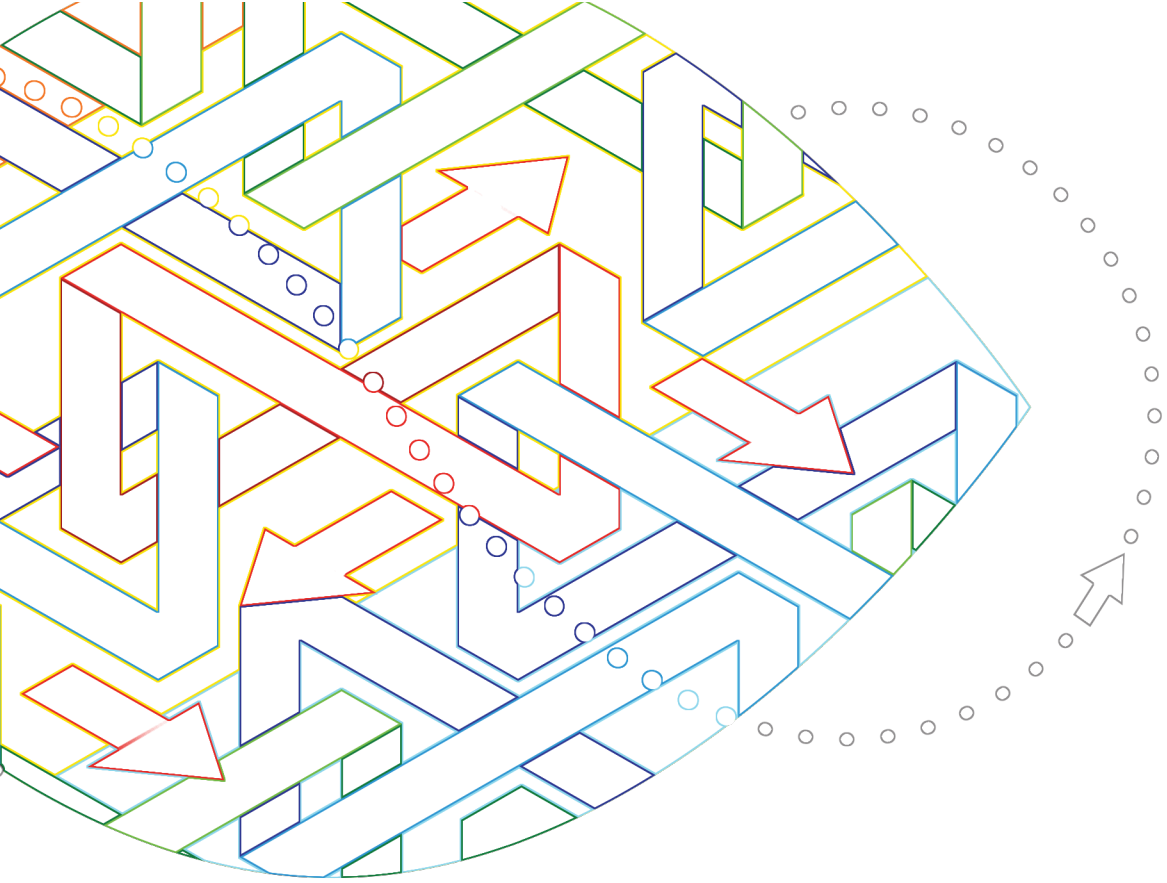
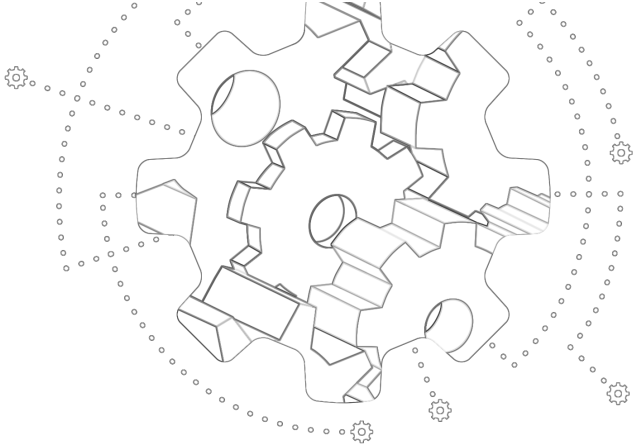


25 Years of Vulnerabilities: 1988-2012

RESEARCH REPORT

Yves Younan
Senior Research Engineer
Sourcefire Vulnerability Research Team (VRT™)

SOURCEfire®



Overview

With 25 years of vulnerability data now available, this report takes a historical look at vulnerabilities over the years. Some of the results were surprising, like the Linux kernel having the most CVE vulnerabilities of all other products, while others were less surprising, like Microsoft being the vendor with the most vulnerabilities, or that the buffer overflow is the most occurring vulnerability in the last quarter century.

Some of the results were surprising, like the Linux kernel having the most CVE vulnerabilities of all other products.

We leveraged two well-respected data sources for our research. First, our classifications of vulnerabilities are based on the **Common Vulnerabilities and Exposures (CVE)** [1] database which is used today as an international standard for vulnerability numbering or identification. The database provides 25 years of information on vulnerabilities to assess, spanning 1988 to current.

Next, we used information hosted in the **National Vulnerability Database (NVD)** [2] at the **National Institute of Standards and Technology (NIST)**. We did some normalization to the data with respect to vulnerability categorization to be able to provide more complete statistics. Additional details on the methodology used for modifying the NVD data is provided at the end of the report. Two important caveats: First, not every vulnerability is assigned a CVE, so those of course aren't counted here. Second, NVD also assigns a CVSS score of 10 when a vendor does not provide sufficient information to be able to assess the impact of the vulnerability[1].

Let's take a look at what our research unveiled so that we can leverage it to help us better protect enterprises today.

[1] NVD. NVD Common Vulnerability Scoring System Support v2 (Incomplete data). http://nvd.nist.gov/cvss.cfm

Vulnerability statistics

By year

Let's start by taking a quick look at the volume of vulnerabilities over the last 25 years, as shown in Figure 1 below.

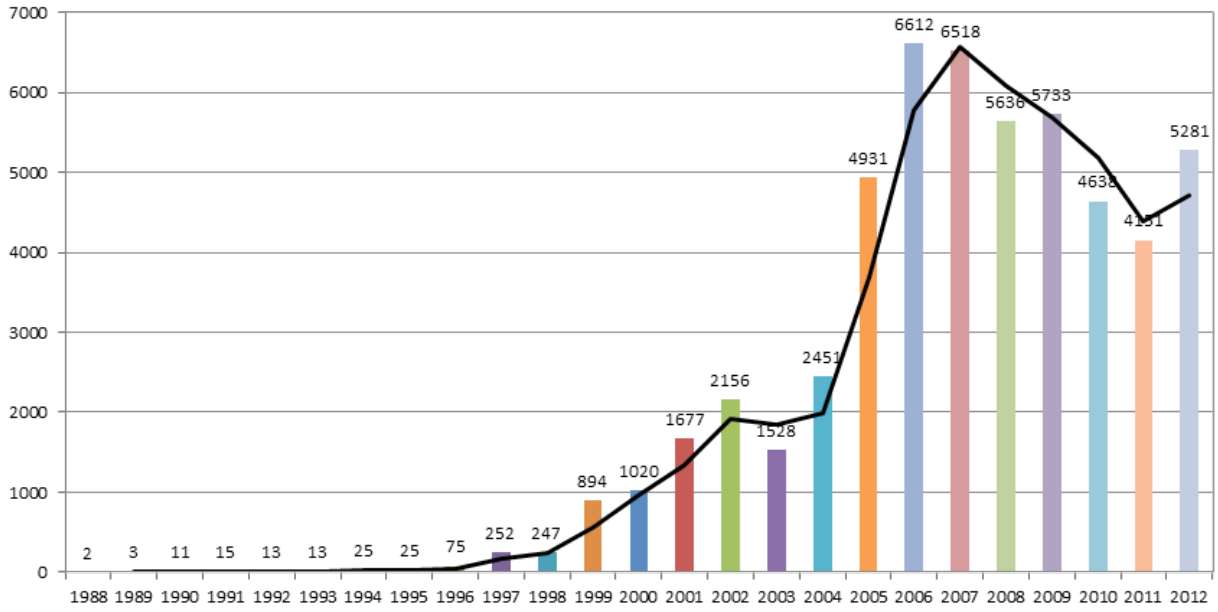


Figure 1. Vulnerabilities by year

This shows some interesting facts about vulnerabilities over the years. The number of discovered and reported vulnerabilities increased rapidly from 1988-2005, with only a small lull in 2003. We reached a peak in the number of vulnerabilities in 2006 and were on a steady decline after that, falling to a six-year low in 2011. However, last year, in 2012, the number of vulnerabilities shot up again to levels comparable to 2008-2009.

This begs the question: Is 2012 an outlier and will the trend of declining vulnerability reports continue in 2013, or will it reverse itself and are we headed back towards 2005-2006 levels?

Maybe the rest of the data can point to the reason for this upswing in vulnerabilities. To dig a little deeper, let's take a look at vulnerabilities with a Common Vulnerability Scoring System (CVSS) score of 7 or higher, which signifies "High" severity.

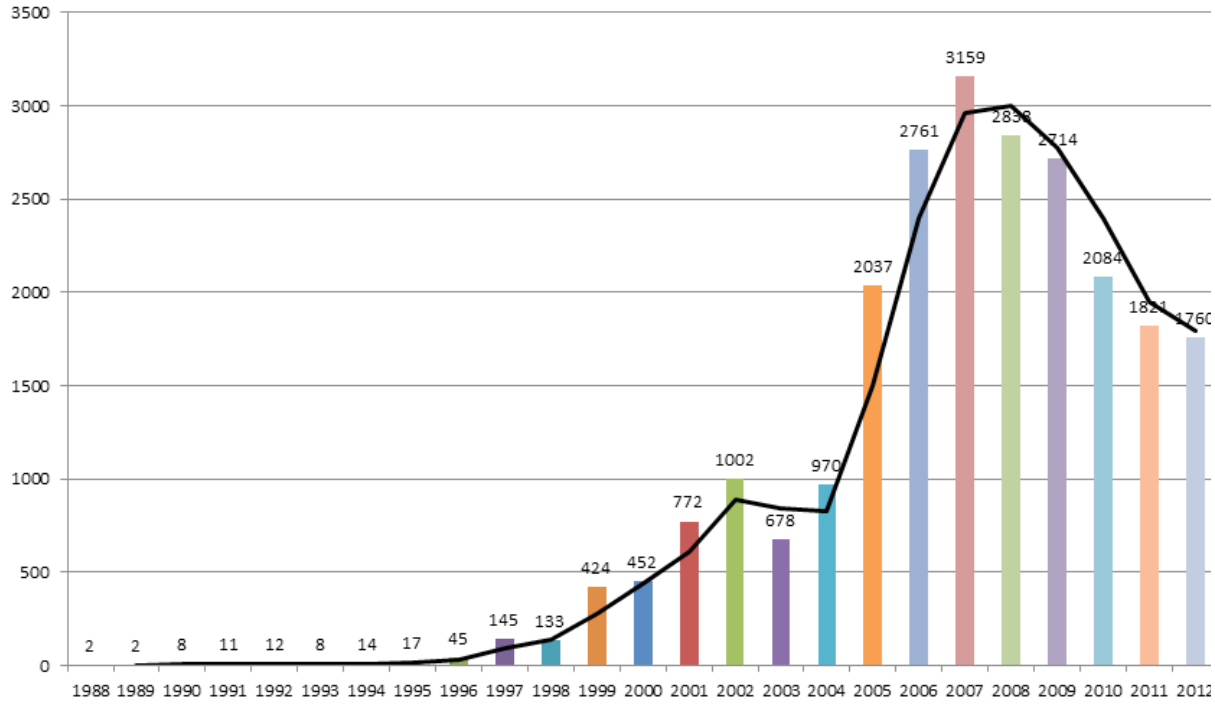


Figure 2. High severity vulnerabilities by year (CVSS >=7)

Figure 2 shows a clear trend: the number of vulnerabilities with a high severity rating increased significantly until 2007 when they reached a high of 3159. Since then they have tapered off and last year fell below the 2005 levels, even though more vulnerabilities were discovered in 2012 than in 2005.

Here's another way to look at that data:

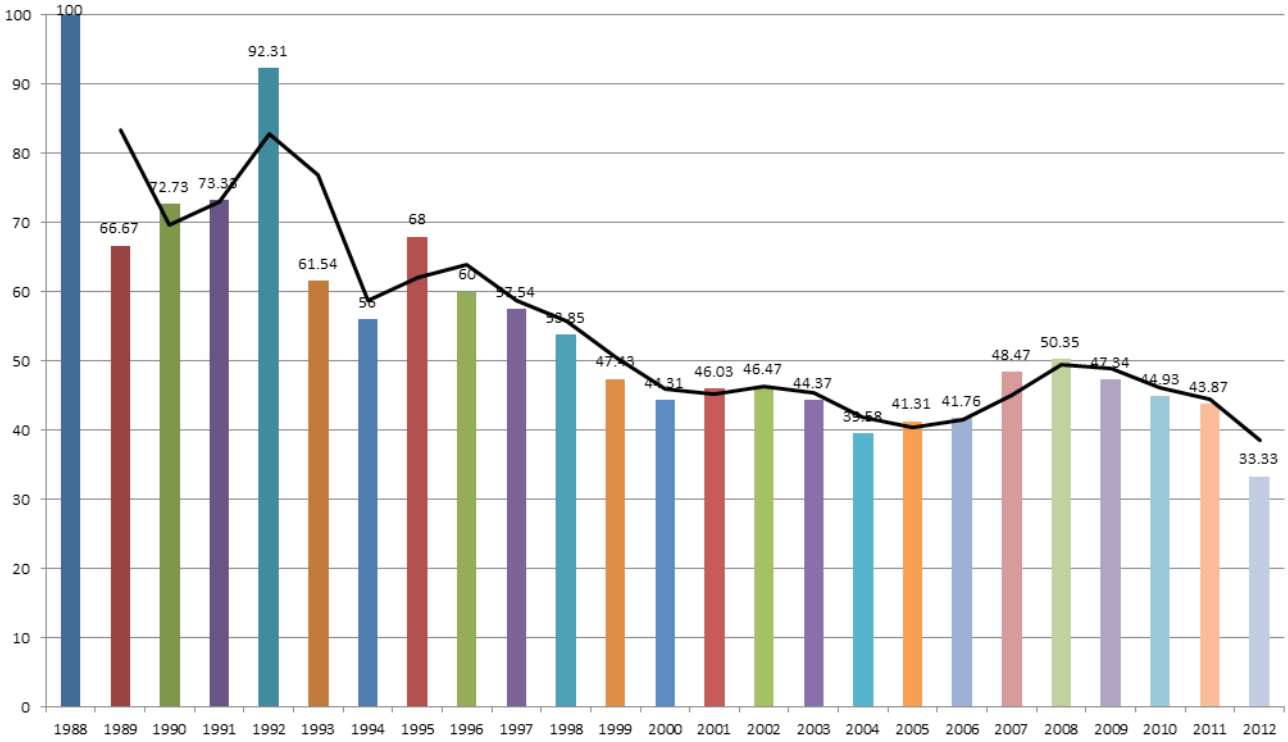


Figure 3. High severity vulnerabilities by year as a percentage of total vulnerabilities

For the first time ever, in 2012 high severity vulnerabilities only make up 33% of the vulnerabilities assigned CVEs. This is a significant improvement over earlier years; in the previous decade high severity vulnerabilities averaged 45%.

If we select only vulnerabilities with a CVSS of 10 (the highest score, considered a critical vulnerability) and depict both the actual number and the volume as a percentage of total vulnerabilities per year, we see the following.

For the first time ever, in 2012 high severity vulnerabilities only make up 33% of the vulnerabilities assigned CVEs.

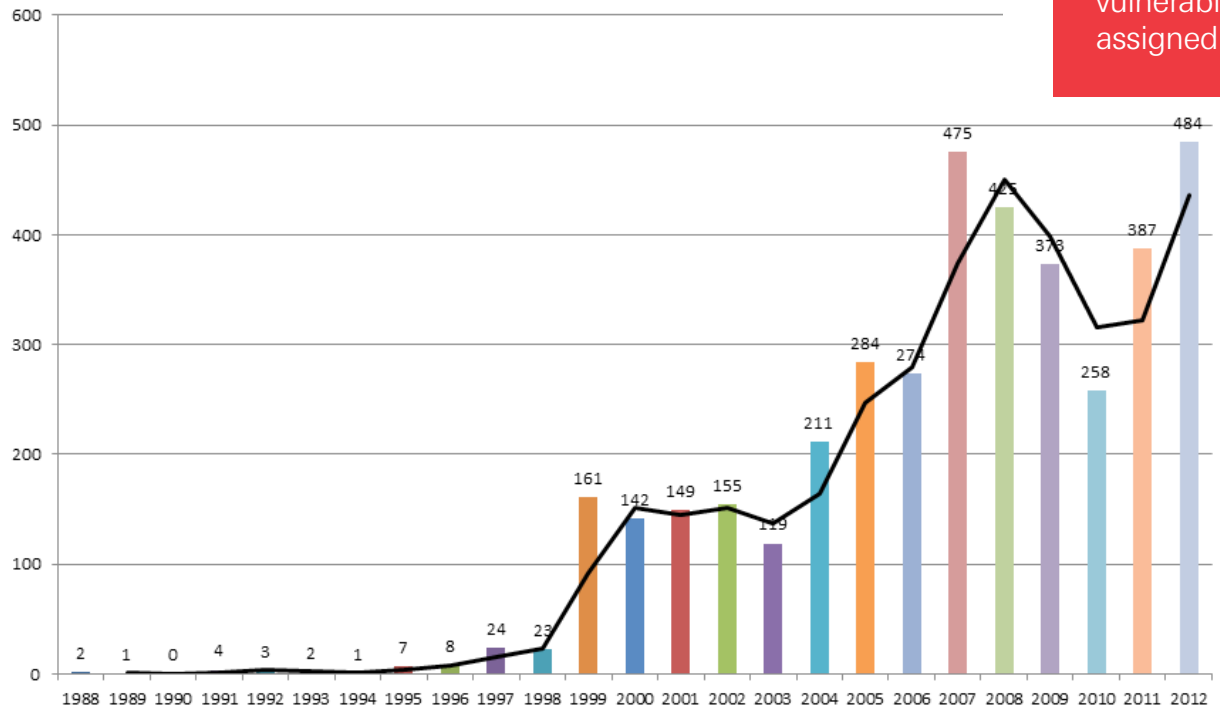


Figure 4. Critical severity vulnerabilities by year (CVSS = 10)

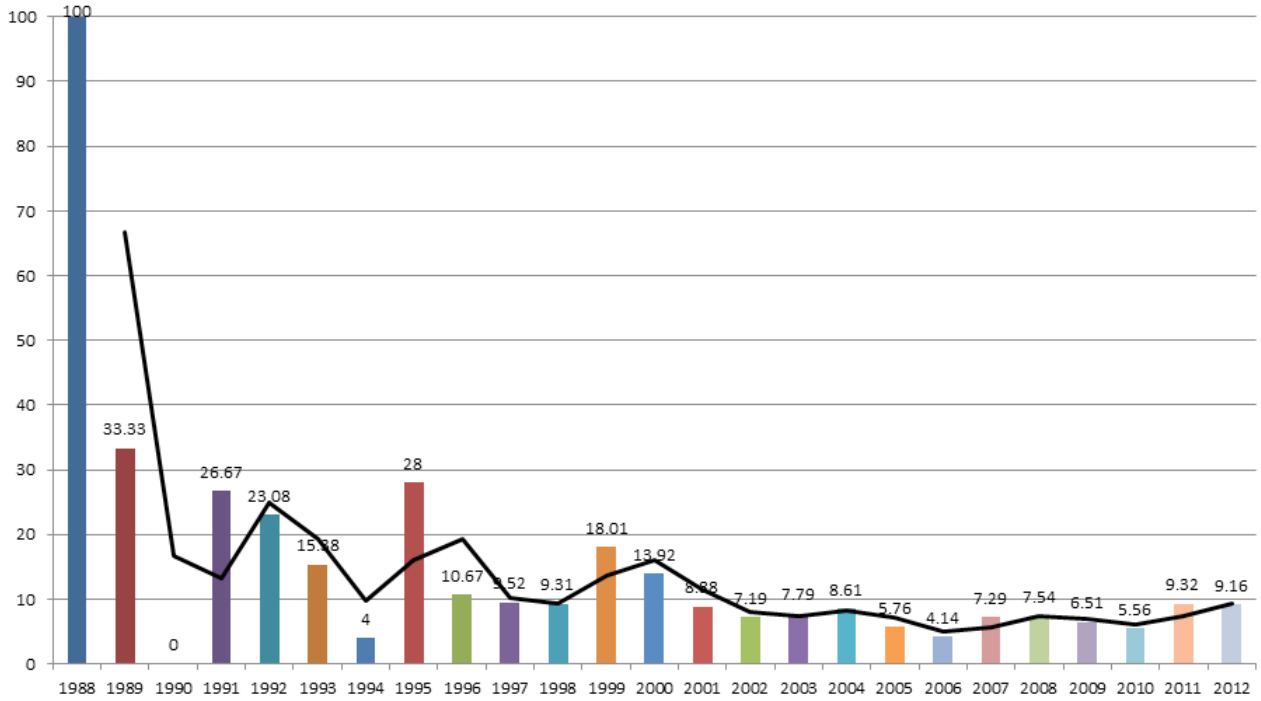


Figure 5. Critical severity vulnerabilities by year as a percentage of total vulnerabilities

Figure 5 again shows a peak in 2007, but also a peak in 2012. If we look at the percentage of all vulnerabilities that have a CVSS score of 10, we get a fairly steady number, which was on the decline from a 2007 high until 2010, but has since risen significantly.

Ignoring the first decade (and we should, given that the sample size of vulnerabilities is so small for those years), in figure 5 we see a peak at the end of the 1990s, followed by a steep decline until reaching a fairly steady value for a couple of years. Then there was a sharp drop in 2005 followed by 2006 which was the year that had the smallest percentage of high severity vulnerabilities compared to total vulnerabilities. The reason for this is actually explained by some of the data in the following sections.

By type

Let's take a look at the top vulnerabilities of the last 25 years by type.

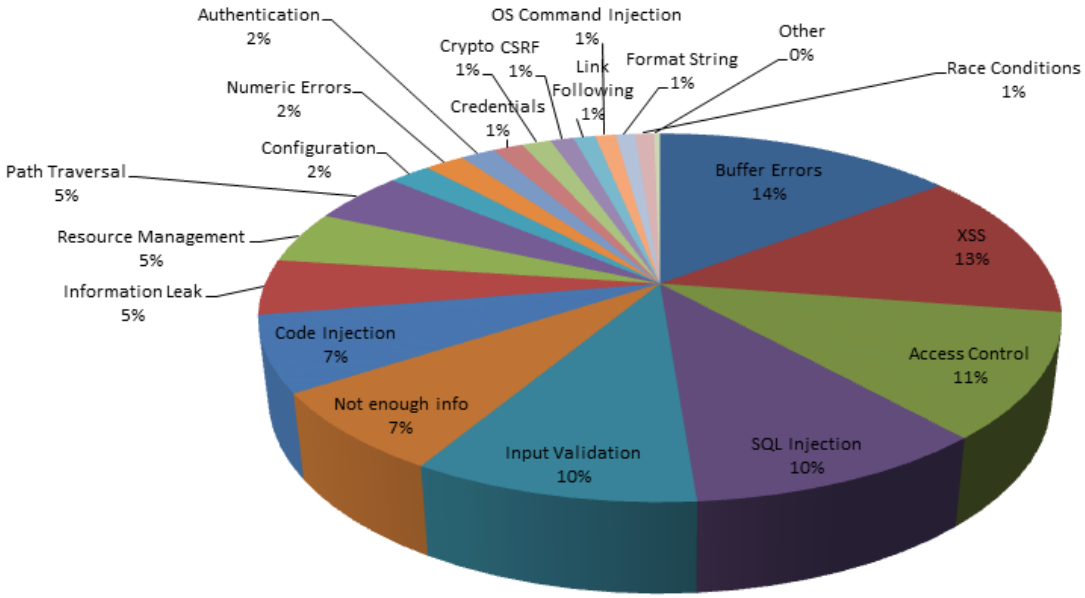


Figure 6. Top vulnerability types

Buffer overflows take the top spot with 7809 reported over the last 25 years. Cross site scripting (XSS) has also become an important vulnerability, clocking in as a close second to buffer overflows with 7006 occurrences.

But what if we look at only the vulnerabilities with a "High" severity rating?

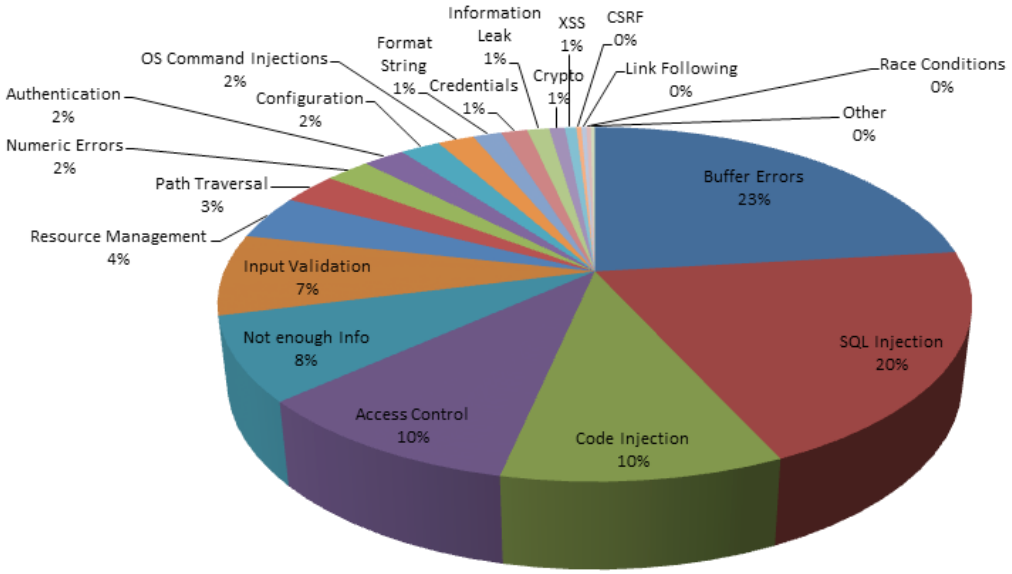


Figure 7. Top vulnerability types with a high severity

We can see that buffer overflows once again take the top spot with 5528 vulnerabilities having a high severity rating, while XSS has dropped to a mere 141 cases. We also see that the fourth most frequently reported vulnerability of the previous chart, SQL Injection, has suddenly become the second most important vulnerability when looking at severity.

Selecting only critical vulnerabilities (CVSS score of 10) yields additional significant information as shown below.

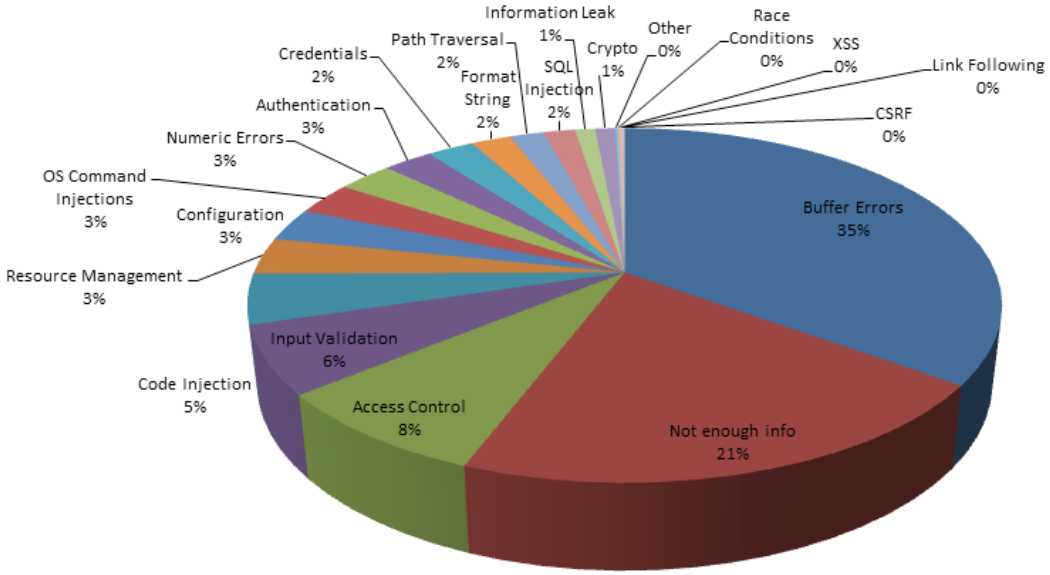


Figure 8. Top vulnerability types with a critical severity

Once again buffer overflows are the most important when only considering critical vulnerabilities, with 1391 occurrences over the past 25 years. We should note that there is also a very large number of critical vulnerabilities in the CVE database that have too little information to classify correctly.

In 2000, **Cowan et al.** [3] declared the buffer overflow *the vulnerability of the decade*. Despite some lack of data, we believe it is now safe to declare the buffer overflow the vulnerability of the quarter-century.

By ranking

Let's take a look at the top three vulnerabilities by year to see if we can predict if this trend will continue.

Given that the total number of vulnerabilities in CVE from 1988-1998 was 682 combined, less than the total number of vulnerabilities reported in 1999, we aggregated the numbers for 1988-1998 and used that for our comparison below.

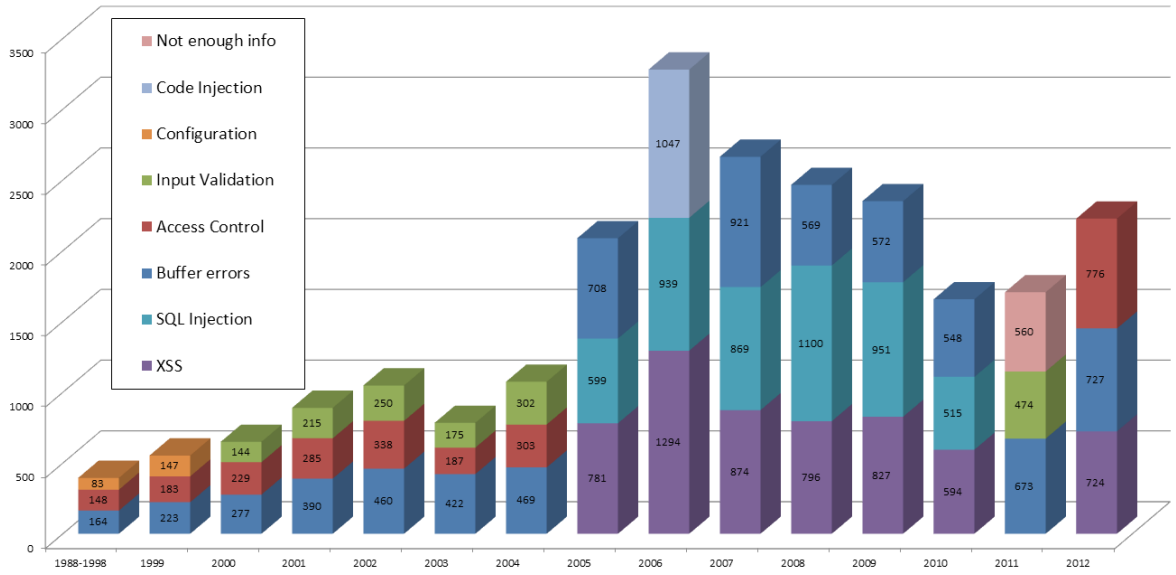


Figure 9. Top 3 vulnerability types by year

Looking at this comparison we can conclude that buffer overflows were the top vulnerabilities from 1988-1998 (Note: we should mention that this is mainly because so many were reported in 1997 and 1998. Prior to 1996 this was not the case, but very few vulnerabilities exist in CVE prior to 1996, so that comparison is not very useful).

It is now safe to declare the buffer overflow the vulnerability of the quarter-century.

Buffer overflows remained the top vulnerabilities every year after until 2005 when they were displaced by XSS. Buffer overflows disappeared completely from the top three in 2006 and XSS was the top vulnerability that year too, followed closely by code injection. In 2007 the trend reversed again and code injection disappeared from the top three and XSS was again displaced by buffer overflows as the top vulnerability. In 2008 and 2009, SQL injection was the most important vulnerability, only to be displaced by XSS and buffer overflows in 2010. In 2011 both SQL injection and XSS dropped out of the top three and buffer overflows took the top spot again, while in 2012 access control issues reigned supreme.

The data here explains some of the results we saw in the statistics "by year" section: Critical vulnerabilities as a percentage of total vulnerabilities dropped significantly in 2005 and 2006 compared to previous years. This data clearly shows that this is due to the high number of XSS vulnerabilities reported that year. These vulnerabilities rarely have a high or critical severity rating. The data also shows why the total number of vulnerabilities has dropped compared to 2005-2006: XSS vulnerabilities are now not assigned a CVE as often, resulting in a lower total number of vulnerabilities in the CVE data.

By vendor

A little more controversial approach is to review vulnerabilities by vendor. The NVD data has affected product information for 53,221 vulnerabilities. The top 10 vendors in terms of vulnerabilities account for 14,162 vulnerabilities or almost 27% of the total number of vulnerabilities. Here's a list of the 10 worst offenders.

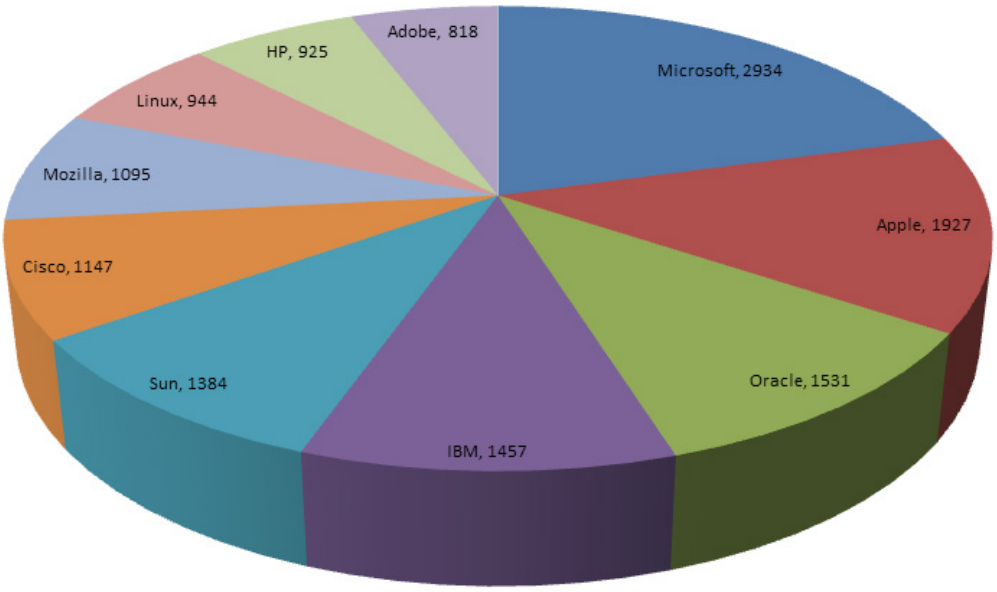


Figure 10. Top 10 vendors in terms of vulnerabilities

We can also look at those vendors with the most high severity vulnerabilities (CVSS score >= 7).

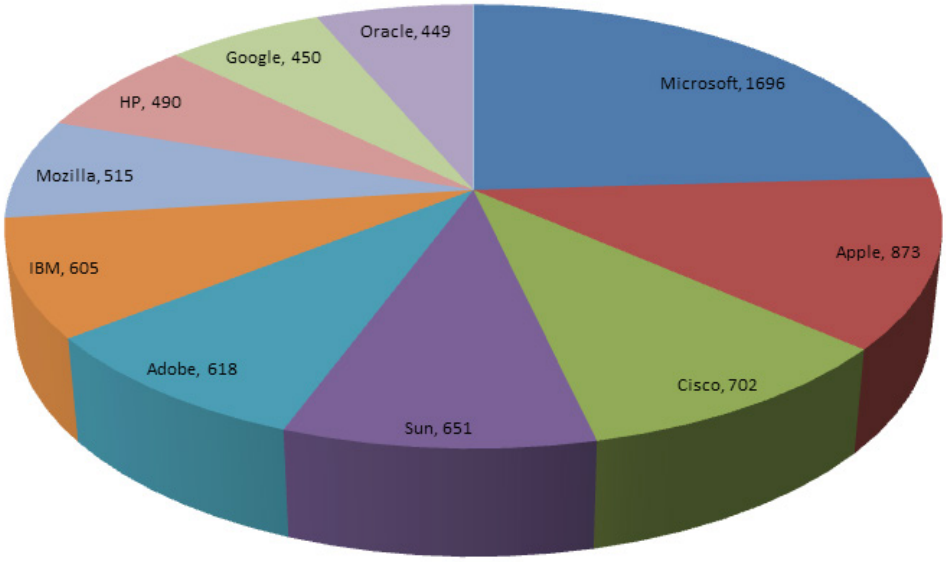


Figure 11. Top 10 vendors in terms of vulnerabilities with high severity

Which vendors are in the top 10 doesn't change that much: Google moves from 11th place in the total vulnerabilities list to 9th place for high severity and Linux drops from 8th place to 11th place. The top two spots stay the same, but the others move around a bit, with Oracle dropping to 10th place.

For vulnerabilities that matter most (CVSS score of 10), the vendors stay the same, but the list is reordered significantly, as shown below.

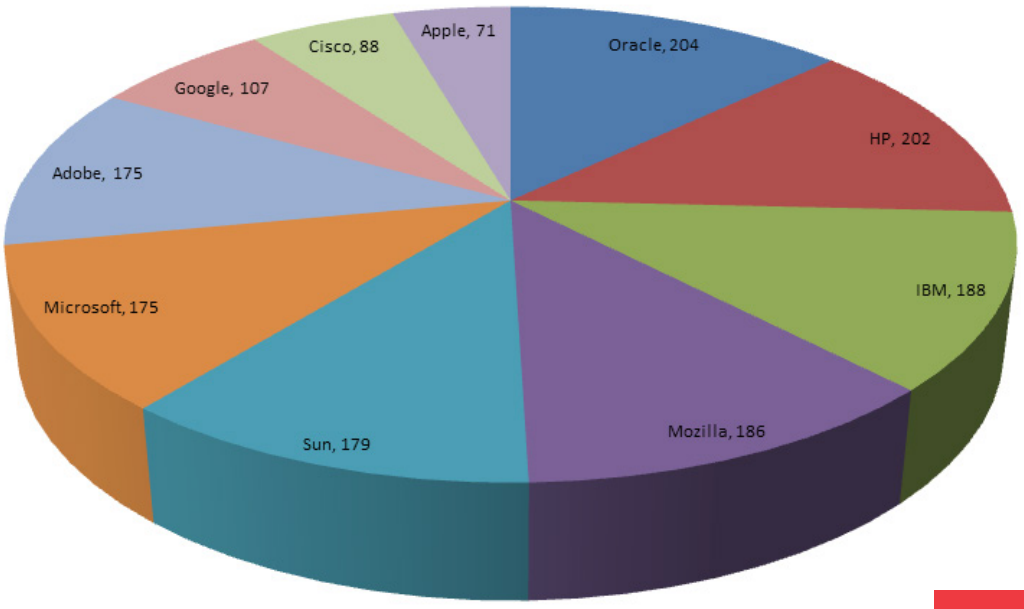


Figure 12. Top 10 vendors in terms of vulnerabilities with critical severity

Apple and Microsoft had more vulnerabilities overall, but fewer critical vulnerabilities.

What is interesting here is that Oracle, which was in 10th place for high or critical vulnerabilities is in 1st place when only critical vulnerabilities are considered. HP jumps up in the ranking as well. So, what these vendors lack in quantity of vulnerabilities, they make up for in severity. The reverse is true for Apple, which drops from second to 10th place, and to a lesser extent for Microsoft which drops from 1st to 6th place. Apple and Microsoft had more vulnerabilities overall, but fewer critical vulnerabilities.

Let's look at how our top 10 vendors fare over the years.

What these vendors [Oracle and HP] lack in quantity of vulnerabilities, they make up for in severity.

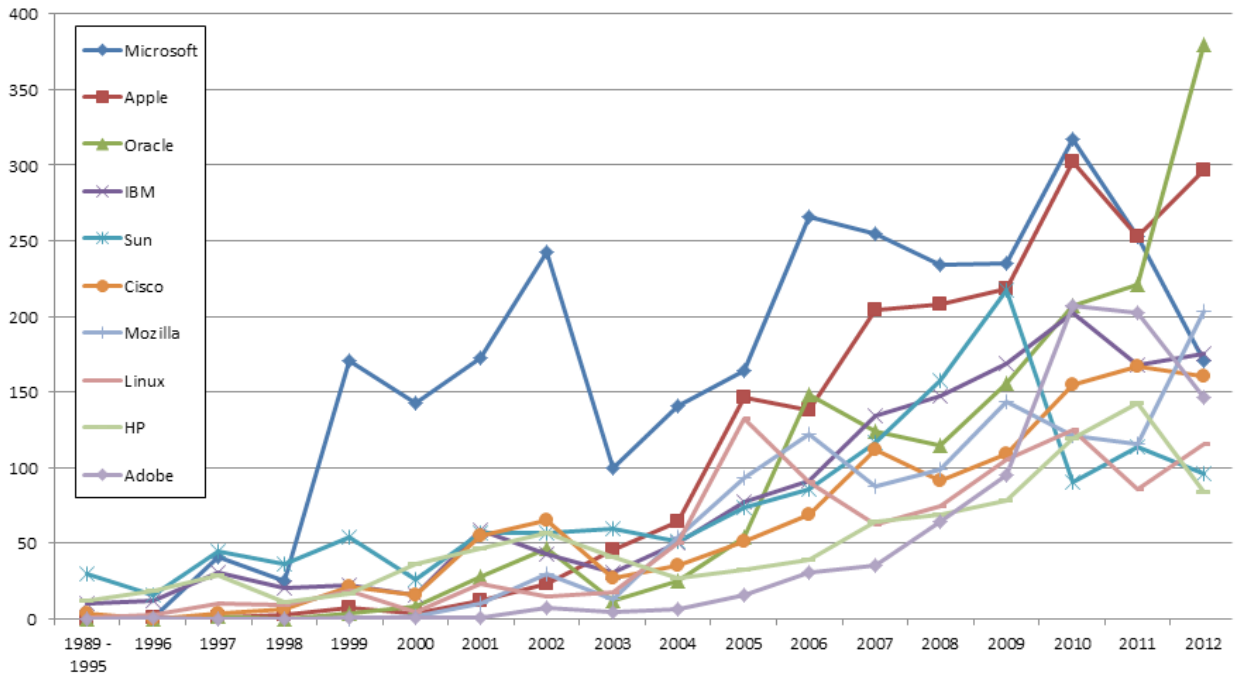


Figure 13. Top 10 vendors in terms of vulnerabilities over the years

Prior to 1996, vulnerabilities were in the single digits for all vendors included, so we totalled them up like we did in previous graphs. Of note, Microsoft had a steady rise of vulnerabilities up to 2002, then a sharp drop in 2003, only to continue to rise again to reach an all-time high in 2010, and has since been on the way down again. What's interesting about this is that in January 2002, Bill Gates sent the **Trustworthy Computing Initiative** [4] memo to all Microsoft employees, which was the start of a major focus on security at the company. The data also shows that Microsoft is the leader (of our top 10 vendors) for every year since 1999, except for 2011, when it tied with Apple at 253 vulnerabilities each and in 2012 when Oracle took the top spot. It is also interesting to note that the vendor who scored highest compared to the other top 10 vendors was also the company with the most vulnerabilities overall (i.e., when compared to all other vendors) for every year, except 2011, where Google (11th overall) was ranked first with 295 vulnerabilities. The reason for this is Chrome, which had 266 vulnerabilities in 2011. You may be wondering why Apple is ranked so high. The reason for this is that Webkit (the underlying browser framework for many modern browsers including Chrome) is considered an Apple product.

This brings us to another interesting view, comparison of products.

By product

So, let's look at which products have had the most vulnerabilities reported for them over the years.

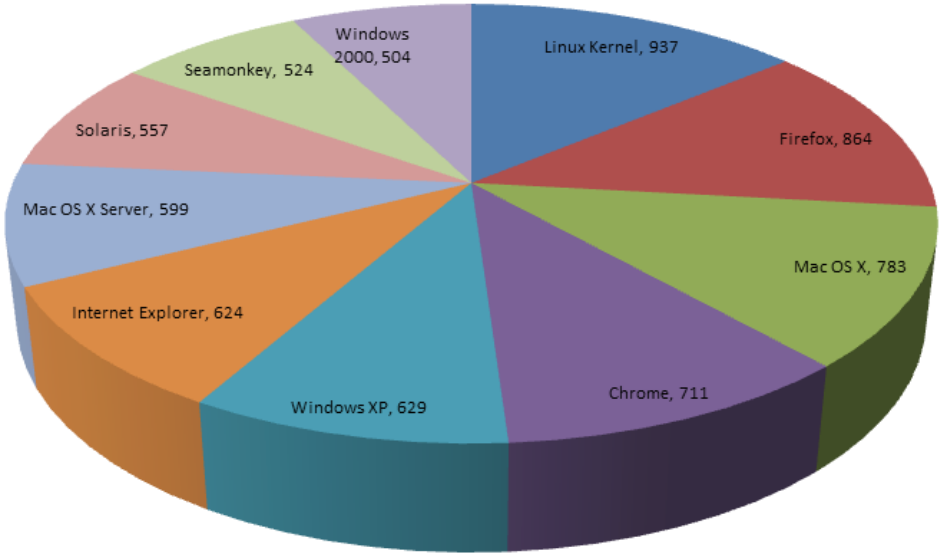


Figure 14. Top 10 products with the most reported vulnerabilities

This was quite a surprise to us; the Linux kernel has the most CVEs reported for it and, even though it has earned a **bad reputation** with respect to security, Flash Player does not make the top 10.

Although Linux is listed as number one, it's worth noting that various iterations of Windows are considered different products, while Linux is considered a single product and Mac OS X are considered three products, which further skews the data. If we account for unique CVEs for every possible version of Windows excluding the mobile ones (that's a total of 13 versions), we get a total of 1114 vulnerabilities in Windows. For Mac OS, which has three versions (including X and the previous Mac OS iterations), we get a total of 827. Of course these vulnerabilities in Windows and Mac OS are not solely in the kernel. Doing the same for Linux as Windows (by adding the unique CVEs assigned to major vendors like Ubuntu and Red Hat), we get a total of 1752 vulnerabilities. So while the top result was surprising to us, it still holds even when controlling for these extra factors. Mac OS, on the other hand, can console itself with the fact that it is not ahead of Windows in terms of total vulnerabilities.

Now let's look at the top 10 products by high and critical severity.

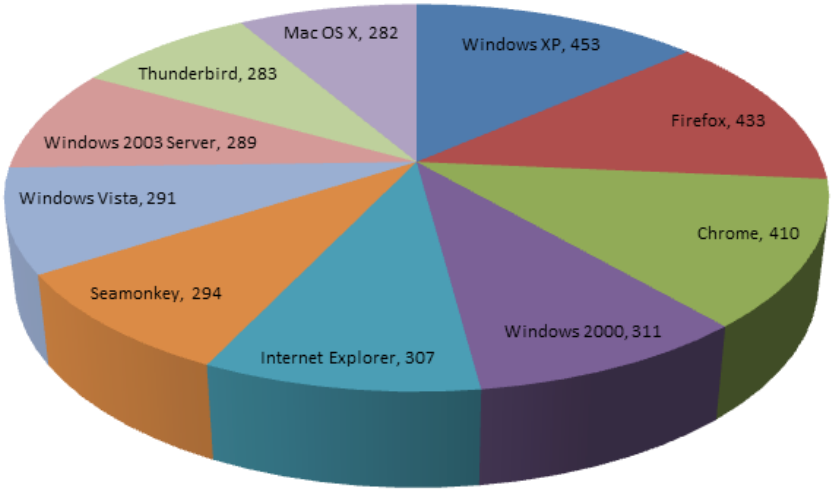


Figure 15. Top 10 products with high severity vulnerabilities

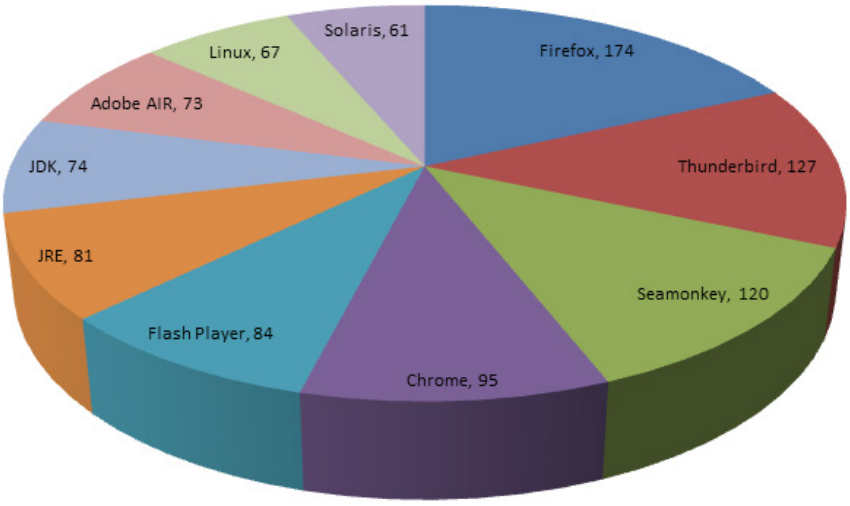


Figure 16. Top 10 products with critical severity vulnerabilities

For high severity, the list isn't that surprising with Windows XP in the number one spot, although Windows Vista is still listed relatively high compared to the level of **effort** [5] that Microsoft put into securing it. The critical list does have Flash Player, however only in 5th place. Firefox on the other hand takes the crown for critical vulnerabilities, while it's listed second for high severity and for total vulnerabilities. In fact, the top three spots for critical vulnerabilities are held by products from Mozilla. Of course, many of the same vulnerabilities probably affect multiple products, which is why Mozilla occupies these spots with three products. But this is the case with other vendors too; many of the same Flash Player vulnerabilities also apply to Air. What's also interesting here is that of the top four browsers that have a **total of 90% of the browser market share** [6], Firefox has the most vulnerabilities in every category, followed by Chrome, then Internet Explorer and finally Safari.

For high severity, the list isn't that surprising with Windows XP in the number one spot.

We discussed Linux versus Windows versus Mac OS X earlier, but how do the various versions of Windows stack up to other Windows versions and which Linux distribution had the most vulnerabilities? The following two charts can answer that question.

The top three spots for critical vulnerabilities are held by products from Mozilla.

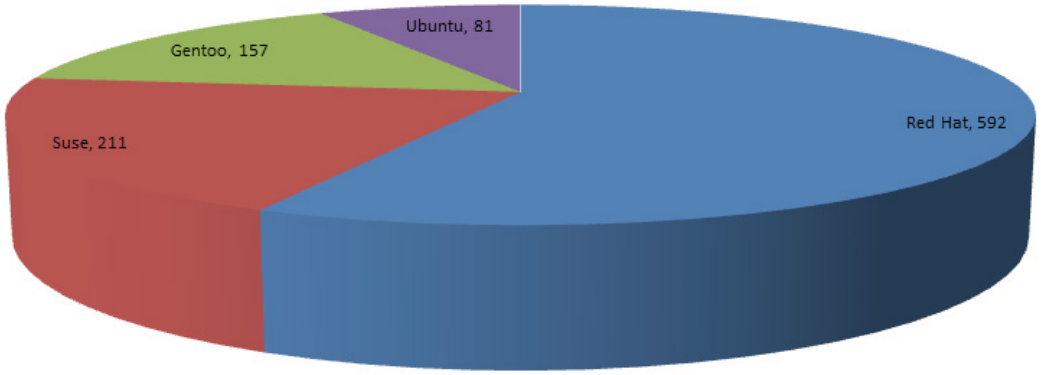


Figure 17. Vulnerabilities by Linux distribution

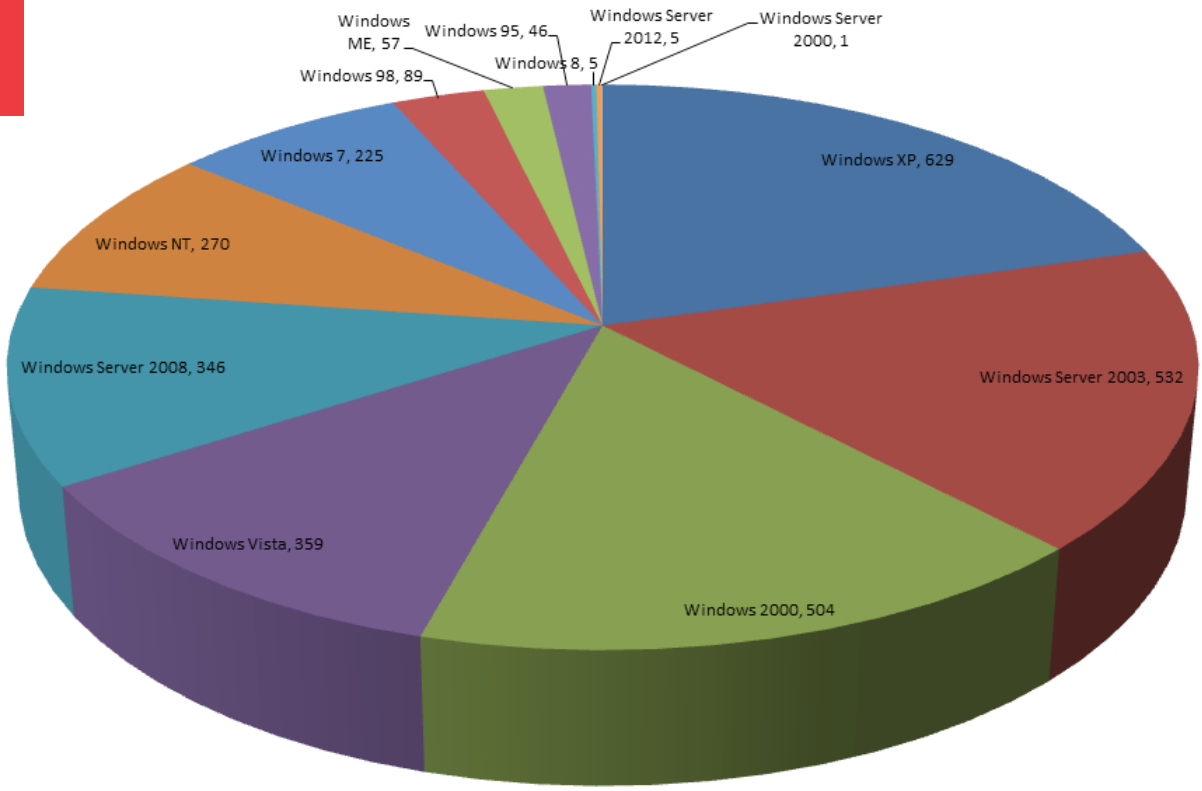


Figure 18. Vulnerabilities by Windows version

Note that in the above result, we considered several entries as equivalent (e.g., "windows_2003_server" and "windows_sever_2003"). While Red Hat had the most vulnerabilities reported for the Linux distributions, Windows XP had the most for Windows.

Let's also take a look at mobile phones.

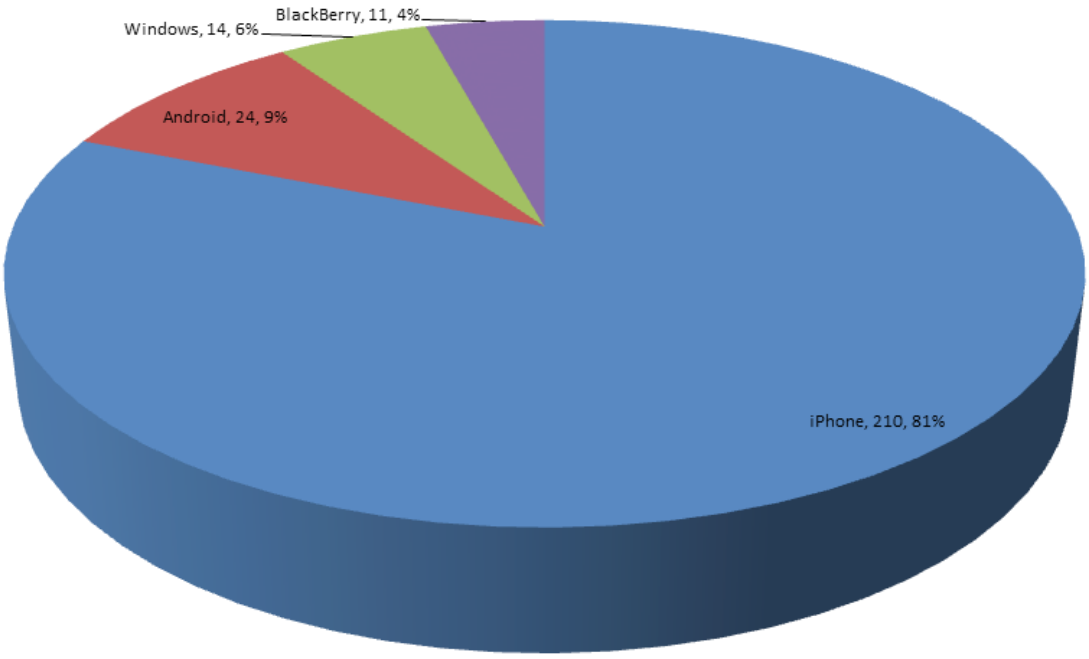


Figure 19. Mobile phone vulnerability market share

As you can see, the vast majority of mobile phone vulnerabilities have been found in iPhone. If we combine all the CVE vulnerabilities of the remaining three larger players they still come out at less than a quarter of the iPhone's CVEs. What's interesting here is that while Apple didn't focus very much on security when they first released the iPhone, they have since made significant improvements and can be considered the current market leaders in terms of mitigations. Note that for "Windows" we consider all Windows Mobile operating systems: Windows CE, Windows Mobile, Windows RT and Windows Phone.

So what have the trends been since 2007 (for completeness: BlackBerry had one in 2004 and one in 2005; Windows had one in 2001 and one in 2006)?

As you can see, the vast majority of mobile phone vulnerabilities have been found in iPhone.

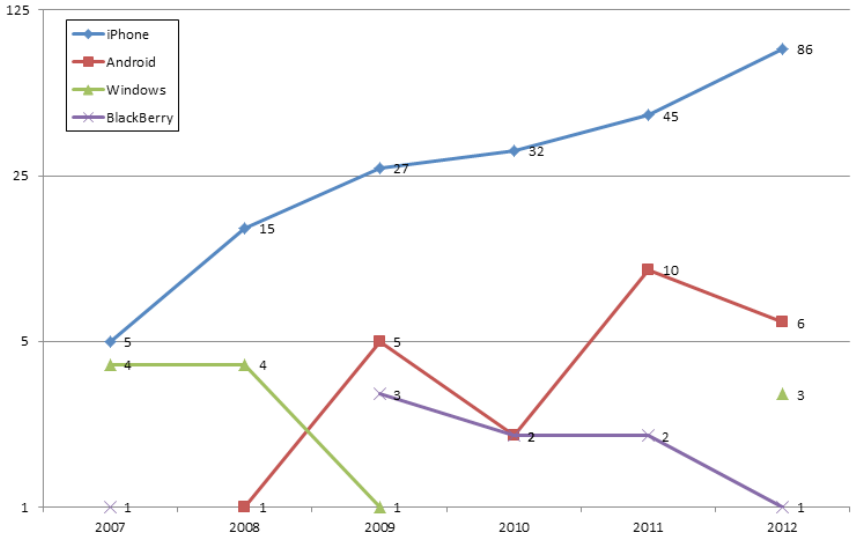


Figure 20. Mobile vulnerabilities trend (logarithmic scale)

As expected from the previous graph, Apple has the highest number of vulnerabilities every year. It's interesting to note though, that Apple has had significant CVE growth year over year, yet their OS has implemented more security features in subsequent iterations. While one may argue that the increase in CVEs is due to the increased popularity of the phone over the years, Android, the current market leader for mobile phone operating systems, has actually received fewer CVEs in 2012 than it did in 2011, even though it had **explosive growth in market share** [7].

By release date

If we combine this data with data from Microsoft about when their security bulletins were released versus when the CVE was published, we get the following chart.

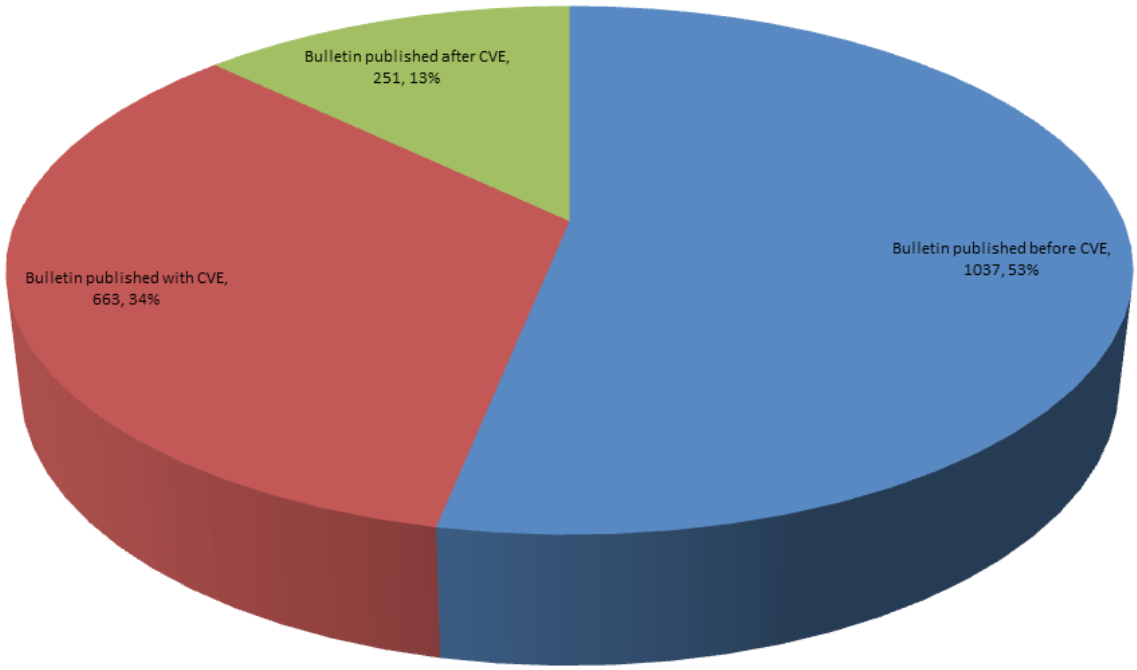


Figure 21. Microsoft security bulletin publication date versus CVE publication date

About 13% of the Microsoft security bulletins were published after the CVE was published. Since CVE normally coordinates with vendors, this means there was either active exploitation of this vulnerability, the vulnerability information was published before Microsoft had a chance to patch, or the vulnerability was in a third-party component that Microsoft relied on but was unable to patch before the vendor released the information publicly. In all three of these scenarios the vulnerability could be considered a 0-day. While this data is specific to Microsoft, it gives us some insight into how many 0-day vulnerabilities we can expect in other vendors.

Conclusion

The data tells us a few interesting things worth noting. Even though we've made lots of progress in mitigating attacks against buffer overflows and in building static analysis tools that attempt to detect these vulnerabilities, *buffer overflows remain one of the top ranking vulnerabilities year over year.*

The number of vulnerabilities that are assigned CVEs every year has declined over the last couple of years - except for 2012. The percentage of these vulnerabilities with high severity dropped significantly in 2012, but the percentage of critical vulnerabilities has risen significantly over the last two years. *So while fewer vulnerabilities were reported, the percentage of more critical vulnerabilities has increased.*

While it has had to deal with the most CVEs over the years, *Microsoft has significantly improved within the last couple of years and their browser and mobile operating systems are actually better than their competitors' in terms of vulnerabilities discovered.*

Google and Apple on the other hand have significantly different track records for their browsers when compared to their mobile operating systems. *Chrome is ranked as one of the highest for vulnerabilities, while Android has very few; iPhone has a significant lead on vulnerabilities, while Safari has the fewest compared to the other browsers.*

Methodology for modifying the NVD data

We started off by downloading the data from the **NVD website**. They maintain a database of all published CVEs together with information like the platforms affected by a vulnerability, using the **Common Platform Enumeration (CPE)** [8] dictionary to provide standardized names; the type of vulnerability, using a **subset** [9] of the **Common Weakness Enumeration (CWE)** [10] dictionary to provide standardized types of vulnerabilities.

NIST adopted CWE and CPE along the way and while they've done a great job of mapping the vulnerable software to CPE, the CWE mapping is incomplete. Of the 54,396 vulnerabilities that we imported into our database from the NVD XML files, only 25,761 have CWEs assigned to them. This meant that over half the vulnerabilities couldn't be used in any meaningful statistics on their types. However, older vulnerabilities, prior to the switch over to CWE in mid-2007, had some categories assigned to them. These categories weren't ideal and were often ambiguous, however they gave us a good starting point to try and categorize the remaining 28,635 vulnerabilities.

We started off by mapping the old categories "input:bounds" and "input:buffer" to CWE-119 which is named "Buffer Errors"; this gave us categories for 4,088 vulnerabilities. Vulnerabilities with category "access" were mapped to CWE-264 (Permissions, Privileges, and Access Control), while "race" was mapped to CWE-352 (Race Conditions) and config was mapped to CWE-16 (Configuration); these three gave us another 2,488 categorized vulnerabilities. Remaining were 22,059 vulnerabilities that were marked as "input" (9,999), design (4,565), other (388), exception (1,974) or env (210). The remaining 4,923 were not categorized.

These 22,059 vulnerabilities were put into categories based on their summaries: sometimes broad categories could be assigned based on the mention of specific keywords in the summaries (e.g., all vulnerabilities that used the word XSS in their summaries were set to CWE-79 (XSS)), but most had to be classified manually. This classification is far from perfect; sometimes not enough information was available to assign a category, which meant we assigned the category CWE-UNKNOWN; other times the vulnerability would not neatly fit into one of the CWE-subset categories that the NVD uses, which meant we assigned the category CWE-OTHER. Finally, a number of CVEs were assigned that ended up being rejected (489); these were assigned the category INVALID and are not included in any of the statistics above.

About Sourcefire

Sourcefire, Inc. (Nasdaq:FIRE), a world leader in intelligent cybersecurity solutions, is transforming the way global large- to mid-size organizations and government agencies manage and minimize network security risks. With solutions from a next-generation network security platform to advanced malware protection, Sourcefire provides customers with Agile Security® that is as dynamic as the real world it protects and the attackers against which it defends. Trusted for more than 10 years, Sourcefire has been consistently recognized for its innovation and industry leadership with numerous patents, world-class research, and award-winning technology. Today, the name Sourcefire has grown synonymous with innovation, security intelligence and agile end-to-end security protection. For more information about Sourcefire, please visit **www.sourcefire.com**.

References

[1] MITRE. About CVE. http://cve.mitre.org/about/index.html

[2] National Institute of Standards and Technology. National Vulnerability Database. http://nvd.nist.gov/

[3] Crispin Cowan, Perry Wagle, Calton Pu, Steve Beattie, and Jonathan Walpole. Buffer Overflows: Attacks and Defenses for the Vulnerability of the Decade. . DARPA Information Survivability Conference and Expo (DISCEX), Hilton Head Island SC, January 2000.

[4] Bill Gates. Memo from Bill Gates. http://www.microsoft.com/en-us/news/features/2012/jan12/GatesMemo.aspx. January 2002.

[5] Microsoft. Security Enhancements in Windows Vista. May 2007. http://www.microsoft.com/en-us/download/details.aspx?id=10591

[6] W3Counter. Web Browser Market Share. December 2012. http://www.w3counter.com/globalstats.php?year=2012&month=12

[7] Gartner. Market Share: Mobile Phones by Region and Country, 3Q12. November 2012.

[8] MITRE. Common Platform Enumeration. http://cpe.mitre.org/

[9] National Institute of Standards and Technology. CWE - Common Weakness Enumeration. http://nvd.nist.gov/cwe.cfm

[10] MITRE. Common Weakness Enumeration. http://cwe.mitre.org/

2.13 I REV1