Analysis of Vulnerability Disclosure Delays from the National Vulnerability Database

Luis Gustavo Araujo Rodriguez[1], **Julia Selvatici Trazzi**[1],
Victor Fossaluza[1], **Rodrigo Campiolo**[2], **Daniel Macêdo Batista**[1]

[1]Institute of Mathematics and Statistics – University of São Paulo (USP)
São Paulo – SP – Brazil

[2]Federal University of Technology – Paraná (UTFPR)
Campo Mourão – PR – Brazil

{luisgar,victorf,batista}@ime.usp.br

rcampiolo@utfpr.edu.br, julia.trazzi@usp.br

Abstract. *The Internet contains vast amounts of data; consequently, hindering information retrieval. Resources, such as the National Vulnerability Database (NVD), have emerged to remedy this situation. Organizations largely depend on the NVD in order to disclose vulnerabilities and collaborate towards a solution. However, there has been evidence that other sources are disclosing vulnerabilities more efficiently and rapidly. The objective of this paper is to evaluate vulnerability disclosure delays from the NVD in order to state its efficiency. Among several findings, we observed that the majority of vulnerabilities are delayed within 1-7 days. Based on these results, we provide recommendations for those who currently rely only on NVD, such as IoT manufacturers and developers.*

1. Introduction

For two decades, the Internet has become a valuable source to collect and analyze information. Currently, considerable effort is required to manage the Internet's vast amounts of data, since information should be handled *efficiently* and *intelligently*. Thus, an approach called Open Source Intelligence (OSINT) emerged to remedy this situation [Yang and Lee 2012].

OSINT can be defined as a service that offers information to users in order to generate knowledge. The end result must: (1) be generated based on verified information; and (2) meet expectations and provide useful feedback [Lee and Shon 2016]. OSINT provides data collected from various sources such as social media sites, reports or journals. The content is publicly available and accessible, thus possessing properties of openness. Furthermore, OSINT provides low-cost and high-level opportunities, as well as up-to-date information [Yang and Lee 2012]. This approach is beneficial to several fields, including cybersecurity. OSINT supports organizations to secure their networks, systems and users from cyberattacks by propagating information about vulnerabilities [Best 2011].

A vulnerability can be defined as a security flaw in an information system. Over the last few years, the number of vulnerabilities has increased to an unexpected degree [Macdonald et al. 2015]. In particular, approximately 10,555[1] vulnerabilities were found

[1] https://www.cvedetails.com/browse-by-date.php (Accessed: 09/21/2017)

between January and September 2017, reaching an annual growth rate[2] of 27.95% (Figure 1). The high number of vulnerabilities incites cyberattackers to disclose security exploits by taking advantage of the Internet's sharing capabilities.

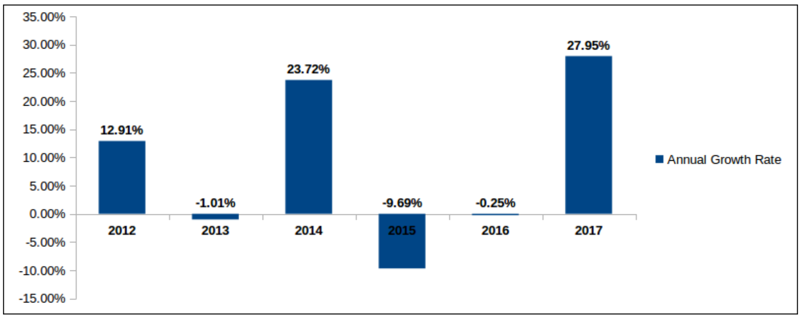


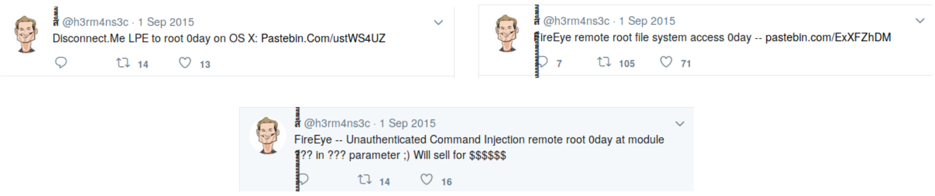
Figure 1. Annual growth rate (2012 - 2017 (January to September))

The targets of cyberattackers can be diverse. However, recently, there has been a growing interest in vulnerabilities of devices on the so called Internet of Things (IoT). Recent and important events related to IoT vulnerabilities include: (1) *Mirai* was found and is the first malware that targets connected devices in a network [Kolias et al. 2017]; (2) Distributed Denial of Service attack on IoT devices operated by DNS provider *Dyn* (2016) [Gharaibeh et al. 2017]; and (3) ransomware on surveillance cameras in Washington D.C. [Washington Post 2017].

The most prominent example of IoT malware is Mirai, which is a malicious software for IoT devices. The malware spreads to devices using default passwords, consequently creating botnets that send large datasets to a target. Furthermore, the malware code is available as open source, and several variations have been created, thereby spreading and infecting further devices. Variants of Mirai include *Satori*, *Okiru*, *Masuta* and *Puremasuta*. Some IoT devices are not envisioned to receive either software or security updates [Sinanovic and Mrdovic 2017]. Thus, Mirai took advantage of IoT's deficiencies. These problems confirm the urgency to better-protect cyberspace. Thus, several mechanisms using Open-Source Intelligence have been proposed over the last few years. The National Vulnerability Database (NVD) is an example of an OSINT resource concerning security vulnerabilities [Joshi et al. 2013]. This database has several components such as the Common Vulnerabilities and Exposure (CVE) initiative, developed by the MITRE corporation, which is an industry-standard dictionary containing a list of security vulnerabilities [Bhuddtham and Watanapongse 2016]. Each exploit has a unique identifier, consequently enabling easier reference and collaboration towards a solution.

Although the purpose of NVD is to promote cybersecurity situation awareness, there has been evidence that other sources are handling this task more efficiently and rapidly [Santos et al. 2012]. An example can be seen in Figure 2, in which a user informs zero-day exploits on Twitter before it being disclosed on official OSINT sources. Furthermore, the user attempts to sell the exploit, thereby engaging in illegal commercial activity.

[2] Formula: (Number of Vulnerabilities of Current Year/Number of Vulnerabilities from Previous Year)$^{(1/2)}$ -1



@h3rm4ns3c · 1 Sep 2015
Disconnect.Me LPE to root 0day on OS X: Pastebin.Com/ustWS4UZ

@h3rm4ns3c · 1 Sep 2015
FireEye remote root file system access 0day -- pastebin.com/ExXFZhDM

@h3rm4ns3c · 1 Sep 2015
FireEye -- Unauthenticated Command Injection remote root 0day at module ??? in ??? parameter ;) Will sell for $$$$$$

Figure 2. Zero-day vulnerabilities disclosed on Twitter.

This is just one out of several examples on how official OSINT sources are handling vulnerability disclosures inefficiently. Thus, this paper proposes evaluating vulnerability disclosure delays from `NVD`, as well as other sources. Web scrapers were developed in order to examine disclosure dates from each chosen source. Based on the collected data, we offer a detailed analysis and have large emphasis on the surface web. The contribution of this paper is to offer cybersecurity situation awareness and propose improvements to better-protect cyberspace. In order to allow replication of the experiments, all implemented code is publicly available under the GNU General Public License v3.0 at `https://github.com/luisgar1990/vuln_delays`.

The rest of this paper is organized as follows. Section 2 presents related works. Section 3 highlights the objectives of this paper. Section 4 explains the methodology used for data collection. Section 5 presents and discusses the results. Section 6 presents recommendations for organizations who rely on `NVD`. Finally, Section 7 marks the conclusions of this paper.

2. Related works

[Santos et al. 2012] analyzed security messages on Twitter to determine if online social networks were an effective approach to spread vulnerability situation awareness. The researchers implemented web crawlers to extract information. The authors confirmed that online social networks can be used as efficient tools to disclose vulnerabilities.

[Macdonald et al. 2015] implemented web crawlers in order to monitor malicious forums and; thus, identify potential threats. The author highlighted the importance of analyzing informal sources to better-protect cyberspace.

[Guojun et al. 2017] implemented web crawlers for threat awareness of public vulnerabilities. The authors stated that dynamic web crawlers can be used for situation awareness, thereby reducing attack time windows between vulnerability disclosure and recovery.

The company *Recorded Future* [Recorded Future 2017] examined vulnerabilities between the initial disclosure to their release on the National Vulnerability Database. The goal was to better-comprehend the vulnerability timeline of both security and adversary communities. Key information was revealed, such as: (1) an average of 7 days passed between vulnerability disclosures and `NVD` publications; (2) extraction-based techniques using Recorded Future's platform are effective; and (3) a list of the top unpublished CVEs on the dark web. It is highlighted the importance of extraction-based approaches in order to gather information about vulnerabilities. However, these aforementioned techniques were not explicitly detailed, thereby hindering replication. Furthermore, the authors used few statistical tests and procedures. Additionally, Recorded Future placed a large emphasis on vulnerability disclosures on the dark web. We strongly believe that its counterpart, the surface web, should be analyzed in detail in terms of vulnerability disclosures. The surface web provides simplicity in terms of disclosing and gathering information, thereby attracting threat actors.

It is worth highlighting that we aim to offer a more detailed analysis than [Recorded Future 2017]. Thus, our analysis will differ from it by:

- showing a detailed explanation of the methodologies and results, thereby allowing replication.
- placing emphasis on the surface web, as we believe it offers more simplicity for disclosing vulnerabilities (the dark web is already well known for acting as a tool for illegal activities [Hurlburt 2017]).
- extracting data from 2017, thereby offering a more current-day analysis.
- listing vulnerabilities from `NVD` that had the highest disclosure delays.

Specifically related to IoT, [Sinanovic and Mrdovic 2017] analyzed Mirai in detail, and considered it to be a prime example of the current and vulnerable state of IoT. The authors highlight the importance of improving security on IoT devices, which were not designed to receive security updates.

3. Objectives

As stated before, there has been evidence that official OSINT sources are not efficient in terms of vulnerability disclosures. Extraction-based techniques for threat awareness have been garnering significant attention over the last few years [Mittal et al. 2016].

The objective of this paper is to evaluate vulnerability disclosure delays from `NVD` in order to provide information concerning its efficiency. We aim to tackle the following research questions:

Q1 How many days transpire between vulnerability disclosures in other sources and `NVD` publications?

Q2 Are other sources more efficient for disclosing vulnerabilities? If so, why?

Q3 Which vulnerability rating is disclosed the most?

Q4 Which types of vulnerabilities published in the `NVD` have longer publication delays?

4. Data collection method

Information-retrieval-based techniques are used to search and collect data from websites. As more information becomes available on the Internet, better strategies are required in order to collect the data effectively. Normally, the techniques are divided into three phases: web crawling; web scraping; and data storing.

Web crawlers are Internet bots that automatically transverse and download webpages. Next, web scraping is done to collect information.Websites may contain unnecessary data and; thus, web scraping should be done effectively, extracting only useful and meaningful information [Mahto and Singh 2016]. In terms of data storing, CSV files are considered to be a standard for data storage, because it offers different file compatibility.

Our setup for collecting vulnerabilities from 2017 can be summarized as follows.

Programming language: Python 3.6, which is currently the latest stable release.

Web crawler: Built-in functions of Python such as *urllib.request* are used for web crawling, thereby offering efficiency.

Web scraper: Beautiful Soup 4, which is the latest version, is used for web scraping. Selenium and PhantomJS were used for scraping Javascript websites. A total of 5 web scrapers, one for each source, were developed. They are available as free software at `https://github.com/luisgar1990/vuln_delays`.

Web storing: The web scrapers compare vulnerability disclosure dates and store the results in a CSV file.

Parser: NVD provides vulnerability information within XML files. Thus, a high-speed library called *lxml* was used for parsing these files.

Selection policy: Vulnerabilities with entry dates from 2017 and publish dates between 01/01/2017 and 11/05/2017 were collected. We've gathered the publish dates, severity level (CVSS) and type (CWE) of the vulnerability.

The sources used for comparison purposes were: SecurityFocus Database (`http://www.securityfocus.com/bid/`), ExploitDB (`https://www.exploit-db.com/exploits/`), Cisco Security Advisory (`https://tools.cisco.com/security/center/`), Wireshark Bug Database (`https://bugs.wireshark.org/bugzilla/`), and Microsoft Official Bulletins (`https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/`). We chose these sources because of their well-established recognition and contribution to cybersecurity [Fang and Hafiz 2014]. All vulnerabilities retrieved and considered in our analysis were disclosed in 2017.

SecurityFocus is one of the most prominent and well-respected vulnerability databases [Zegeye and Sailio 2015]. It is the second most widely-used vulnerability database [Fang and Hafiz 2014], after NVD. However, the former discloses more vulnerabilities [Fang and Hafiz 2014], thereby offering a significant advantage.

ExploitDB is another widely-used database, which discloses at least the code to replicate the vulnerability [Younis and Malaiya 2015]. Furthermore, a survey shows that reporters favor ExploitDB over SecurityFocus for disclosing vulnerable code to users [Fang and Hafiz 2014].

Wireshark is an open source packet-analyzer used by security engineers in order to detect vulnerabilities in the network. Devices such as the Catalyst, network switches sold by Cisco, support Wireshark. Cisco is considered a prominent and influential network company [Bhardwaj and Kole 2016]. Based on these statements, blogs from Cisco and Wireshark were taken into consideration in our analysis.

Microsoft also uses its official security bulletin to disclose vulnerabilities related to their products. CVE claims that Microsoft is the vendor with the highest number of distinct vulnerabilities[3]. Thus, Microsoft's official security bulletin was also considered.

5. Results and analysis

We extracted vulnerability disclosure dates (commonly known as *original release* or *publish* dates) from both NVD and the aforementioned sources. Table 1 presents the number

[3] `https://www.cvedetails.com/top-50-vendors.php?year=0` (Accessed: 11/02/2017)

of vulnerabilities collected from each source as well as the number of vulnerabilities from which NVD had disclosure delays. Furthermore, the vulnerability with the highest disclosure delay is also shown.

Table 1. Disclosure delays of NVD

Source	Total number of vulnerabilities collected from 2017	Total number of NVD delays	Highest disclosure delay from NVD	
SecurityFocus	5525	3973	CVE-2017-5637	244 days
ExploitDB	732	263	CVE-2017-1002000	195 days
Cisco	321	291	CVE-2017-3848	37 days
Wireshark	50	50	CVE-2017-6467	392 days
Microsoft	590	406	CVE-2017-8575	10 days

SecurityFocus managed to disclose 71.91% of vulnerabilities before NVD. ExploitDB published 35.93% of vulnerabilities before NVD. It is worth highlighting that although this result may not be more than 50%, it can be considered severe because the exploit code is published, thereby offering opportunity to spread the vulnerability. In terms of network-related vulnerabilities, Wireshark's Bug Database managed to disclose all vulnerabilities before NVD. Moreover, Cisco's security advisory forums managed to disclose 90.65% of vulnerabilities before NVD. Microsoft's security bulletin managed to disclose 68.81% of vulnerabilities faster than NVD.

Based on these results, NVD has an undeniable inefficiency for vulnerability disclosures.

5.1. Top disclosure delays on 2017

Table 2 presents the top 5 disclosure delays, from NVD, for each source. It is worth highlighting the significant delays for CVE-2017-6467, CVE-2017-5637 and CVE-2017-1002000, which were published by NVD in 392, 244, and 195 days, respectively, after their initial disclosure.

5.2. Disclosure delays based on CVSS

Figure 3 presents the number of vulnerabilities, based on their severity level, that had disclosure delays.

SecurityFocus Vulnerabilities classified as *Medium* had the highest disclosure delays, being 56.68%. *High* and *Critical* vulnerabilities ranked second and third, being 15.58% and 12.36%, respectively. Vulnerabilities classified as *Low* had the fourth highest disclosure delay, being 11.13%. It is worth mentioning that 4.25% of vulnerabilities delayed were Not Defined (*ND*).

ExploitDb Vulnerabilities classified as *Medium*, *High*, and *Critical* had once again the highest, second-highest and third-highest disclosure delays, being 52.47%, 22.43% and 12.17% respectively. *ND* and *Low* vulnerabilities were delayed the least, being 10.65% and 2.28% accordingly.

Wireshark Only vulnerabilities classified as *Medium* and *High* were delayed, being 58% and 42% respectively.

Table 2. Top disclosure delays

CVE	CWE	CVSS	NVD-Publish-Date	Site-Publish-Date	Days Delayed
NVD vs CISCO					
CVE-2017-3848	CWE-79	Medium	04-07-2017	03-01-2017	37
CVE-2017-6674	CWE-20	Medium	06-13-2017	05-24-2017	20
CVE-2017-6736	CWE-119	Critical	07-17-2017	06-29-2017	18
CVE-2017-6737	CWE-119	Critical	07-17-2017	06-29-2017	18
CVE-2017-6738	CWE-119	Critical	07-17-2017	06-29-2017	18
NVD vs EXPLOIT DATABASE					
CVE-2017-1002000	CWE-434	High	09-14-2017	03-03-2017	195
CVE-2017-1002001	CWE-434	High	09-14-2017	03-03-2017	195
CVE-2017-1002002	CWE-434	High	09-14-2017	03-03-2017	195
CVE-2017-1002003	CWE-434	High	09-14-2017	03-03-2017	195
CVE-2017-1002008	CWE-434	High	09-14-2017	03-16-2017	182
NVD vs MICROSOFT					
CVE-2017-8575	CWE-200	Low	06-29-2017	06-19-2017	10
CVE-2017-8576	CWE-264	Medium	06-29-2017	06-19-2017	10
CVE-2017-8579	CWE-264	Medium	06-29-2017	06-19-2017	10
CVE-2017-8552	CWE-264	High	06-14-2017	06-05-2017	9
CVE-2017-8518	CWE-119	High	08-10-2017	08-04-2017	6
NVD vs SECURITYFOCUS					
CVE-2017-5637	ND	ND	10-09-2017	02-07-2017	244
CVE-2017-9368	ND	ND	10-16-2017	03-02-2017	228
CVE-2017-5635	ND	ND	10-19-2017	03-06-2017	227
CVE-2017-5636	ND	ND	10-19-2017	03-06-2017	227
CVE-2017-5208	CWE-190	Medium	08-22-2017	01-08-2017	226
NVD vs WIRESHARK					
CVE-2017-6467	CWE-20	Medium	03-03-2017	02-05-2017	392
CVE-2017-11409	CWE-399	High	07-18-2017	04-15-2017	94
CVE-2017-13766	CWE-787	Medium	08-30-2017	06-24-2017	67
CVE-2017-9352	CWE-399	High	06-02-2017	04-14-2017	49
CVE-2017-11411	CWE-399	High	07-18-2017	06-01-2017	47

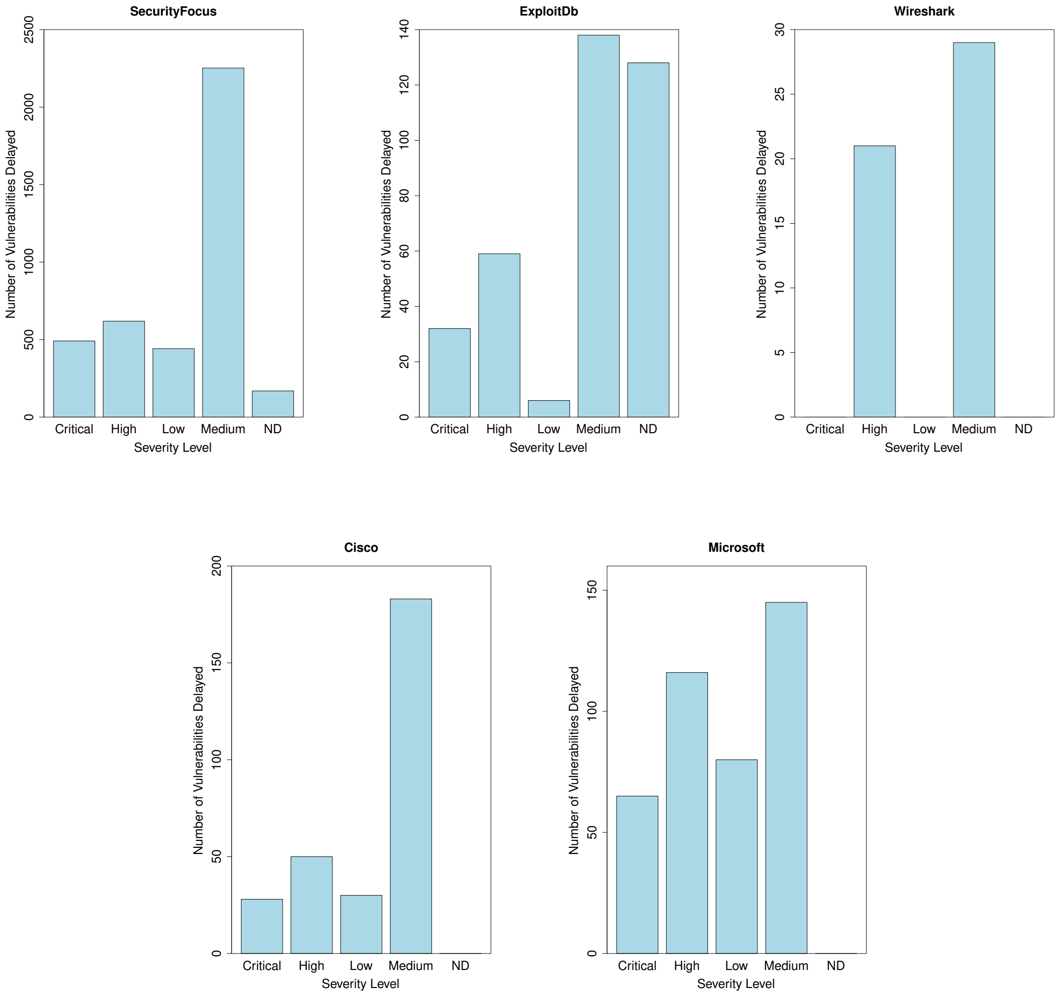


Figure 3. Disclosure delays from NVD based on severity levels.

Cisco *Medium* and *High* vulnerabilities were delayed the most, being 62.89% and 17.18% accordingly. 10.31% and 9.62% of vulnerabilities delayed were *Low* and *Critical*.

Microsoft Vulnerabilities classified as *Medium* and *High* vulnerabilities had the highest disclosure delays, being 35.71% and 28.57% respectively. *Low* and *Critical* were delayed the least, being 19.70% and 16.01% accordingly.

Based on these results, a pattern can be detected in which vulnerabilities classified as *Medium* and *High* are delayed the most. Therefore, `NVD` should place greater emphasis on these levels of vulnerabilities because they possess a considerable threat to cyberspace.

5.3. Disclosure delays based on vulnerability types

Figure 4 presents the top 5 disclosure delays based on vulnerability types.

SecurityFocus *Improper Access Control* was the highest vulnerability type delayed, reaching 21.77%. *Buffer Errors* and *Information Leak/Disclosure* reached 13.97% and 11.81%, thus being the second and third highest vulnerability type delayed.

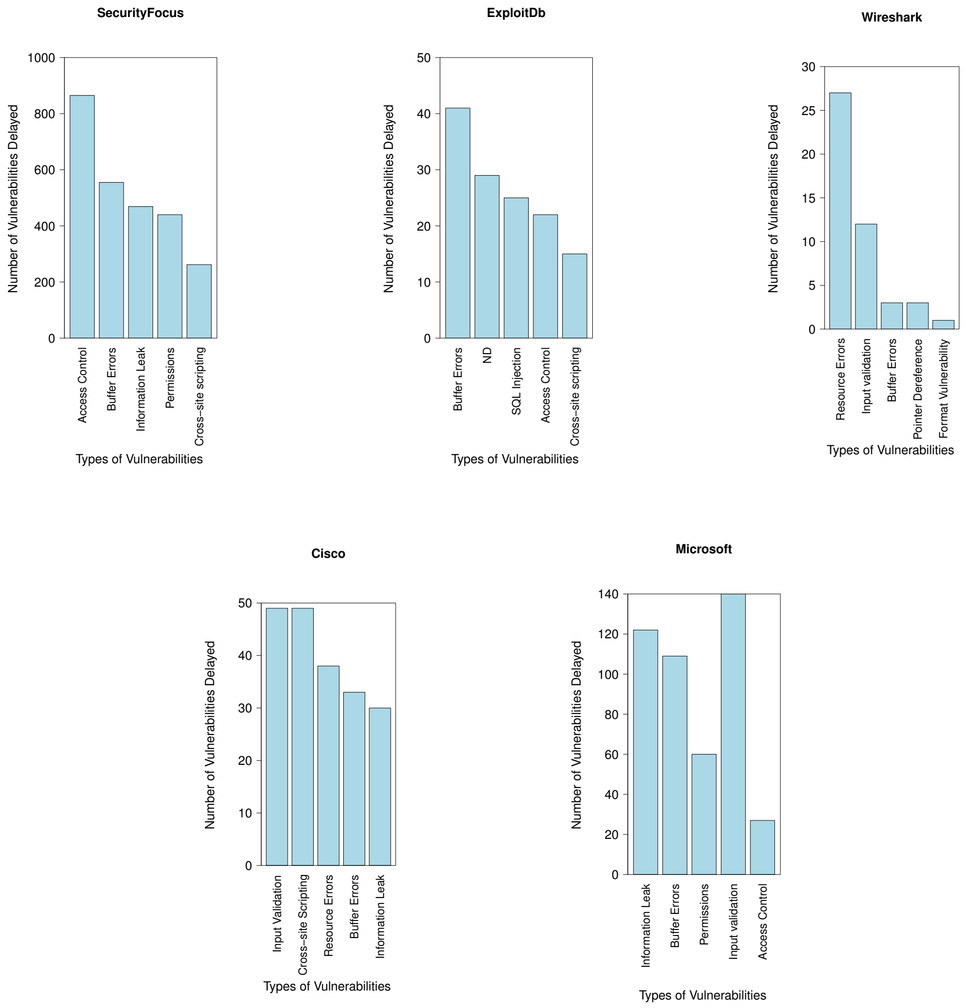


Figure 4. Disclosure delays from NVD based on vulnerability types.

Permissions, Privileges, and Access Control and *Cross-site Scripting (XSS)* were the fourth and fifth highest, being 11.07% and 6.59% respectively.

ExploitDb *Buffer Errors* was 15.59%, being the highest vulnerability type delayed by NVD. ND vulnerabilities were the second highest, reaching 11.03%. *SQL Injection*, *Improper Access Controls*, and *Cross-site Scripting(XSS)* were the third, fourth and fifth highest vulnerability type delayed, being 9.51%, 8.37% and 5.70% respectively.

Wireshark *Resource Management* and *Input Validation* were the highest vulnerability types delayed, being 54% and 24% respectively. *Buffer Errors* and *NULL Pointer Dereference* had an equal number of vulnerability types delayed, both reaching 6%. 2% of vulnerability types delayed were related to *Format String Vulnerability*.

Cisco *Input Validation* was the highest vulnerability type delayed by NVD, being 16.84%. *Cross-site Scripting (XSS)* and *Resource Management Errors* were the second and third highest vulnerability type delayed, reaching 16.84% and 13.06% respectively. *Buffer Errors* and *Information Leak Disclosure* reached lower, but proximate delays, being 11.34% and 10.31% respectively.

Microsoft *Information Leak Disclosure* and *Buffer Errors* were the highest and second highest vulnerability types delays, reaching 30.05% and 26.85% accordingly. *Permissions, Privileges and Access Control* was the third highest vulnerability type delayed by NVD, being 14.78%. *Input Validation* and *Improper Access Controls* reached 9.85% and 6.65% respectively.

The analysis of these sources reveals that *Buffer Errors* was the most common type of vulnerability delayed, occurring in 4/5 sources. Furthermore, *Improper Access Control*, *Information Leak Disclosure*, *Cross-site Scripting(XSS)*, and *Input Validation* tied as the second most common type of vulnerability delayed, appearing in 3/5 sources. *Permissions, Privileges, and Access Control* and *Resource Management* were the third most common type of vulnerability delayed, appearing in 2/5 sources. Therefore, NVD should work towards improving disclosures of these types of vulnerabilities.

5.4. Disclosure delays based on time-frame (days)

Figure 5 presents the top 5 disclosure delays based on their time-frame (days).

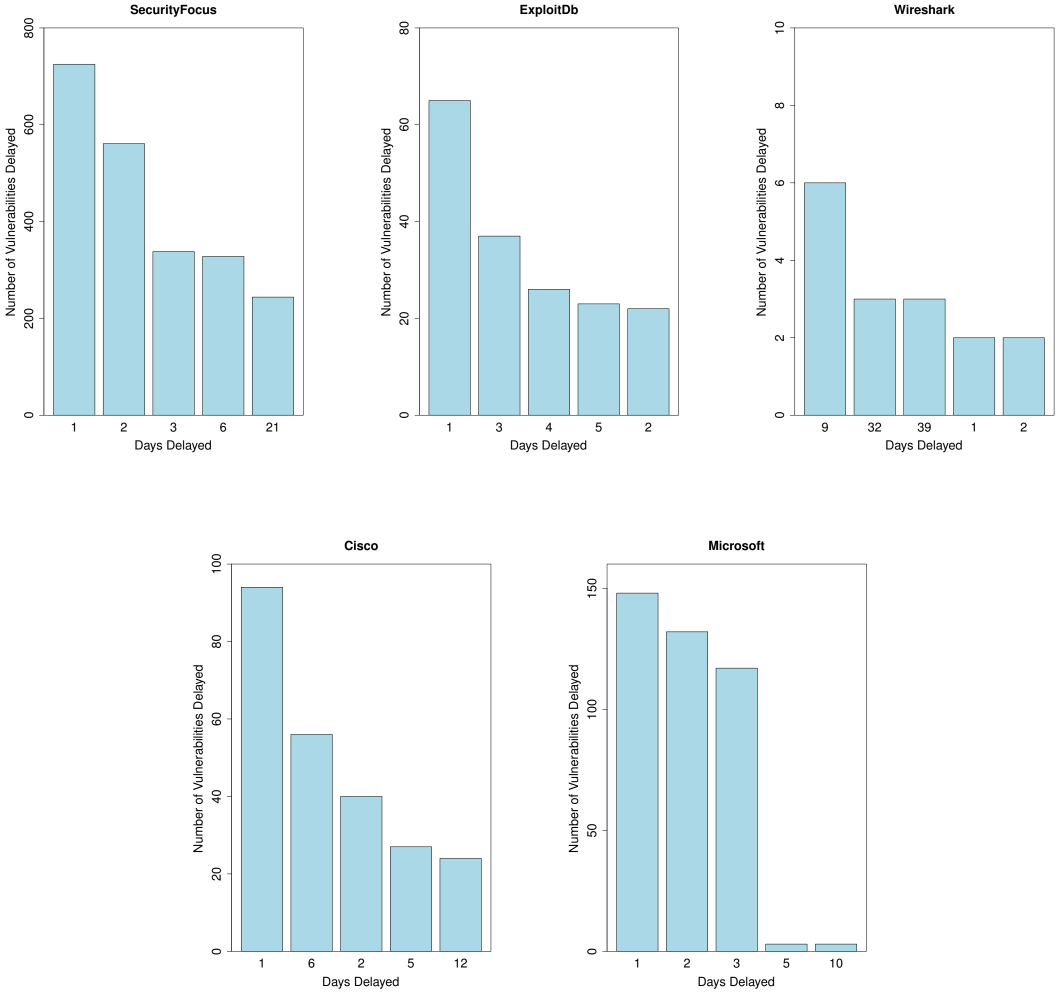


Figure 5. Disclosure delays from NVD based on days.

SecurityFocus The majority of vulnerabilities (18.24%) had a disclosure delay of 1 day. Approximately 14.12% of vulnerabilities had a disclosure delay of 2 days. Furthermore, 338 (8.51%) , 328 (8.26%), and 244 (6.14%) vulnerabilities were delayed for 3, 6, and 21 days, respectively.

ExploitDb 24.71% of vulnerabilities had a disclosure delay of 1 day. 14.07%, 9.89%, 8.75%, and 8.37% of vulnerabilities were delayed by 3, 4, 5, and 2 days, respectively. It is worth noting that all of these vulnerabilities were delayed within 1 week.

Wireshark The majority of vulnerabilities (12%) had a disclosure delay of 9 days, a stark contrast from the aforementioned sources. Furthermore, the second and third highest number of vulnerabilities had a disclosure delay of 32 and 39 days, respectively. However, the fourth and fifth highest number of vulnerabilities had a disclosure delay of 1 and 2 days respectively. These results confirm the urgency for improving vulnerability disclosures from NVD.

Cisco 32.30% of vulnerabilities had a disclosure delay of 1 day. 19.24%, 13.75%, 9.28% and 8.25% of vulnerabilities had a disclosure delay of 6, 2, 5 and 12 days, respectively.

Microsoft The majority of vulnerabilities (36.45%) had a disclosure delay of 1 day. 32.51%, 28.82% and 0.74% had a vulnerability disclosure delay of 2, 3 and 5 days respectively. Approximately 0.74% of vulnerabilities had a disclosure delay of 10 days.

The analysis of these sources reveals that most vulnerabilities are delayed within 1 day. Although the majority of vulnerabilities have disclosure delays within a week, greater emphasis should be placed on reducing this time-frame.

5.5. Answers to the research questions

Based on the results, we can answer the research questions presented in Section 3:

- **Q1** How many days transpire between vulnerability disclosures in other sources and NVD publications? **Answer:** Based on our results, the majority of vulnerabilities are delayed within 1-7 days.
- **Q2** Are other sources more efficient for disclosing vulnerabilities? If so, why? **Answer:** Yes. NVD does not disclose vulnerabilities in real-time, consequently providing a disadvantage to organizations that depend on it. Furthermore, other sources do not depend on CVE and; thus, disclose vulnerabilities faster.
- **Q3** Which vulnerability rating is disclosed the most? **Answer:** Based on our results, Medium and High vulnerabilities.
- **Q4** Which types of vulnerabilities published in the NVD have longer publication delays? **Answer:** Based on our results, buffer errors, improper access controls, information leak disclosure, cross-site scripting, and input validation.

It is worth highlighting that NVD states[4] that their database is updated whenever a new vulnerability is added to the CVE dictionary and; after that, NVD analysts add further information of the vulnerability within 2 days, excluding federal holidays. We believe that vulnerability disclosures should not be centralized on a specific database. A

[4] https://nvd.nist.gov/general/faq (Accessed: 10/19/2017)

collaboration between various security institutions can significantly improve vulnerability disclosures. In addition, NVD should reference Twitter or other popular social media sites as their efficiency have been proven for vulnerability disclosures [Santos et al. 2012].

5.6. Proportion analysis

The null and alternative hypothesis for comparing NVD with SecurityFocus, Cisco, Wireshark and Microsoft were $H_0 = p = 0.5$ and $H_1 = p > 0.5$ respectively. Null and alternative hypothesis for comparing NVD with ExploitDB were $H_0 = p = 0.5$ and $H_1 = p < 0.5$. All hypothesis were evaluated with a confidence level $\alpha = 95\%$. Table 3 shows the p-value for each source. It is worth mentioning that n (Total number of vulnerabilities collected from 2017) and y (Total number of NVD delays) are variables with a binomial distribution.

Table 3. Proportion analysis

Source	n	y	P-value
SecurityFocus	5525	3973	$p - value = P[X > 3973 \vert p = 0.5]$ $= 1 - P[X <= 3973 \vert p = 0.5]$ $= 1 - P[X = 1 \vert p = 0.5] + P[X = 2 \vert p = 0.5] + \cdots + P[X = 3973 \vert p = 0.5]$ $= 1 - (\sim 1)$ $= 0$
ExploitDb	732	263	$p - value = P[X <= 263 \vert p = 0.5]$ $= P[X = 1 \vert p = 0.5] + P[X = 2 \vert p = 0.5] + \cdots + P[X = 263 \vert p = 0.5]$ < 0.5
Cisco	321	291	$p - value = P[X > 291 \vert p = 0.5]$ $= 1 - P[X <= 291 \vert p = 0.5]$ $= 1 - P[X = 1 \vert p = 0.5] + P[X = 2 \vert p = 0.5] + \cdots + P[X = 291 \vert p = 0.5]$ $= 1 - (\sim 1)$ $= 0$
Wireshark	50	50	$p - value = P[X > 50 \vert p = 0.5]$ $= 1 - P[X <= 50 \vert p = 0.5]$ $= 1 - P[X = 1 \vert p = 0.5] + P[X = 2 \vert p = 0.5] + \cdots + P[X = 50 \vert p = 0.5]$ $= 1 - (\sim 1)$ $= 0$
Microsoft	590	406	$p - value = P[X > 406 \vert p = 0.5]$ $= 1 - P[X <= 406 \vert p = 0.5]$ $= 1 - P[X = 1 \vert p = 0.5] + P[X = 2 \vert p = 0.5] + \cdots + P[X = 406 \vert p = 0.5]$ $= 1 - (\sim 1)$ $= 0$

Since $p - value < 0.5$ for all sources, we reject the null hypothesis and; thus, we can extend, with 95% confidence, these statements for future observations or vulnerabilities.

6. Recommendations

Our web parsers for SecurityFocus and ExploitDb use official reference maps to retrieve information concerning vulnerabilities, being `http://cve.mitre.org/data/refs/`

refmap/source-BID.html and http://cve.mitre.org/data/refs/refmap/source-EXPLOIT-DB.html , respectively. The remaining web parsers collect vulnerabilities from blogs. Therefore, these web parsers do not need to be modified to collect future vulnerabilities. All of the chosen sources are referenced by NVD in their Data Feed, thereby offering trust and simplicity.

We recommend using our web parsers because they are flexible and simple for users. Extending these parsers will aid in providing a more detailed analysis of disclosure delays. Furthermore, additional web parsers can be developed by simply modifying the reference map and code, thereby retrieving information from a different source.

7. Conclusions

This paper showed NVD vulnerability disclosure delays in order to provide a detailed analysis concerning its efficiency, which is useful for IoT developers and manufactures. Web scrapers for each source were developed, being flexible for future projects. Related works place large emphasis on vulnerabilities disclosed in the dark web. We compared NVD with well-established sources. The results showed that these sources managed to surpass NVD in terms of vulnerability disclosures, thereby confirming the urgency to improve the former.

8. Acknowledgments

We offer our sincere gratitude to UTFPR and CAPES. This paper would not have been possible without their support.

References

[Best 2011] Best, C. (2011). Challenges in Open Source Intelligence. In *Proceedings of the European Intelligence and Security Informatics Conference*, pages 58–62.

[Bhardwaj and Kole 2016] Bhardwaj, S. and Kole, A. (2016). Review and Study of Internet of Things: It's the Future. In *Proceedings of the International Conference on Intelligent Control Power and Instrumentation*, pages 47–50.

[Bhuddtham and Watanapongse 2016] Bhuddtham, T. and Watanapongse, P. (2016). Time-related Vulnerability Lookahead Extension to the CVE. In *Proceedings of the JCSSE*, pages 1–6.

[Fang and Hafiz 2014] Fang, M. and Hafiz, M. (2014). Discovering Buffer Overflow Vulnerabilities in the Wild: An Empirical Study. In *Proceedings of the International Symposium on Empirical Software Engineering and Measurement*, pages 23:1–23:10.

[Gharaibeh et al. 2017] Gharaibeh, A., Salahuddin, M. A., Hussini, S. J., Khreishah, A., Khalil, I., Guizani, M., and Al-Fuqaha, A. (2017). Smart cities: A survey on data management, security, and enabling technologies. *IEEE Communications Surveys Tutorials*, 19(4):2456–2501.

[Guojun et al. 2017] Guojun, Z., Wenchao, J., Jihui, S., Fan, S., Hao, Z., and Jiang, L. (2017). Design and Application of Intelligent Dynamic Crawler for Web Data Mining. In *Proceedings of the YAC*, pages 1098–1105.

[Hurlburt 2017] Hurlburt, G. (2017). Shining Light on the Dark Web. *Computer*, 50(4):100–105.

[Joshi et al. 2013] Joshi, A., Lal, R., Finin, T., and Joshi, A. (2013). Extracting Cybersecurity Related Linked Data from Text. In *Proceedings of the IEEE International Conference on Semantic Computing*, pages 252–259.

[Kolias et al. 2017] Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. (2017). DDoS in the IoT: Mirai and Other Botnets. *Computer*, 50(7):80–84.

[Lee and Shon 2016] Lee, S. and Shon, T. (2016). Open Source Intelligence Base Cyber Threat Inspection Framework for Critical Infrastructures. In *Proceedings of the Future Technologies Conference*, pages 1030–1033.

[Macdonald et al. 2015] Macdonald, M., Frank, R., Mei, J., and Monk, B. (2015). Identifying Digital Threats in a Hacker Web Forum. In *2015 IEEE/ACM ASONAM*, pages 926–933.

[Mahto and Singh 2016] Mahto, D. K. and Singh, L. (2016). A Dive into Web Scraper World. In *Proceedings of the International Conference on Computing for Sustainable Global Development*, pages 689–693.

[Mittal et al. 2016] Mittal, S., Das, P. K., Mulwad, V., Joshi, A., and Finin, T. (2016). Cybertwitter: Using Twitter to Generate Alerts for Cybersecurity Threats and Vulnerabilities. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 860–867.

[Recorded Future 2017] Recorded Future (2017). The Race Between Security Professionals and Adversaries. https://www.recordedfuture.com/vulnerability-disclosure-delay/. [Online; accessed 23-March-2018].

[Santos et al. 2012] Santos, L. A. F., Campiolo, R., Gerosa, M. A., and Batista, D. M. (2012). Analysis of Security Messages Posted on Twitter. In *Proceedings of the Brazilian Symposium on Collaborative Systems*, pages 20–28.

[Sinanovic and Mrdovic 2017] Sinanovic, H. and Mrdovic, S. (2017). Analysis of mirai malicious software. In *Proceedings of the International Conference on Software, Telecommunications and Computer Networks*, pages 1–5.

[Washington Post 2017] Washington Post (2017). Hackers hit D.C. police closed-circuit camera network, city officials disclose. https://www.washingtonpost.com/local/public-safety/hackers-hit-dc-police-closed-circuit-camera-network-city-officials-disclose/2017/01/27/d285a4a4-e4f5-11e6-ba11-63c4b4fb5a63_story.html?utm_term=.09d45f71c953&tid=a_mcntx. [Online; accessed 19-march-2018].

[Yang and Lee 2012] Yang, H.-C. and Lee, C.-H. (2012). Mining Open Source Text Documents for Intelligence Gathering. In *Proceedings of the International Symposium on Information Technologies in Medicine and Education*, volume 2, pages 969–973.

[Younis and Malaiya 2015] Younis, A. A. and Malaiya, Y. K. (2015). Comparing and Evaluating CVSS Base Metrics and Microsoft Rating System. In *Proceedings of the International Conference on Software Quality, Reliability and Security*, pages 252–261.

[Zegeye and Sailio 2015] Zegeye, L. and Sailio, M. (2015). Vulnerability Database Analysis for 10 Years for Ensuring Security of Cyber Critical Green Infrastructures. In *AFRICON 2015*, pages 1–5.