Some Analysis of Common Vulnerabilities and Exposures (CVE) Data from the National Vulnerability Database (NVD)

Jillian Glyder
jag9603@uncw.edu

Andrew Kyle Threatt
akt7028@uncw.edu

Randy Franks
rtf4785@uncw.edu

Lance Adams
gla7029@uncw.edu

Geoff Stoker
stokerg@uncw.edu
University of North Carolina Wilmington
Wilmington, NC 28412 USA

Abstract

Vulnerability trends can be very useful for informing the cyber risk management process. The objective of this paper is to analyze trends in Common Vulnerabilities and Exposures (CVE) data feeds from 2003 to 2021 using Common Vulnerability Scoring System (CVSS) version 2.0 scores. Data for 147,547 CVEs through June 2021 were downloaded from the National Vulnerability Database (NVD), parsed via Python-based text mining, and analyzed to identify various trends. Findings include a sharp increase in vulnerability integration, a slight decline in average base score of vulnerabilities over time, and the prevalence of exploits surrounding the Android operating system and man-in-the-middle attacks. This information may aid security measurement and management by helping information technology and security professionals form a security strategy based upon conclusions drawn from the analysis.

Keywords: CVE, NVD, vulnerability

1. INTRODUCTION

The National Institute of Standards and Technology (NIST) defines vulnerability as "weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source." (NIST, n.d.-a). Vulnerabilities are the catalyst of what may be perceived as a constant arms race between attackers that seek to exploit systems via these vulnerabilities and security professionals who wish to mitigate system compromise; thus, having knowledge of vulnerability trends provides information technology professionals with a better means of managing risk. The relative importance of security intelligence continues to increase alongside an ever-growing presence of

vulnerabilities. As we finish writing this paper, the number of vulnerabilities tracked by NIST in the National Vulnerability Database (NVD) and assigned a Common Vulnerabilities and Exposures Identifier (CVE ID) now exceeds 150,000 and is rapidly approaching 160,000. As tracked by NIST, the number of CVEs more than doubled from 2016 to 2017 and have continued to grow (figure 1). As this paper is being written during 2021, we do not yet know what the total number of vulnerabilities will be for this year. Figure 1 reflects the 2021 value as of the time of our analysis – we note that it is on pace to match the 2020 total number of CVEs.

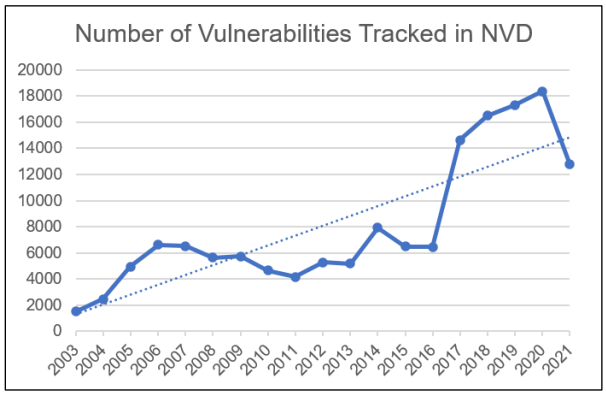

Figure 1 – Number of Vulnerabilities Tracked by NIST in the NVD (NIST, n.d.-b)

This rapid growth denotes the necessity to convert security data into actionable information for utilization but begs the question of how to convert such vast quantities of data. An avenue of resolution may be found in evaluating trends in enumerated CVE vulnerabilities over time. The objective of this paper is to analyze trends in CVE data feeds via Common Vulnerability Scoring System (CVSS) version 2.0 metrics from the year 2003 to present, with a significant subset of data focusing on analysis of major operating systems and products from 2010 to 2020.

2. BACKGROUND

Created in 1999 by MITRE Corporation and sponsored by the Department of Homeland Security, CVEs are a free, publicly available source for those interested in viewing different vulnerabilities, security tools, and services along with their effects (Armerding, 2017). CVEs provide a standard identifier for vulnerabilities and exposures that have been discovered, as well as what actions were taken towards mitigation. The overall goal is to share information with the public so that they may access information on cyber threats and possibly use the CVSS when prioritizing remediation.

In the spirit of why CVEs were created, we analyzed data using keywords to surmise the upward trend in vulnerabilities and exposures. CVEs are put into the database once a vendor or researcher requests it. They could also not make that request or put many of their fixes under one CVE. Because of this, the number of CVEs does not indicate quality of security. Understanding that the number of CVEs for a given product does not tell you if the product is more secure or not, we used keywords to establish trends in the data and attempt to come up with reasons as to why these trends are occurring.

3. METHODOLOGY

In order to analyze these trends, we downloaded NIST-provided JavaScript Object Notation (JSON) formatted files from the NVD site (NIST, n.d.-c) and created various Python scripts to iterate through the CVE descriptions. Any programming language with the ability to parse JSON-formatted data sets may be utilized, but Python was used in this situation based upon prior experience. For each operating system, product, or exploit that we were interested in, there could be multiple keywords. For example, the keywords for the man-in-the-middle exploit used the keywords: "Man in the middle", "man in the middle", "Man-in-the-Middle", "man-in-the-middle", and "MITM". If the description contained any of the keywords, then that CVE was counted once and included in the total. We could have used the Common Platform Enumeration (CPE) list, but it only shows platforms and some products, and does not include exploits. We determined that we could get a bigger picture by analyzing more data in the descriptions.

4. NUMBER OF VULNERABILITIES

Figure 1 presents yearly changes in the number of vulnerabilities tracked in the NVD. The trendline reflects a major increase in the appearance of vulnerabilities since 2016. This growth is likely attributable, in part, to an increase in the global number of assets connected to the internet and the increase in the number of CVE Numbering Authorities (CNAs), of which there are now more than 150 worldwide (Redscan, 2021, p. 6, para. 1). After a small declining trend starting around 2007, vulnerability integration skyrocketed in 2017, with the largest single year increase being from 2016 to 2017 at 127%. Since that notable jump, year-over-year change is trending about 5% in

recent years, excluding 2021 for which we currently have only partial data.

5. BASE SCORE OF VULNERABILITIES

The CVSS assigns scores of 0-10 to CVEs with CVSSv2 in use since 2007 and CVSSv3 since 2015. Figure 2 presents annual change in the average CVSSv2 base score of vulnerabilities across NVD data feeds. The base score has remained relatively stable, with a standard deviation of .34. The number has stayed around 6, starting at 6.2 in 2003 and ending at 5.5 in 2021, displaying a small decline over time.

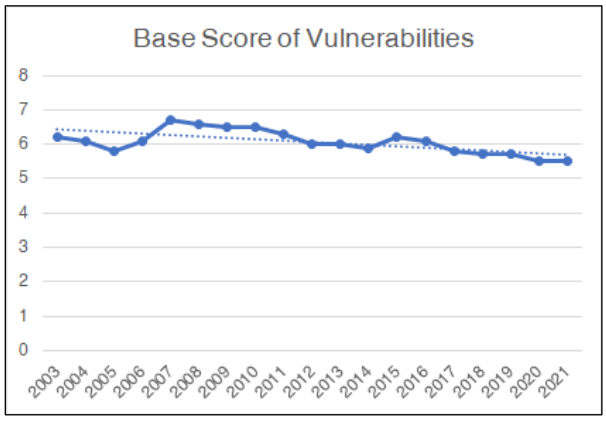

Figure 2 – Annual average CVSSv2 base score for vulnerabilities

6. SEVERITY LEVEL

In addition to the use of a numerical score 0-10, the CVSS further assigns CVEs to a severity group. CVSSv2 uses three groups, while CVSSv3 introduced a fourth severity category as can be seen in figure 3.

CVSS score	CVSSv2	CVSSv3
9.0 – 10.0	High	Critical
7.0 – 8.9		High
4.0 – 6.9	Medium	Medium
0.1 – 3.9	Low	Low
0.0		None

Figure 3 – numeric ranges, categories, and colors of the CVSS versions 2 and 3.

Figure 4 presents the CVE population distribution of vulnerabilities annually by CVSSv2 severity levels. Collectively, the CVEs since 2003 are distributed among the severity levels as: 32% high, 58% medium, and 10% low. Two views of the same data are provided in figure 4 with the top view reflecting the actual numbers per year and the bottom view showing each year as a total of 100% and making it somewhat easier to compare distributions across years. Growth in medium severity vulnerabilities is rapid, with low severity vulnerabilities being slightly less so and high severity vulnerabilities fluctuating over the years. Medium severity vulnerabilities remain the highest sample.

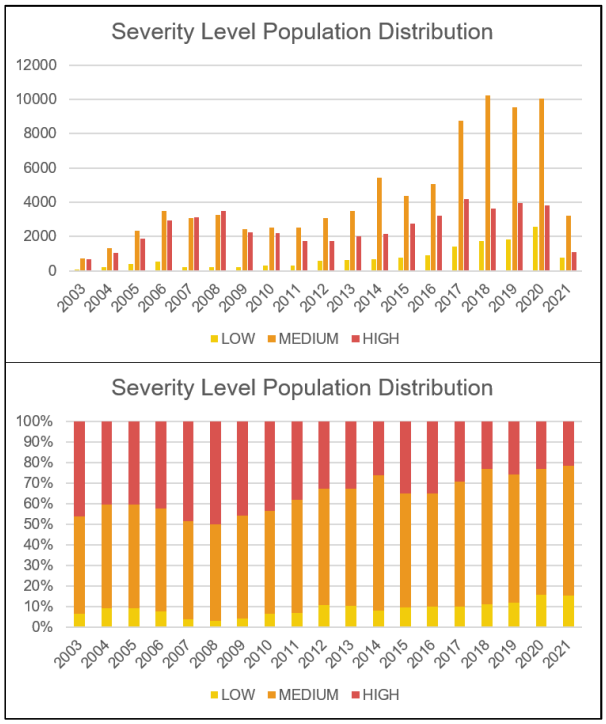

Figure 4 – two different views of the annual severity distribution of CVEs

7. ACCESS VECTOR

The access vector for each vulnerability – local, network, or adjacent – indicates how an attacker could exploit a given CVE. Local means that a CVE is not bound to the network stack, but that the attacker exploits the vulnerability by accessing the target system locally or relies on a user to perform the actions required for exploitation. Network indicates that a CVE is remotely exploitable and does not requires that an attacker gain direct local access. Adjacent means the attack comes across the network but is limited to a logically adjacent topology (Mell, Scarfone, & Romanosky, 2007).

Figure 5 depicts two views of the population distribution of vulnerabilities yearly by access vector. Of the total population across the years analyzed, 83% of vulnerabilities utilize a network access vector, 14% of vulnerabilities require local access, and 3% of vulnerabilities exploit an adjacent network access vector. One notable outlier in the data is an 831% increase in adjacent

network attacks from 2013 to 2014, an explosive spike relative to any other year or metric – we will discuss this more in the next section. Vulnerabilities with a network access vector have had the most dramatic increase in real numbers and remain the largest percent in every year examined.

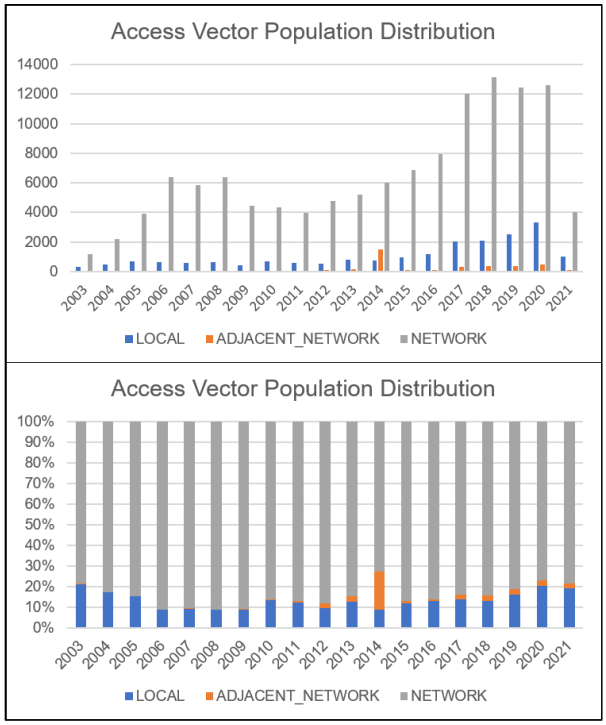

Figure 5 – two different views of the annual distribution of access vector for CVEs

OS Context of Adjacent Access Vector

We were curious about the spike in adjacent network exploits in 2014. We suspected some kind of man-in-the-middle (MITM) exploit and examined the description of each CVE from 2010-2020 that included one of the MITM terms (figure 6).

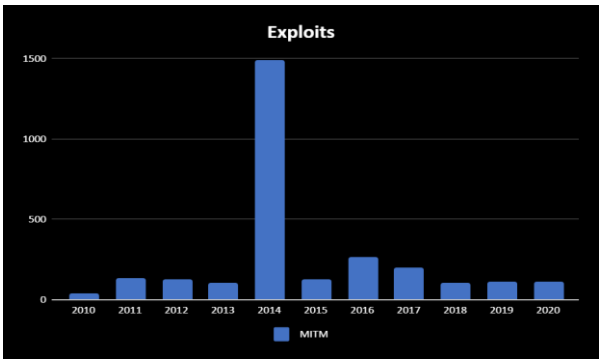

Figure 6 – number of man-in-the-middle attacks from 2010-2020

After seeing the descriptions, the reason became apparent. In that year, we found that 1399 Android apps were not verifying X.509 certificates from SSL servers.

Since they were not being verified, a MITM exploit could be executed to spoof a server and gain access to private information on the devices. For some reason, each app was listed individually and given a unique CVE number. This also explains why the Android operating system spiked in 2014 as well (figure 7).

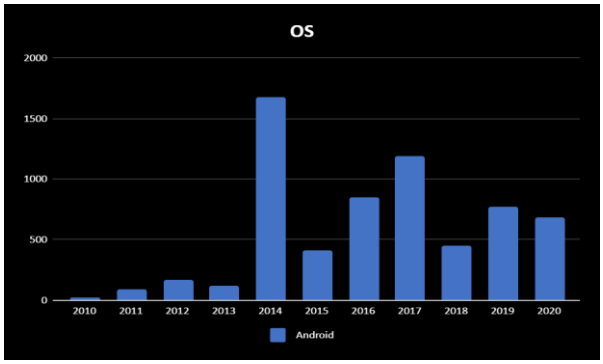

Figure 7 – number of Android-related CVEs 2010-2020

8. ACCESS COMPLEXITY

Access complexity is divided into three metrics: low, where specialized access conditions or circumstances do not exist; medium, where access conditions are somewhat specialized; and high, where specialized access conditions exist (Mell et al., 2007). From the total population of vulnerabilities examined, 57% are of low access complexity, 40% of medium access complexity, and only 3% are of high access complexity. Low and medium access complexity vulnerabilities have fluctuated over the years, with low complexity vulnerabilities having steady growth over medium complexity vulnerabilities in recent years. Figure 8 shows two views of the population distribution of vulnerability access complexity on an annual basis.

9. AUTHENTICATION

The CVE authentication metric is divided into three categories: none, where authentication is not required to access and exploit a vulnerability; single, where an instance of authentication is required for access and exploitation; and multiple, where an attacker must be authenticated multiple times (Mell et al., 2007). Figure 9 presents the population distribution of the authentication metric by year. Analyzing the number of vulnerabilities in this category, 84% of

vulnerabilities require no authentication, 15% require a single instance of authentication, and less than one percent require multiple instances of authentication. While the none and single subsets have historically fluctuated, they have resulted in steady growth in recent years, while the multiple subset has remained statistically, albeit not contextually, insignificant. Attacks not requiring authentication remain the largest sample.

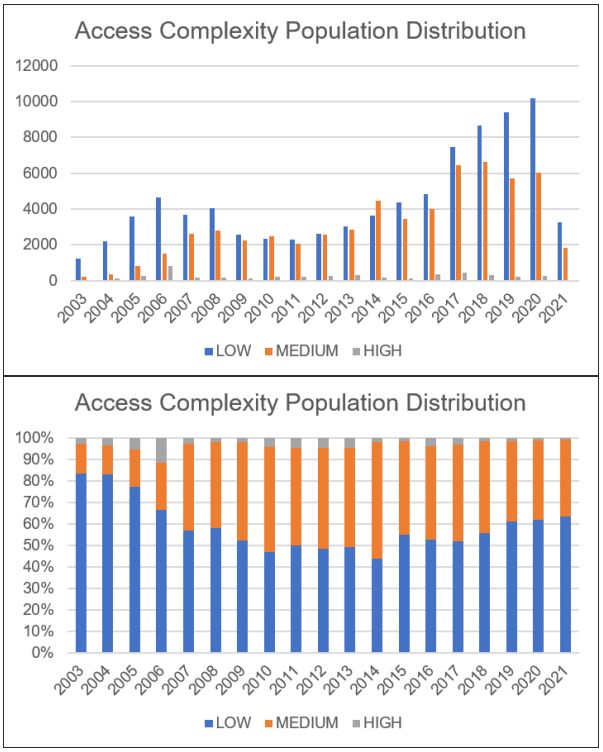

Figure 8 – annual distribution of low, medium, and high access complexity

10. EXAMPLE POTENTIAL CAUSES OF VULNERABILITIES

Simply looking at numbers is not sufficient to determine which systems are more secure, but they do provide a starting point for investigation. For example, there have been more CVEs for Android than Apple's iOS. The process of creating a CVE begins with the discovery of a potentially harmful vulnerability or exposure. Is this because Android is inherently more unsecure than iOS or are there more things to consider?

According statcounter GlobalStats (2021), Android as an OS constitutes ~72% of the world mobile operating system market share as opposed to iOS with ~27%. From 2010 to 2020 Android rose from about 5% market share to its current position while iOS has fluctuated between 30% and 20% (see figure 10). This information provides insight into why there are more CVEs on Android than on iOS. More potential targets use Android, so it is reasonable to expect there will also be more potential threat actors interested in Android.

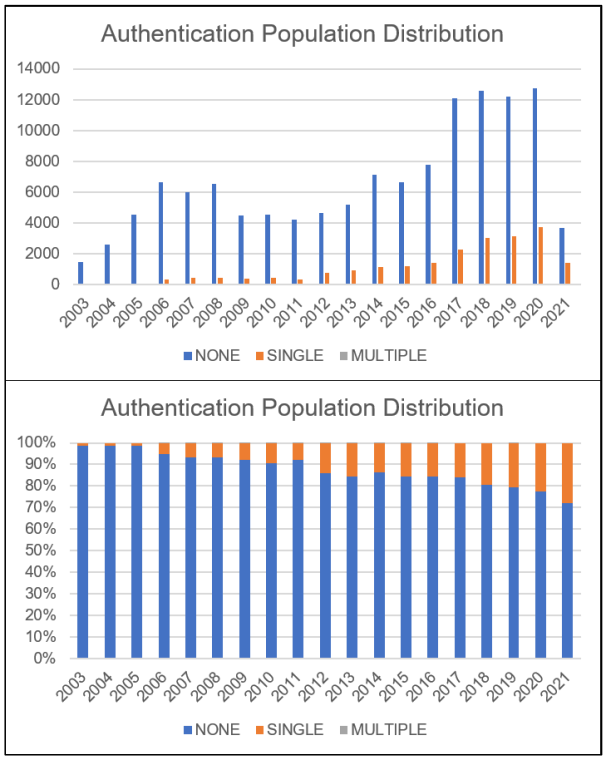

Figure 9 – annual distribution of vulnerability authentication

On top of the many users of the Android OS, there are also many different original equipment manufacturers (OEMs). A short list of the OEMs of Android (Figure 10) includes, but is certainly not limited to, Samsung, Nokia, Vivo, Motorola, LG, Asus, Sony, and Huawei. With each OEM, there will be additional "bloatware" added to the device which adds an extra layer of potential risk.

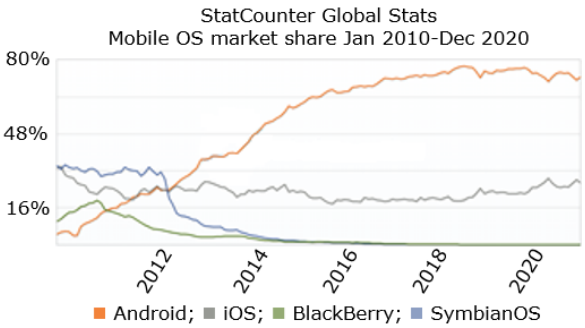

Figure 10 – mobile OS marketshare from 2010-2020; (StatCounter, 2021)

Apple, on the other hand, produces its own devices, preventing extra layers of risk. Because of the extra bloatware, patches and updates from certain OEMs can be sparse which gives threat actors more time to launch attacks. The Pixel line of Android phones is an exception among them due to its consistent updates and being made by Google, the developer of Android.

One of the primary weaknesses in any system is the user. Android is based on the foundation of the Linux kernel. From the stats we received from CVEs, Linux had an overall vulnerability increase of 6% from 2010 to 2020 which is fairly low for an operating system. Linux makes a secure kernel for a mobile device based on the numbers, though that may or may not be true depending on what is edited in the code to make installing applications easier. Although Android is fragmented there are many options for security, but only potentially 1% of cell phones carry any kind of security software (Doffman, 2021).

With Apple, they take charge of the security of your phone but with Android, the user has more of a responsibility in ensuring phone security.

> If you use an Android, then the onus is on you, the user, to secure your device. There are plenty of security platforms available from leading vendors. And they can wrapper the device. If you're an enterprise user, then your company can do the same for you. This overcomes the issues with Android's fragmented ecosystem, the lag in deploying security patches and general updates, the relative lack of security on the Play Store compared to Apple's equivalent (Doffman, 2021).

11. DISCUSSION AND CONCLUSIONS

In this paper, we analyzed 147,547 CVEs to discern useful and/or interesting trends from 2003 to 2021 utilizing CVSS v2.0 scores. Based on data aggregated, analyzed, and displayed via the above figures, the following conclusions may be drawn:

- Given the steadily increasing number of internet-connected assets and larger number of CNAs, the annual number of CVEs is likely to continue to increase. The prospect of seeing a downward trend similar to that from 2006-2011 seems remote.
- CVEs rated as medium severity will likely remain the most common, while low severity will likely remain the least common. With high severity vulnerabilities numbering around 4,000 annually for the past several years, this still poses a significant challenge for vulnerability analysis as more than 10 high severity vulns, on average, must be evaluated and actioned per calendar day.
- Vulnerabilities targeting a network access vector will very likely continue to vastly overshadow the populations of vulnerabilities targeting adjacent network and local access vectors in the near future.
- Low access complexity vulnerabilities appear likely to continue to remain the most common in the near future.
- Attacks requiring no authentication will presumably continue to out-populate attacks requiring authentication, though CVEs requiring authentication are on an upward trajectory. This could bode well for enterprises as multi-factor authentication becomes more commonplace and would presumably make it more difficult for these CVEs to actually be exploited.
- While CVE analysis is useful for cyber risk management, there are potential pitfalls. Two examples regarding Android were discovered during our analysis. First, the 2014 spike in Android-related MITM attacks appears to expose a potential flaw in the NVD for allowing over-reporting of certain kinds of vulnerabilities. Arguably, there was a single general problem with Android regarding certificates and not over a thousand as were reported for each individual application. Second, top-level examination of CVE data may lead to conclusions that are not warranted – for example, by inferring Android is a more insecure mobile OS simply because there are more CVEs reported. This analysis fails to consider second-order issues like the popularity of Android compared to iOS, the number of OEMs involved, and the relative freedom of use of the technology permitted by the manufacturer.

Finally, while analysis of NVD data feeds is certainly worthwhile, a major limitation is that it does not distinguish vulnerabilities exploited in the wild from those that have never been exploited. That being said, this information may still greatly aid security measurement and management by helping IT and security professionals form a security strategy based upon conclusions drawn from CVE analysis.

12. FUTURE WORK

There will always be analysis to do on CVE data including continuing to redo analysis done in the past. Some future work we are interested to

investigate includes discerning trends for vulnerabilities based on:

- the "lastmodifiedDate" and "description" metrics
- population distribution of respective impacts on CIA triad facets
- creation and analysis of "exploitabilityScore" and "impactScore" population distributions based upon the respective metrics
- analysis of vulnerability type population distributions
- all of the above comparing CVSSv2 and CVSSv3 since the introduction of v3 in 2015

13. REFERENCES

Armerding, Taylor. "What Is CVE, Its Definition and Purpose?" CSO Online, CSO, 10 July 2017, www.csoonline.com/article/3204884/what-is-cve-its-definition-and-purpose.html

Doffman, Zak. "No, Your iPhone Is Not More Secure Than Android, Warns Cyber Billionaire." Forbes, Forbes Magazine, 16 Mar. 2021, https://www.forbes.com/sites/zakdoffman/2021/03/16/iphone-12-pro-max-and-iphone-13-not-more-secure-than-google-and-samsung-android-warns-cyber-billionaire/?sh=85be53f23f84

Mell, P., Scarfone, K., & Romanosky, S. (2007, July). A Complete Guide to the Common Vulnerability Scoring System Version 2.0. NIST. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=51198

NIST. (n.d.-a). Computer Security Resource Center; Glossary. CSRC. https://csrc.nist.gov/glossary/term/vulnerability

NIST. (n.d.-b). National Vulnerability Database. https://nvd.nist.gov/vuln/search/statistics?form_type=Basic&results_type=statistics&search_type=all&isCpeNameSearch=false

NIST. (n.d.-c). National Vulnerability Database. NVD Data Feeds. https://nvd.nist.gov/vuln/data-feeds

Redscan. (2021). NIST security vulnerability trends in 2020: an analysis. Redscan. https://www.redscan.com/media/Redscan_NIST-Vulnerability-Analysis-2020_v1.0.pdf

StatCounter. (2021). Mobile Operating System Market Share Worldwide. https://gs.statcounter.com/os-market-share/mobile/worldwide