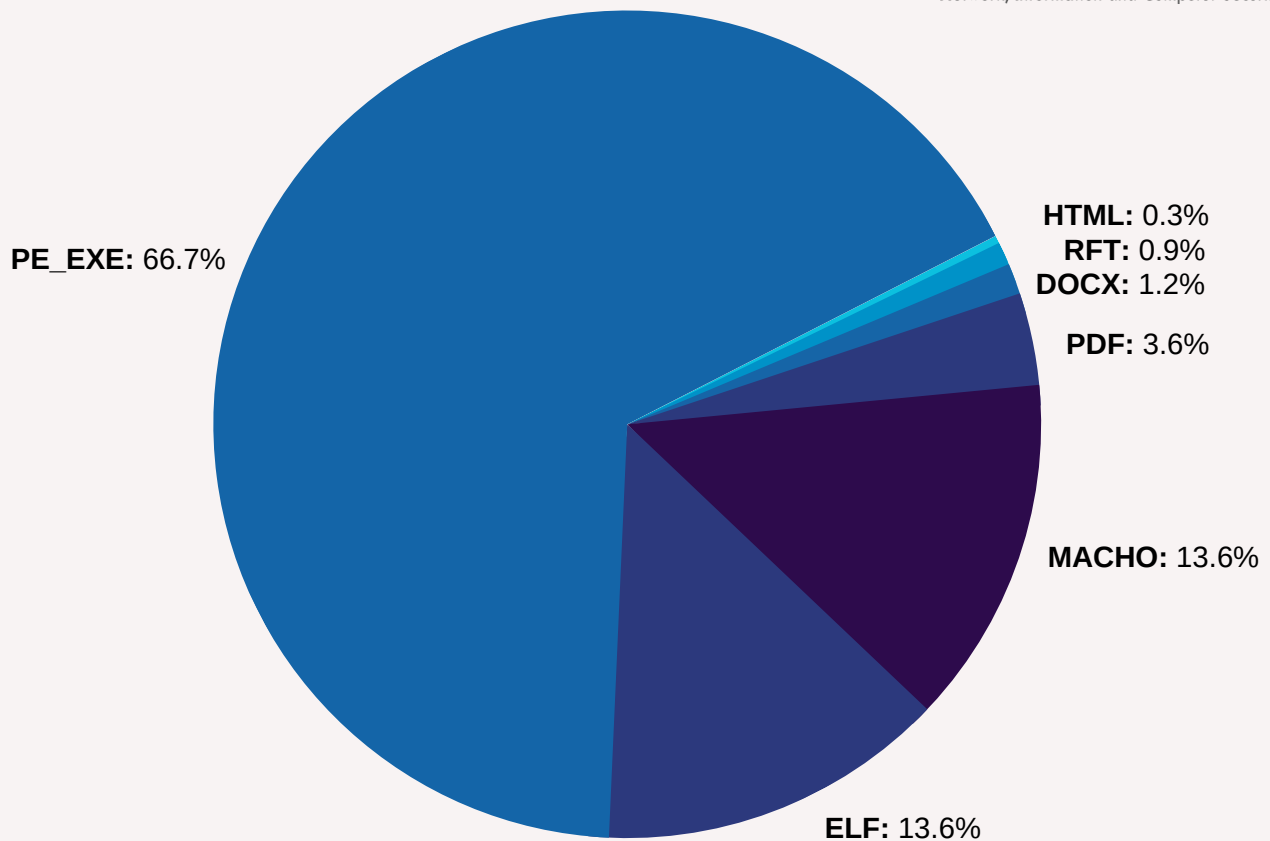


FIGURE 3. *Distribution of samples in our dataset*

Other aspects related to the selection of malware samples are as follows:

- Samples for macOS had to be downloaded prior to the start of the analysis, due to the shortage of samples within the timeframe used.
- Currently, there are not many ChromeOS-specific samples reported in the VirusTotal database. Furthermore, among the results found we observed that they were reported at very disparate time instants, making it challenging to match ChromeOS samples with those of other operating systems which do have files from a more recent timeframe. To overcome this situation, these samples, which were identified as malicious by independent developers on Twitter on the same day as the analysis, were extracted from new apps installed from Play Store.
- As for office documents (doc, pdf, excel...), since most are applicable to all operating systems, all samples were downloaded on the same day (9 November). This approach allowed us to obtain realistic results on how the same sample impacts different operating systems.

Finally, the methodology used to run the malware samples during the study is specified in **Annex II** of this report, detailing both the general methodology and the specific methodology for each operating system.

4. RESULTS

The overall infection data is shown in **Table 2**, representing the percentage of malware detected (and blocked) by each operating system. **Table 3** shows the results obtained when only considering executable files. Finally, **Table 4** shows the detection rates of malicious documents for each operating system.

The nomenclature used in these tables is as follows:

- **Malware resistance:** The capacity of an operating system to prevent malware infection.
- **Samples detected as harmful:** The operating system's built-in mechanisms (e.g., MS Defender in Windows, PlayProtect in Chrome OS) that have identified the samples as harmful.
- **Persistence:** The sample has executed various tasks to survive operating system restarts (e.g. registry modification).
- **Spread:** The sample has infected other files or computers, either locally or at the network level.
- **Execution of commands on the system:** The sample has succeeded in executing its payload, compromising the integrity and security of the installation.
- **Permission to run external applications:** The sample required interaction with the operating system user (e.g., SmartScreen in Windows, activation of macro functionality in office programs) to attempt to perform its malicious task.

Operating System	Resistance against malware	Samples detected as harmful	Persistence			Propagation	Execution of commands in the system	Permission to run external applications
			Modify OS configuration	Create tasks at regular time intervals	Execution of binary file (DLL, ELF...)			
Windows	94.54 %	93.1 %	0 %	3.18 %	2.72 %	0 %	5.45 %	3.63 %
Ubuntu	80 %	0 %	0 %	0 %	15.55 %	0 %	20 %	4.44 %
macOS	37.77 %	0 %	22.22 %	0 %	0 %	0 %	62.22 %	26.66 %
ChromeOS	100 %	39.39 %	0 %	0 %	0 %	0 %	0 %	0 %

TABLE 2: Overall performance results of the samples (specific files and office documents)

Operating System	Resistance against malware	Sample detected as harmful	Persistence			Propagation	Execution of commands in the system	Permission to run external applications
			Modify OS configuration	Create tasks at regular time intervals	Execution of binary file (DLL, ELF...)			
Windows	97.5 %	97.5 %	0 %	1 %	0.5 %	0 %	2.5 %	1 %
Ubuntu	64 %	0 %	0 %	0 %	28 %	0 %	36 %	0 %
macOS	28 %	0 %	0 %	0 %	0 %	0 %	72 %	0 %
ChromeOS	100 %	100 %	0 %	0 %	0 %	0 %	0 %	0 %

TABLE 3. Sample (executable) files specific to each O.S.

Operating System	Resistance against malware	Sample detected as harmful	Persistence			Propagation	Execution of commands in the system	Permission to run external applications
			Modify OS configuration	Create tasks at regular time intervals	Execution of binary file (DLL, ELF...)			
Windows	65 %	50 %	0 %	25 %	25 %	0 %	35 %	30 %
Ubuntu	100 %	0 %	0 %	0 %	0 %	0 %	0 %	10 %
macOS	50 %	0 %	50 %	0 %	0 %	0 %	50 %	60 %
ChromeOS	100 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %

TABLE 4. Samples of office document files (doc, pdf, excel...)

While these results demonstrate the likelihood of successfully detecting and/or resisting attacks for the analyzed sample set, they are not representative of the actual exposure risks over a longer timeframe, where the number of samples for each operating system is several times higher than that used in the study. Indeed, as we can see in **Figure 2**, which shows the Top 10 malicious file formats uploaded to VirusTotal, the percentage of malware samples for certain operating systems is considerably higher compared to others – thus the risk of exposure could be much higher. Therefore, it is crucial to analyze the true risk of infection for each operating system, taking both factors into account.

For this reason, we estimated the risk of infection using malware data submitted to VirusTotal in 2022, grouping the most common file formats by operating system. To do this, we calculated how many files could compromise security by applying the ratios in **Tables 2 and 3** to the total malware samples for each one. The distribution of malware by platform or type of infection is as follows:

- **Windows:** PE files, DLLs, VBA and MSI files account for 85.66% of malicious files.
- **Linux:** ELF, PKG, DEB and RPM account for 1.91% of malicious files.
- **MacOS:** MACH_O and DMG account for 0.06% of malicious files.
- **Documents:** PDF, XLS, XLSX, DOC, DOCX and RTF account for 12.36% of malicious files.

Thus, we can see that although Windows has an excellent capacity to defend itself against malware, the sheer volume of existing malware means that the real risks of exposure are higher: 74.5% (2.5% of 85.66% of files) malware capable of executing commands on Windows vs. 23.8% (36% of the 1.91% of files) malware capable of executing commands on Ubuntu.

Table 5 shows the percentages for each operating system considering the previous discussion; that is, of all the files calculated with the ratios in **Tables 2 and 3**, how many belong to each operating system.

Operating System	Type	Infection Rate	Percentage of the total number of files
Windows	Executables	2.5%	74.5%
	Executables and Documents	5.45%	40.5%
Ubuntu	Executables	36%	23.8%
	Executables and Documents	20%	21.6%
macOS	Executables	72%	1.5%
	Executables and Documents	40%	37.7%
ChromeOS	Executables	0%	0%
	Executables and Documents	0%	0%

TABLE 5. *Estimated actual risk for each operating system*

Following these results obtained from the various rounds of analysis, we can see that the only operating system that has not been affected by a malicious sample – be it a native operating system executable or an office file – is Chrome OS for the timeframe studied.

Other conclusions drawn from the analysis are as follows:

1. The number of samples affecting Windows is extremely high compared to other operating systems. Conversely, the number of samples affecting Chrome OS is the lowest.
2. Having an integrated malware detection mechanism is highly beneficial: the ratio of samples analyzed in the study that manage to infect the operating system is 0% for Chrome OS and only 5.45% for Windows –compared to 20% for Ubuntu and 62.22% for Mac OS.
3. In the particular case of MacOS, the executable samples in the study were granted permission to run, as otherwise they could not have been executed given they were not considered trustworthy due to the files' signatures.
4. Windows and MacOS operating systems are susceptible to infection by office files: in the samples analyzed, 35% affected Windows, and 50% affected Mac OS. This ratio drops to 0% for Linux and ChromeOS.
5. It is worth noting that on ChromeOS, the malicious office documents were not detected by the system as malicious, although they were in turn unable to execute their payload, thus failing to compromise the OS.
6. It was observed that the vast majority of the office files analyzed correspond to phishing campaigns distributed via e-mails or by accessing malicious websites.

ANNEX I: TESTING LABORATORY

All computers were up to date on the same date as the test run, with default security settings (e.g. built-in antivirus protection, installation of programs from trusted sources, secure boot), and without additional security applications. The versions of each operating system used, as well as the additional software installed, are the following:

- **Windows 10 v22H2**
 - Additional software installed:
 - Microsoft Office 365 Suite
 - Adobe Acrobat Reader
- **Windows 11 v22H2**
 - Additional software installed:
 - Microsoft Office 365 Suite
 - Adobe Acrobat Reader
- **macOS Monterey v. 12.6.1**
 - Additional software installed:
 - Microsoft Office 365 Suite
 - Adobe Acrobat Reader
- **Ubuntu v. 22.04.1 LTS**
 - Additional software installed:
 - LibreOffice Suite
- **ChromeOS v. 105.0.5195.134**
 - Additional software installed: None (office suites are installed by default)

ANNEX II: SAMPLE ANALYSIS TOOLS AND METHODOLOGIES

II.1 General Methodologies

The tests were executed according to the following steps:

1. Each operating system and additional software were installed on the corresponding hardware.
2. Each operating system was configured to access the Internet using an isolated and monitored network to prevent the spread of malware. This network made use of a Wi-Fi hotspot created from a SIM card with mobile data access.
3. Once the initial configuration of the operating system was completed, it was cloned using the "Clonezilla" tool.
 - a. In the case of the Chromebook, a tool from the Chrome Web Store called "Chromebook Recovery Tool" was used for recovery.
4. Specific samples (malware), extracted from the VirusTotal database, were executed on the operating systems.
 - a. The specific steps to carry out this execution and to test the reaction of both the operating system and the malware are detailed in the following sections ("Specific methodologies").

II.2 Specific methodologies

II.2.1 Windows

Tools

The tools used to perform the analysis of the malware samples are the following:

- *Process monitoring:*
 - **System Monitor (Sysmon):** Provides detailed information (logs) on process creation, network connections and changes in file creation time
 - **Autoruns:** Displays the applications that are configured to auto-start, as well as the registry locations and system files available for auto-start configuration
 - **PsService:** Provides the status, configuration and dependencies of a service

- *Network traffic analysis:*

- **Wireshark:** Tool to analyze all existing network traffic in real time. It has access to all the information about the packets that are being exchanged.
- **Snort:** IDS (Intrusion Detection System) that allows to analyzing traffic through rules that are configured to detect intrusions or attacks. Another great advantage of Snort is that it can analyze traffic from a Wireshark capture.

- *Analysis of logs/records:*

- **Sigma rules:** They are to logs what Snort rules are to network traffic. In other words, they detect potential anomalies within the logs.

In addition, the **Splunk** tool, a SIEM (Security Information & Event Management) solution, was used to monitor and analyze all events more quickly.

Methodology

To execute each malware sample on Windows, the following procedure was carried out:

1. Wireshark was launched on the corresponding interface to capture the network traffic.
2. The Sysmon logs were emptied, so that they start capturing from the execution of the malware sample.
3. The malware sample was run.
4. After a given period of time, the Sysmon logs and the traffic capture generated by Wireshark were saved.
5. Autoruns and PsService were run to check persistence.
6. Traffic capture was analyzed with the Snort IDS, and Sysmon logs were analyzed with the Splunk tool.
7. The computer was recovered with CloneZilla, so that the initial configuration of the operating system was restored.

II.2.2 macOS

Tools

In addition to the Wireshark and Snort tools, the specific tools used to analyze the malware samples come from the "Objective-see" suite, and are as follows:

- **Do Not Disturb (DND):** This tool performs continuous monitoring of system events.
- **KnockKnock:** This tool allows users to identify software installed on their computer that is not legitimate, including malware that achieves persistence – i.e. starts when the operating system boots up.
- **TaskExplorer:** This tool shows all the processes running on the computer. Additionally, its integration with VirusTotal makes it possible to display the number of detections of a given process.
- **BlockBlock:** This tool monitors the most common locations used by malware to gain persistence, alerting its users whenever these locations are modified.
- **RansomWhere?:** This tool continuously monitors the file system for the creation of encrypted files by "suspicious" processes.

Methodology

The following procedure was performed to execute each malware sample on macOS:

1. Wireshark was launched on the corresponding interface to capture the network traffic.
2. Processes were controlled by running TaskExplorer.
3. The malware sample was run.
4. After a given period of time, the process was checked to see whether cron jobs, which are used to maintain the persistence of the sample, had been created.
5. Traffic capture was analyzed with the Snort IDS.
6. DND records were checked to see if files had been created/modified.
7. The KnockKnock and BlockBlock tools were run to see if the sample had achieved persistence.
8. The computer was recovered with CloneZilla in order to restore the operating system's initial configuration.

II.2.3 Ubuntu

Tools

In addition to the Wireshark and Snort tools, the specific tools used to analyze the malware samples are the following:

- **Procmon Script:** Script that monitors the processes running on the system.
- **Persistence Script:** Script that notifies if any of the most common paths to achieve persistence in Linux environments are modified.
- **Ubuntu Logs:** "Logs" of the system itself that store information about events occurring on the system.

Methodology

The following procedure was performed to executed each malware sample on Ubuntu:

1. Wireshark was launched on the corresponding interface to capture the network traffic.
2. Procmon and Persistence scripts were executed.
3. The malware sample was run.
4. After a given period of time, the process was checked to see whether cron jobs, which are used to maintain the persistence of the sample, had been created.
5. Traffic capture was analyzed with the Snort IDS.
6. The Ubuntu logs were scanned, to see what changes had been made to the system.
7. The computer was recovered with CloneZilla in order to restore the operating system's initial configuration.

II.2.4 ChromeOS

Tools

The tools used to perform the analysis are those that are integrated into the Chrome browser itself. To access them, type **chrome://chrome-urls/#internals** in the search bar. A list of different addresses will then be displayed, each with a specific functionality. Among them, we can highlight the following:

- **chrome://internals/session-service:** Displays information about the current or previous sessions.
- **chrome://net-export:** Generates a JSON file with all the network information collected by the Chrome browser. This file can later be viewed with the **netlog-viewer** tool, also implemented by Chrome.
- **chrome://safe-browsing/:** Displays information related to safe browsing. For example, information about download requests, responses, whether a URL has been checked, about phishing requests, etc.
- **chrome://device-log/:** This category allows access to all the logs of the device, which can be accessed by category.

Other tools that were used during the analyses were **Task Manager**, which displays information about the processes currently running on the system, and Play Store's **Packet Capture** application for capturing existing network traffic.

Methodology

The following procedure was performed to execute each malware sample on the Chromebook:

1. ChromeOS Task Manager and Packet Capture were opened.
2. We started capturing the traffic `chrome://net-export` and with the Packet Capture application.
3. We actively checked `chrome://safe-browsing/` requests made from the browser.
4. We stopped the traffic capture and analyzed it in `netlog-viewer`.
5. We looked at `chrome://internals/session-service` to obtain information about crashes, unintentional shutdowns, and other actions in the browser.



 www.nics.uma.es

 [@nics_lab](https://twitter.com/nics_lab)

 [NICS Lab](https://www.linkedin.com/company/nics-lab)

 contact@nics.uma.es



Ada Byron Research Building
Extension of the Teatinos Campus
University of Malaga
C/ Arquitecto Francisco Peñalosa, 18
29071 - Malaga (Spain)