App A

Computer vulnerability categorization

Abstract

It is a known fact that computers have vulnerabilities and weaknesses. This is made evident by the increasing international cry for advanced security controls protecting the information technology (IT) environment. The first step in curing any disease is to recognize that it exists.

Management must be acutely aware of problems regarding the protection of the organization's information assets. This means that they must know what types of vulnerabilities exist in their IT environments. Products available today in the IT market which attempt to identify problems in IT environments are categorized as either intrusion detection systems or vulnerability assessment tools. These tools present management with a detailed list of vulnerabilities in their IT environments. However, managers first need to know which major types of vulnerabilities exist. This paper proposes a model enabling the identification of vulnerability types or categories.

1. Introduction

The increased use of computers in the digital world has amplified our dependency on their proper functioning. Companies are connected through large extranets, intranets and the Internet, transporting sensitive information to different sectors of their organizations. These networks have improved storage, communication and use of information in organizations.

The dependence on these large networks has caused rising concern over the safety and security of the information being relayed across them. Organizations need assurance that their information assets are safe from prying eyes and that their information is not compromised in any way.

There are some who take advantage of this technological age by stealing information, spying and sabotaging by using their knowledge of vulnerabilities in computer technology. It is important to know what these types of

vulnerabilities are and how they impact the daily functioning of an organization.

This paper will attempt to define and classify computer vulnerabilities in such a way that an overall view of the types of vulnerabilities in any computer system is evident.

2. Computer vulnerabilities and a case study

The term *computer vulnerability* has been used quite loosely in the previous section. Some formal definition seems necessary.

Let us firstly examine the difference between weakness and vulnerability. A vulnerability always has a resolution, whereas a weakness may never have one [KNIG 00].

The problem with formulating a definition that describes computer vulnerability is that *vulnerability* means different things to different people. The Merriam-Webster Collegiate Online Dictionary gives the following definition of *vulnerability* [MERR 02]:

1) *Capable of being physically wounded*
2) *Open to attack or damage*

Defining vulnerability within the scope of a computer-based system reveals the following definitions of computer vulnerability:

"*A computer vulnerability is a flaw in the security of a computer system. The security is the support structure that prevents unauthorized access to the computer. When a vulnerability is exploited, the person using the vulnerability will gain some additional influence over the computer system that may allow a compromise of the system's integrity*." [KNIG 00]

"*Bugs in a system that enable users to violate the site security policy, are called vulnerabilities.*" [BISH 99]

According to the definitions on the previous page, it can be stated that a *computer vulnerability* is any flaw in the software or hardware of a computer which has the potential of being exploited. This exploitation may lead to the use of the computer or its resources for means other than those for which they were intended. From this point all references to *vulnerability* will imply *computer vulnerability* unless stated otherwise.

The next section describes a case study that reveals the influence and impact of a group of vulnerabilities on a certain company and that company's associates.

Case study: CCBill client information leak

CCBill use an online billing system that does credit card and online checks to determine the validity of an e-commerce transaction. They are a large institution affiliated with merchants all over the world. Their merchants use CCBill's services to verify the credit card transactions that are made when customers purchase online products from their websites.
When the customer of a merchant wishes to purchase products online, the customer is referred from the merchant home page via link to the CCBill verification page. Here, the customer gives his or her personal details and credit card number and CCBill's Fraud Screening System verifies the information validity. The customer information is transferred online to the CCBill database and stored. CCBill then prints a summary of the purchase if the customer credit check allows it [CCBI 02].

Fig. 1 is a graphical representation of all the actors in the CCBill environment from the merchant customer through to CCBill themselves.

A vulnerability was exposed in the CGI scripting that handled the merchant database at CCBill as well as FTP/SSH passwords and logins. CCBill were notified, but they only fixed the faulty CGI and did not report the vulnerability to any of their clients. The fact that the vulnerability was reported externally meant that secure information from merchant web servers might already have

been compromised at the time of the notification and this included the private customer information stored on the web servers.

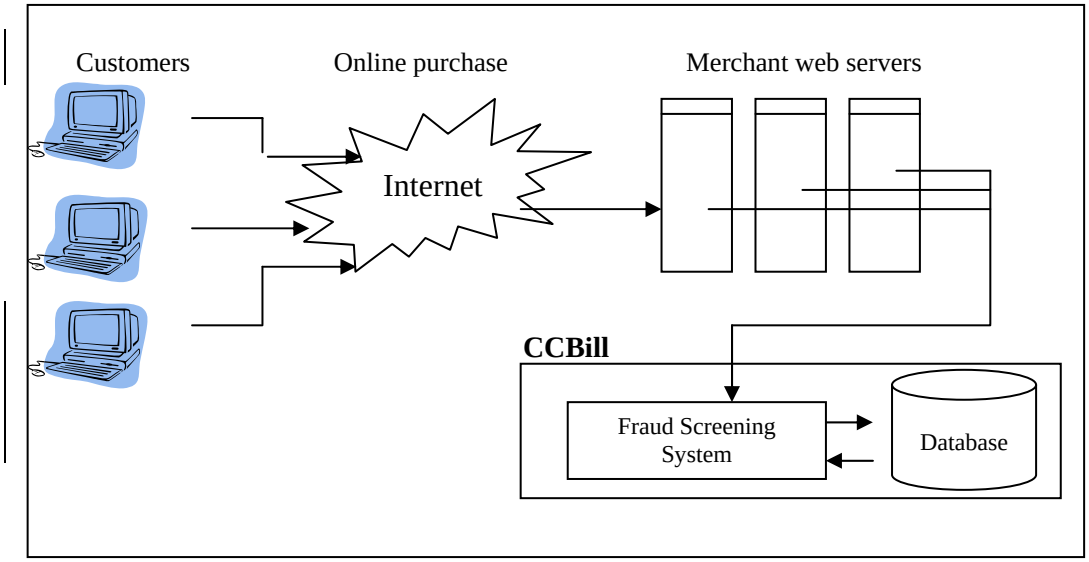

Fig. 1: Actors in the CCBill environment

CCBill chose to remain silent and hope for the best. Had CCBill notified their clients, some of the merchant customers' private details may have been saved. Some time has passed and it may be assumed that nearly all their web servers have been compromised.

This illustrates the domino effect that a single vulnerability can have. Not only was the private merchant information compromised, but the merchant customer credit card information as well [GREE 01].

3. Vulnerability classification

The case study in the previous section shows the far-reaching effect that a single vulnerability may have if left undetected. The CGI-script vulnerability mentioned is a single example of the hundreds of existing vulnerabilities.

Although it makes sense for a computer technician to have a list of the vulnerabilities that a computer system is exposed to, this list means little or nothing to IT business managers. The author suggests that management will need a detailed report of the following important factors:

- The vulnerabilities found on a computer system.
- The identification of the vulnerability categories of which they form a part.
- The cost involved in eliminating the most critical vulnerabilities to resolve overall risk.

If CCBill had some sort of classification of the vulnerabilities identified, areas they influenced and their impact, then management might have limited the effects of the disaster by making better decisions according to the bigger picture.

There are certain problems that make vulnerability classification difficult. The section that follows will examine one, namely the lack of a standard for the assignment of a unique identification to every existing vulnerability.

3.1 A standard for vulnerability identification

Hundreds of new vulnerabilities are exposed annually and nearly every computer security institution worldwide has its own identification for a vulnerability. This creates difficulties when security institutions attempt to share information and resources of this kind.

One organization that is attempting to create a standard for vulnerability identification is the Common Vulnerabilities and Exposures (CVE) group. CVE [CVE 02] propose a standard for the unique identification of all known computer vulnerabilities and are inviting other security organizations to pledge their support in creating this standard.

At the time of writing this article, they boasted 1 604 entries in their vulnerability database, which they refer to as their vulnerability dictionary. Forty-two organizations have joined their effort in creating this standard for implementation in 67 compatible products. Some of the companies associated include Symantec [SYMA 03], CERT Coordination Center [CERT 03] and Bindview Development [BIND 03] to mention but a few.

Only when such a standard exists will it be possible to categorize all known vulnerabilities. This standard will eliminate all confusion surrounding vulnerabilities publicized by different security institutions.

Assume that this standard for vulnerability identification is already in place. The categorization of computer vulnerabilities can now continue. The technique of categorization should, however, adhere to certain criteria and the author proposes that the criteria discussed in the following section be applied.

3.2 Vulnerability classification scheme

Creating a global view of the areas of vulnerability within a computer system may be accomplished in a number of ways. In one approach, the computer system itself could be divided into a number of physical areas of vulnerability. For example, vulnerable areas such as the primary memory, secondary memory and hardware peripherals may be identified.
The problem with this approach is that it is difficult to classify all individual vulnerabilities into these few areas of vulnerability. Also, deciding which area of vulnerability an individual vulnerability belongs to is problematic, as an individual vulnerability may influence more than one area of vulnerability.

Another method of categorization is by shifting attention to the vulnerability itself and examining its nature. Bishop [BISH 99] states that vulnerabilities should be categorized according to the following criteria:

- Vulnerabilities of **similar nature** should be grouped together. For example, all vulnerabilities which cause the denial of some sort of service to a user when exploited should be grouped together.
- The classification should be **atomic.** A vulnerability may not fall in two different vulnerability classes and is either in a class or not. Therefore the conditions of classification should be well defined.
- Classification should **not** be based on the **social cause** of the vulnerability. This includes issues such as motive, intent and malicious or

accidental cause. Classification should rather take place according to technical details such as environment and field of influence.

The author accepts these categorization criteria on the grounds that vulnerability categorization is simplified and more accurate through their implementation.

Adhering to the classification scheme presented above and through the examination of the literature sources ([KNIG 00], [DAVI 99], [LONG 97], [CYBE 01], [HOWA 01], [REAL 01], and [VENT 02]), this paper proposes **four** main vulnerability categories and a further **twelve** subcategories. All the sources listed above have endeavoured to identify the main types of computer vulnerabilities and the author of this paper has compiled a combination of these classification examples.

3.3 Vulnerability categories

A summary of the four main vulnerability categories and twelve subcategories into which vulnerabilities may be classified is shown in *table 1*. Every main category has three subcategories associated with it and every subcategory has a number of vulnerabilities linked to it.

For example, the Misconfiguration subcategory may contain 41 vulnerabilities and the Hardware specific subcategory contains 12, but ultimately all these vulnerabilities fall under the Logic errors main category.

Each category name, be it main or subcategory, is a description of the nature of the vulnerabilities in that category.

Table 1: Computer vulnerability categories

Main categories	Subcategories
Logic errors	Misconfiguration Software specific and updates Hardware specific

Principle violations	Security policy violations User privilege User enumeration and information
Security violations	Back doors, Trojans and remote controls Spoofing or masquerading Denial of services and buffer overflows
Weakness assessment	Password sniffing and encryption Network and system information gathering Unauthorized access to remote connections

3.3.1 Logic errors

These are errors in design or implementation of software packages as well as hardware products which cause a breach in computer system security and may lead to compromise of the system.

To be competitive in the open market, computer organizations need to deliver products quicker than any of their competitors. This means that some products are exposed to the market with flaws in programming and design and released before they have been thoroughly tested for stability. The end-users of the product or service are usually the ones who make the nasty discovery that something in their software or hardware is not functioning as it should. These errors are usually known as “bugs”.

The following three subcategories are part of the Logic errors main category:

a) Misconfiguration

Upon installing any new software on a computer system, whether the system is a server or a normal workstation, it must be assumed that the security of the software has its default settings in place. This means that the security of the computer system may be compromised because the software has not been configured to an optimum security level.

For example, assume a firewall is in place. If the company security policy forbids any connection to unknown Internet websites and the firewall has not been configured to block such connections, unsuspecting users may visit untrustworthy websites and this may lead to the downloading of malicious

software and code. The security structures are in place, but the software is not configured correctly.

Misconfiguration usually happens as a result of inexperienced users or new, untested software packages. Thus all computer vulnerabilities of this nature are added to this subcategory.

b) Software specific and updates

The vulnerabilities in this subcategory are all related errors in code, design and programming of existing software packages. As mentioned earlier, some software is released without thorough testing, although it may be argued that testing will not reveal all errors that appear under unexpected circumstances.

These “bugs” are the most dangerous vulnerabilities in the market because of their lack of predictability and the time they take to manifest. When the user becomes aware of the problem, it is already too late to do anything about it. They are so rampant that most software packages require the use of regular patches to eliminate vulnerabilities identified by users.

For example, Microsoft Windows NT release service packs to counter the effects of vulnerabilities that users have identified in the operating system.

c) Hardware specific

Hardware vulnerabilities are more easily recognizable than other vulnerabilities. It is usually quite obvious when some hardware product is not functioning properly, and the only way of rectifying the situation is by replacing the faulty machinery.

Hardware products, such as printers and routers, have some software running in the background to establish communication and proper functioning. This software may not be very useful to the average user because of the low level of programming used, but if this software is malfunctioning it may compromise the system and reinstallation may be necessary.

For example, if there is a vulnerability in a router that relays network communication, then malicious users may use it to attack different sectors in a network if they become aware of the error.

3.3.2 Principle violations

The security principles of an organization refer to the organization's point of view concerning certain security issues. These issues range from locking office doors at night to the size and strength of passwords used when logging into a computer system. The security principles encompass the concepts of security policies, procedures and standards. A company's security policy is the company's documented strategy on dealing with security issues [CHOL 97]. While the security policy defines the rules concerning security issues, procedures list how the policies are enforced. Security standards are widely accepted, fixed practices incorporated by the organization to ensure security.

Sound security principles are important in that they create security awareness and measures used for countering security threats. If a company's security principles are thorough and all users are aware of it, a secure environment is created within the organization. Any violations in these security principles may lead to weakened security and vulnerabilities. Although companies address their security issues in different ways, there is common ground. For example, the website security of a company that draws its lifeblood from an e-commerce website is radically different from one that has a normal information website, although both have some sort of viewpoint concerning their Internet websites.

The following subcategories form part of the Principle violations main category:

a) Security policy violations

This vulnerability subcategory holds all vulnerabilities deemed to be in violation of the documented security policy. This is quite a broad area to cover, and all vulnerabilities that do not fall in the other two subcategories of Principle violations are placed here. This section may differ from company to company, depending on their point of view.

For example, one company may feel it is too dangerous for users to download CGI-scripts because of the security risk they pose. The security policy would relate the download prohibition. If CGI-scripts are downloaded contrary to the

security policy specification, it may be considered a vulnerability. Another company may feel the security threat they pose is acceptable and will allow CGI-script downloading. The oversight in having no point of view in connection with the CGI-script may also be seen as a vulnerability.

b) User privilege

The privileges of users in an organization refer to the levels of computer resource access the users enjoy. User account procedures should be defined within the company and these procedures should specify how user accounts should be configured and which user accounts have higher levels of computer resource access and which have lower levels. If these procedures are not in place, the configuration of user account settings becomes problematic in the sense that accounts that need more access are given less and accounts that should have lower access levels are configured to a higher level. This creates lower overall system performance and a definite security risk through incorrect user account privileges.

For example, if a company security procedure stipulates that the guest account login on a computer system has only read privileges and a user is able to execute write commands, the account configuration may be considered a vulnerability.

c) User enumeration and information

There are certain software tools that enable users to gather information about other users connected to the network. This subcategory should not be confused with the Network and system information gathering subcategory, because in this instance, information gathered is concerned only with users.

The reasons for using such software tools are not always malicious, but intruders may use the information gathered from such a tool differently from a normal user. The intruder may be searching for the weakest link to attack in the network.

This does not mean that the tool itself is a threat and should not be used. It simply means that it should be installed and used with caution given the fact that it may be implemented to do harm. In this case, a certain security standard should dictate the implementation of such tools.

For example, the “finger” command on a computer system gives a list of all users connected to the network.

3.3.3 Weakness assessment

Weakness assessment can be seen as the preparations made by an intruder before attacking a certain computer system or network of computer systems. It is therefore a systematic approach to determine the weakest point in the security structure of a system or network of systems [KNIG 00].

The main actors in the Weakness assessment vulnerability category are users. Users demand more convenience and an easy-to-use environment for their computing experience. This can have a detrimental effect on the security structures in place.

For example, weak passwords are used extensively. A certain company was warned to change its passwords frequently. Now it changes its passwords monthly, but has opted to give everyone in the company the same password [KRUL 01].

One of the main differences between weakness assessment and a logic error is that weakness assessment occurs over a longer period than the more instantaneous logic error. The following subcategories are related to the Weakness assessment category:

a) Password sniffing and encryption

Passwords, when implemented correctly, can be a very difficult security barrier to break. It is possible to crack passwords through use of brute force because of the different cracking tools on the market and the speed of modern computer processors, but it still takes time to accomplish. Thus the stronger the password, the longer it takes to break and this increases the chances of the intruder drawing attention.

Other software tools, called sniffers, listen to network traffic and try to intercept passwords and messages [KLAU 96]. It is therefore critical that all password transfer, as with sensitive information, be strongly encrypted.

Weak passwords and unencrypted communication of sensitive material are examples of vulnerabilities that are categorized under the Password sniffing and encryption subcategory.

b) Network and system information gathering

The more information a potential intruder can gather from the computer system or the network of computer systems that he or she is trying to infiltrate, the better. The type of operating system and the services currently running on a computer system are examples of such information. Scanning tools are used to do this information gathering and when the intruder has gathered all he or she needs on the target, he or she may begin the attack on some area of weakness.

For example, a port scanner may be used to determine the open ports on a computer system. The intruder may try to connect to an open port if he or she knows the type of service running on the port.

Assume the port uses an RPC (Remote Procedure Call) service. The intruder may then use this service to gain access to and control over the computer system by executing commands through the RPC service.

These open ports and services are examples of vulnerabilities in this subcategory.

c) Unauthorized access to remote connections

This vulnerability subcategory includes examples of connections to remote hosts that are in violation of the security policy specification. Although the listing of all acceptable and unacceptable connections in a security policy is impractical, some sort of broad classification of what is satisfactory and what is not should be stipulated.

For example, assume that a company security policy prohibits connection to FTP (File Transfer Protocol) server sites. Connection to such a site may be seen as a vulnerability according to the security policy.

3.3.4 Security violations

Security violations can be seen as any measures employed, such as the use of lies and deceit as well as engineered tools, to interfere with system availability, data integrity or data confidentiality [RADA 97].

Hackers and crackers use stolen information and software tools to gain access to higher levels of security and, depending on their motive for intrusion, either steal or sabotage valuable information sources. The author suggests that there are three ways in which intruders can undermine system availability and interfere with data integrity and confidentiality. The three subcategories below describe these methods and form part of the Security violations main vulnerability category.

a) Back doors, Trojans and remote controls

Vulnerabilities in this subcategory are related to a computer's susceptibility to attack from software tools that have been created by shrewd programmers bent on breaching data confidentiality and integrity.

- Trojans are pieces of software code connected to services running on a computer system that seem harmless. However, the Trojan performs some malicious task in the background of which the user is unaware [CHAN 01]. The task is usually malicious in nature.
- Back doors are secret points of entry created by programmers in normal software packages that can be used to gain unauthorized access to the computer system on which they are implemented [ZHAN 98].
- Remote controls are tools that give an intruder total remote control over a system upon which the control has been installed. An example of a remote control is Netbus Pro, which gives the intruder control over a computer system and its resources [KUCA 98].

b) Spoofing or masquerading

To spoof or to masquerade means to hide one's identity or pretend to be someone else. This includes impersonation over the phone to mislead a person into giving away sensitive information.

When attackers wish to penetrate a computer system or network of computer systems, they would prefer their origin of attack and identity to remain unknown. There are ways of tracing the origins of an attack and if the intruders' location became known, the authorities may apprehend them.

An intruder hides his or her true identity by attacking from one or more previously compromised systems. This leads the attack trace on a wild goose chase to the compromised system and the true identity of the intruder remains unknown.

Spoofing and masquerading are often used prior to or during denial of service attacks. All vulnerabilities in this subcategory are potential spoofing or masquerading mechanisms.

c) Denial of services and buffer overflows

The vulnerabilities in this subcategory have the inherent trait of the denial of some sort of computer system resources or service. A denial of service attack or buffer overflow will usually cause the computer system that has been attacked to "crash" or "freeze".

There are many ways of causing such a system crash, but in the simplest form, a system bombarded with too many requests of some kind will stall because its buffers are filled to capacity and cannot cope with all the requests.

A prominent example of denial of service is a ping flood. A ping flood means a system has been sent so many ping requests that it cannot handle the influx and the system is rendered useless, until it has been restarted. All vulnerabilities that are susceptible to such buffer overflows and denial of service attacks fall into this subcategory of classification.

4. Conclusion

This paper has discussed computer vulnerabilities and has identified the need to categorize these vulnerabilities. These vulnerability categories reveal the main areas of vulnerability in a computer system with a universal perspective in mind.

Four main vulnerability categories and twelve subcategories were identified. Individual vulnerabilities are assigned to each subcategory while three

subcategories are assigned to a main category. This completes the global perspective of the areas of vulnerability in a computer system, which was the purpose of this discussion.

5. References

[BISH 99] Bishop M.; 1999; Vulnerability Analysis; Second RAID conference; Department Computer Science; University of California

[CCBI 02] Credit Card Billing; 2002; Overview of Services; CCBill LLC Inc.; http://www.CCBill.com

[CHAN 01] Chander S.; 17 November 2001; All about Trojans; *Basic articles about virus and Trojans*; Astalavista Group; http://www.astalavista.com

[CHOL 97] Cholvy L., Cuppens F.; 1997; Analyzing Consistency of Security Policies; *18th IEEE Computer society symposium on research in security and privacy*

[CVE 03] CVE; 2003; Common Vulnerabilities and Exposures; www.CVE.mitre.org

[CYBE 01] Network Associates; 2001; CyberCop Scanner 5.5 & CyberCop Monitor; PGP Securities; www.nai.com

[DAVI 99] David J.; June 1999; Vulnerabilities Assessment - Part 1; *Vulnerability Basics*; pp. 17 – 18; Elsevier Science Ltd

[GREE 01] Greene C. T.; 21 December 2001; CCBill knew of the Credit Database breach in March; *The Register*; http://www.theregister.co.uk

[HOWA 01] Howard B., Paridaens O., Gamm B.; 2001; Information Security: Threats and Protection mechanisms; *Alcatel Telecommunications review*; 2nd Quarter

[KLAU 96] Klaus C.; July 1996; Sniffers – FAQ's; Internet Security Systems; Atlanta Georgia

[KNIG 00] Knight E.; 8 March 2000; Computer Vulnerabilities; *Security Paradigm*

[KRUL 01] Krull R. A.; October 2001; The Lighter Side of Information Security; *Computer Fraud and Security*; ISSN 1361-3723

[KUCA 98] Kucan B.; November 1998*; Interview with Carl-Fredrik Neikter*; Helpnet Security; http://www.net-security.org

[LONG 97] Long H.; 1997; A Comprehensive Approach to Building a Secure Networking Environment; Lucent Technologies, Bell Lab Innovations

[MERR 02] Merriam-Webster Inc.; 2002; Merriam Webster Collegiate Online Dictionary, *Franklin Electronic Publishers*

[RADA 97] Radatti P. V.; May 1997; Mister Mean the Hacker: Social Engineering Systems Administrators on the Internet; Cybersoft; http://www.cybersoft.com

[VENT 02] Venter H.S.; May 2003; A model for vulnerability forecasting; Department of Computer Science; University of Pretoria

[ZHAN 98] Zhang Y.; February 1998; Detecting Backdoors; Department Computer Science, Cornell University; Ithaca NY