UDC 004.056:004.77

K. LOBACHOVA[1], V. KHARCHENKO[2]

[1]*National Tavrida University,Simferopol, Ukraine*
[2]*National Aerospace University Named After M.E. Zhukovsky, Kharkiv, Ukraine*

ASSESSING SOFTWARE VULNERABILITIES AND RECOVERY TIME USING OPEN RESOURCES: ELEMENTS OF TECHNIQUE AND RESULTS

Modern vulnerability resources are considered, their content, security and recovery time of different software products are analyzed. To make the process smoother and more efficient three main stages are distinguished: source overview, general security and severity analysis, and more detailed vulnerability consideration including recovery time. The proposed approach is universal and can be used for almost any software projects and systems.

vulnerability resources, security assessment, severity, recovery time

Introduction

The number of discovered vulnerabilities is increasing every day compelling software companies to be always on the alert to provide immediate protection. It is particularly important for web servers since they have always been very attractive targets for malicious attacks [1]. The term "zero-day exploit" as well as "zero-day protection" has gained widespread popularity as it often refers to extremely effective attacks that are really difficult to defend against. However, according to the CERT Institute, 99% of network attacks leverage known vulnerabilities [2]. The situation is especially critical when it comes to security problems as nowadays security is considered one of the essentials of good computer systems. And that is why different security vulnerability groups, researchers, databases and testing tools have been actively growing and developing in recent five years.

In this paper we examine several most sophisticated vulnerability resources, analyze their structure and content and use the provided information to compare several world-popular products in terms of failures, severity and recovery time (here by failure we mean vulnerability report). Our focus is primarily on the security issues as they are at the heart of all system threats and violations. But of course non-security bugs shouldn't be neglected and in our future works we are going to take them into consideration as well.

The paper is structured into three main sections. Section 1 provides a brief overview of the existing vulnerability resources, Sections 2 and 3 present the key contribution of this paper – content processing and security analysis based on the obtained data, where Section 2 describes the main approach and provides general vulnerability and severity assessment while Section 3 works with more specific information and presents more detailed vulnerability results. All the presentations are given using Apache and IIS web servers as an example. The paper ends with a summary, concluding remarks and statement of possible future work.

Overview of software vulnerability data resources

CSI/FBI (Computer Security Institute) study found that 90% of the respondents were impacted by security breaches. Of those who suffered a breach, 70% said the breaches were serious resulting in theft of proprietary information, financial fraud or sabotage of their data or networks [2]. And all these problems could have been avoided if people had taken proactive steps to eliminate the multitude of already discovered and published system vulnerabilities. There are number of resources specially designed to help identify and solve the known security problems before a hacker takes advantage of them. CVE, NVD, Secunia, SecurityFocus, OVAL, CERT – it is not a complete list of such vulnerability

channels. Below we review the most popular of them, used in further vulnerability analysis.

CVE. Maintained by Mitre, CVE is not a standalone database but rather a dictionary of standardized names for vulnerabilities and other information security exposures. CVE aims to standardize the names for all publicly known vulnerabilities and security exposures. Its goal is to make easier to share data across separate vulnerability databases and security tools. The content of CVE is a result of collaborative work of various security experts; the resource is free and funded by US government. Absolute majority of all vulnerability resources are based on and synchronized with the CVE vulnerability naming standard.

NVD. The National Vulnerability Database (NVD) is one of the most sophisticated vulnerability channels that integrates all publicly available U.S. government vulnerability resources. It contains more that 22,000 vulnerability entries with about 22 new ones added every day. NVD is based on CVE and synchronize all its updates with CVE dictionary. Severity scores are assigned using the Common Vulnerability Scoring System standard. NVD's vulnerability information is available for free to the public as an xml feed.

Secunia. A Danish commercial organization, that provides internet security services and offers software products for tracking computer viruses and security vulnerabilities. The Secunia research staff test, verify, validate and assess public vulnerability reports, also conducting their own research in various products. The discovered vulnerabilities are reported to the vendors who issue updates and actively co-operate with the Secunia Research team.

The resource also provides a free web based information database with a good search that can yield the list of particular product vulnerabilities, show their severity range and indicate whether the patches are available, draw diagrams and provide other visual information representation.

Security failure analysis. Elements of analysis technique

Although many trusted security data recourses exist, we decided in favor of National vulnerability database (NVD) due to its large vulnerability base and well-structured data provided in xml format. For our analysis its content was parsed and CVE vulnerability indices, publish dates and severity scores for each particular product were picked out. Then product security characteristics were compared in terms of the number of discovered vulnerabilities and total severity rates, the information was visualized and the obtained results were analyzed.

For our work two rival (free and commercial) software - Apache and IIS servers were chosen as very security-conscious and very popular products in the modern engineering. To be precise, we were interested in 1.3, 2.0, 2.2 Apache branches (2.1 is omitted as an internal development version never released officially) from “Apache Software Foundation” or “Apache.Group” vendors and 5.x (5.0 and 5.1), 6.0 IIS versions from “Microsoft” company. The retrieved data refer to “Apache”, “Apache HTTP Server” and “IIS”, “Internet Information Services” product names for Apache and IIS servers respectively.

Having parsed the database content a summary table is compiled for each software branch in the following format (Table 1).

Combining the obtained results, we get the bar graphs that illustrate the comparison of three Apache and two IIS branches (fig.1 and fig.2).

Table 1

The database content for each software branch

CVE ID	Published	Severity	CVSS_Score	Apache branch
CVE-2002-0843	11.10.2002	High	8	1.3
CVE-2003-0460	27.08.2003	Low	3,3	1.3
CVE-2003-0542	03.11.2003	Medium	4,9	1.3
CVE-2003-0987	03.03.2004	High	7	1.3

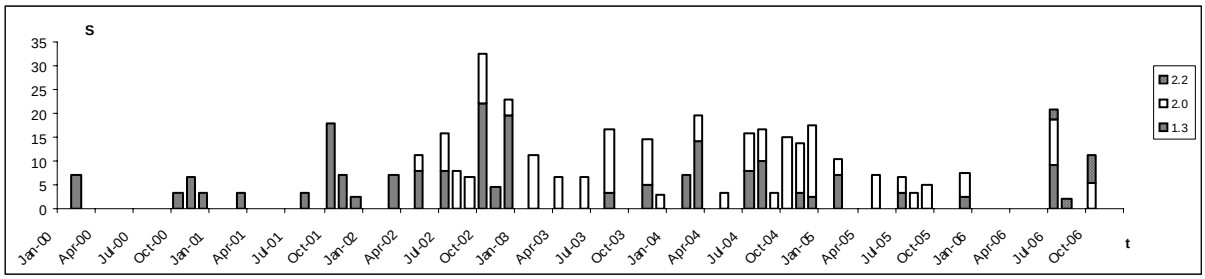

Fig. 1. Severity rate of Apache vulnerabilities calculated for each month

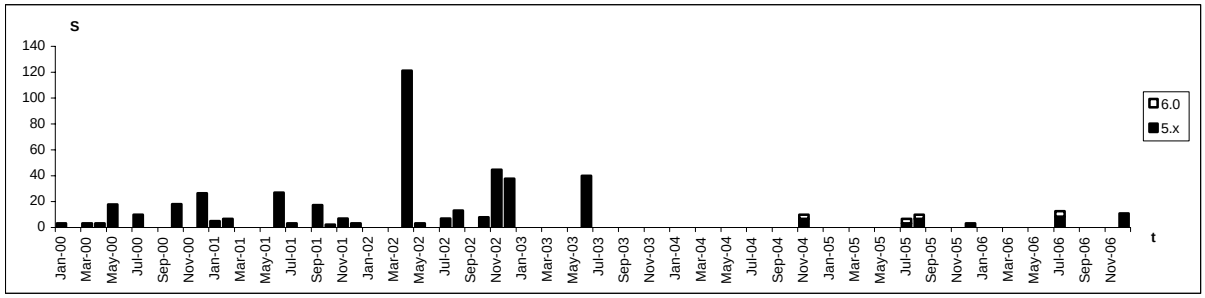

Fig. 2. Severity rate of IIS vulnerabilities calculated for each month

Graphs show total severity rates of the discovered bugs for each month from January 2000 through December 2006. The colors indicate each branch's contribution to the total severity value. It is calculated as $S_m = \sum_{n=1}^{N} S_n$, where S_m represents a total severity rate assessed within one month, N is the number of vulnerabilities found during this month and S_n is the rate of n-th severity. Proceeding from *Alhazmi-Malaiya work* [1] and fitting vulnerabilities data for each Apache branch to the time-based model, will give us the following results (fig. 3).

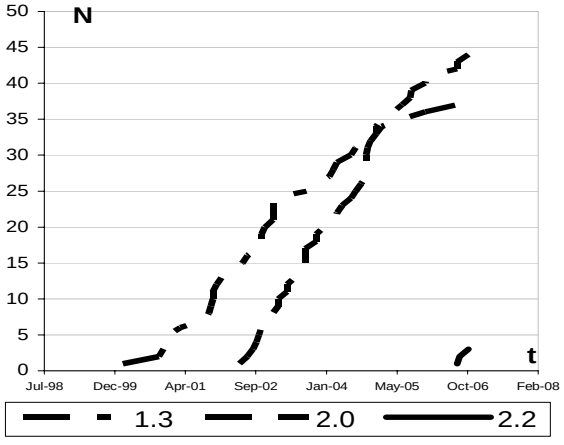

Fig. 3. Graph of cumulative number of vulnerabilities

It is clearly seen that the slope of the graph for Apache 2.0 is steeper. Steep graph shows higher vulnerability consistence than that of 1.3 version. You can read the growth rate directly from the slope of the graph: For example, in May 2002, when the first Apache 2.0 version was released, the number of Apache 1.3 vulnerabilities already made 15, but in November 2004 Apache 2.0 came up with Apache 1.3 and then outrun the old version. Now if we examine severity ranges assigned to the same vulnerabilities by different resources we can find great dissimilarity as the methods of evaluation are not identical and even not exactly comparable. Secunia provides criticality graphs on its web site so we decided to present similar illustrations for NVD database vulnerabilities (fig. 4). It is easy to see that the majority of discovered vulnerabilities, luckily, has low severity. The most serious security loopholes were disclosed in 1.3 branch though it is excusable as 1.3 is the first and the most long-lived Apache version.

In addition, it is worth noting that the great number of found loopholes are not branch-specific and pose a threat to all the current product versions. That means that source code is often reused and the existing or potential vulnerabilities pass through different software branches.

Recovery time analysis

As a rule researchers discussing product security focus their attention on the number of found vulnerabilities considering it a kind of a dependability rate. Sometimes severity score is mentioned. But to obtain a complete dependability description of a certain product, other important factors must be taken into account. The one we are going to consider is the amount of time it takes for a vendor to prepare a patch and return back to normal secure operation after a vulnerability report. We call it "recovery time" as vulnerability disclosure is very similar in its essence to system failure. On their web site eEye Digital Security company wrote that "The window of time to remediate new vulnerabilities has shrunk to just hours, compared to months in the past." We decided to investigate this question and analyze the recovery time needed for Apache and Microsoft companies to fix the reported vulnerabilities.

The time of vulnerability disclosure is defined differently in the security community and industry. Usually, vulnerability information is discussed in mailing lists and only then is approved and published as a security advisory. The time it appears in different databases often differs and this nonoccurrence can make up several years. So to get the correct discovery time information we decided to use CVE library since it is the most reliable and standardized resource other databases are based on. Patches and updates are maintained by software vendors and so the dates of fixes for our products are available in Apache Security Reports and IIS Microsoft Security Bulletins. Having all the necessary information at our disposal, we can now extend product detail tables to look like table 2.

The improved graph (Fig. 5) takes proper account of recovery time and shows the cumulative severity of all the discovered and non-fixed bugs at each period of time. The review of Apache products clearly shows that Apache 2.0 has always been the least stable branch with a lot of security loopholes while time-proved Apache 1.3 version is pretty reliable and secure. Apache 2.2 branch was released relatively recently, and as it is not yet popular there is no tangible benefit of vulnerability discovery and exploitation. IIS 5.0 proved itself to be the most stable and reliable web server over the last three years, however, you can see that it had a lot of security problems in the past.

Table 2

Extend product

CVE ID	Published (NVD)	Discovered (CVE-mitre)	Fixed (Apache report)	Recovery time (days)	Secu-nia rate	NVD rate	Apache branch
CVE-2002-0843	11.10.2002	08.08.2002	03.10.2002	56	3	8	1,3
CVE-2003-0460	27.08.2003	26.06.2003	18.07.2003	22	2	3,3	1,3
CVE-2003-0542	03.11.2003	14.07.2003	27.10.2003	105	2	4,9	1,3
CVE-2003-0987	03.03.2004	16.12.2003	12.05.2004	148	2	7	1,3

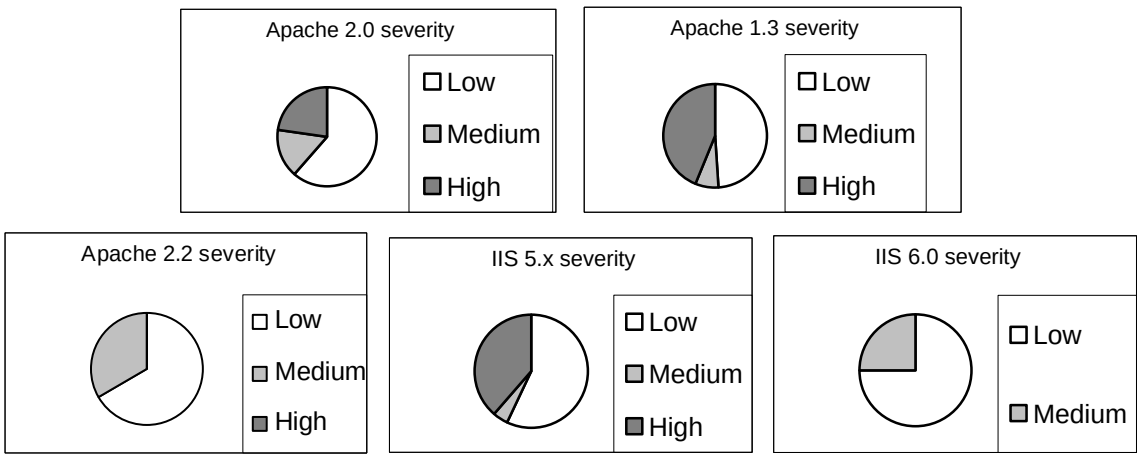

Fig. 4. Pie charts illustrating the distribution of severity groups

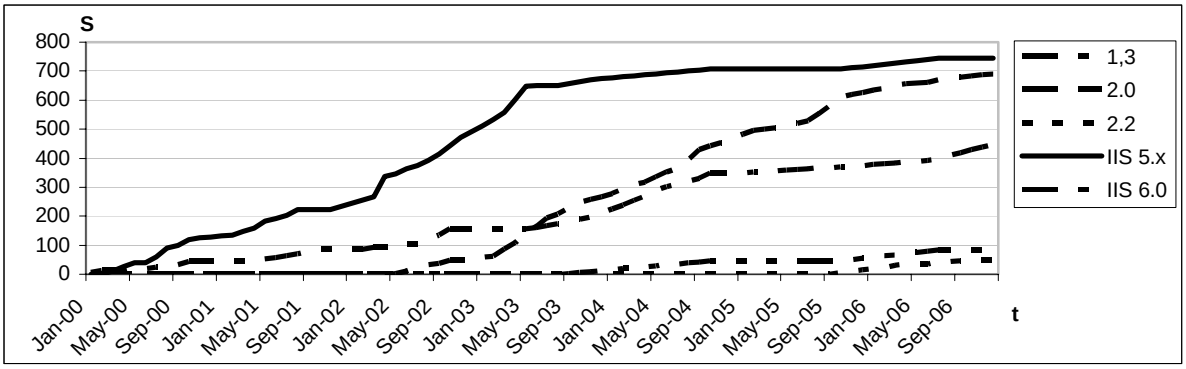

Fig. 5. Graph of cumulative vulnerability severity

Its proximate successor – IIS6 also shows good promise in terms of security.

Comparing IIS and Apache product recovery time shows that in average open Apache community takes 84 days to fix the reported vulnerability.

It is twice more than that of commercial Microsoft organization.

Conclusion

In this paper different vulnerability resources were considered to evaluate and compare product security. The approach used represents a general solution that makes comparing security vulnerabilities possible. The method is generic and can be applied to the wide range of software products. When used for Apache and IIS http servers, it demonstrates the comparison between an open source and commercial software in terms of secure operation. Both analyzed organizations have good potentialities to create reliable and secure software. It should be emphasized that proceeding with other recourses, for example [1], allows us to analyze not only the number of discovered vulnerabilities but their severity and recovery time and, therefore, to estimate the product security as a whole.

Further research is needed to evaluate the non-security issues and estimate its contribution to the whole software dependability.

The proposed technique can also be extended for service-oriented architectures and other complex software component-based systems if it is necessary to consider the system dependability as a whole.

References

1. Sung-Whan Woo, Omar H. Alhazmi, Yashwant K. Malaiya. Assessing Vulnerabilities in Apache and IIS HTTP Servers. 2006. – 240 p.

2. Network Security and Vulnerability Assessment [Електрон. ресурс]. – Режим доступу: www.eeye. com, February 2007.

3. Mitre Corp, Common Vulnerabilities and Exposures [Електрон. ресурс]. – Режим доступу: http:// www.cve.mitre.org/, February 2007.

4. Vulnerability and Virus Information [Електрон. ресурс]. – Режим доступу: http://secunia.com, February 2007.

5. National Vulnerability Database [Електрон. ресурс]. – Режим доступу: http://nvd.nist.gov/, February 2007.

6. Apache Software Foundation Bug System [Електрон. ресурс]. – Режим доступу: http://issues. apache. org/bugzilla/, February 2007.

7. An Open Online Encyclopedia [Електрон. ресурс]. – Режим доступу: http://en.wikipedia.org, February 2007.

8. Apache Security Updates [Електронний ресурс]. – Режим доступу: http://httpd.apache.org/security_ report.html, February 2007.

9. Microsoft Security Bulletin [Електрон. ресурс]. – Режим доступу: http://www.microsoft.com/technet/security/bulletin, February 2007.

Надійшла до редакції 5.03.2007

Рецензент: д-р техн. наук, проф. В.М. Илюшко, Національний аерокосмічний університет ім. М.Є. Жуковського «ХАІ», Харків.