



White Paper

nCircle IP360 Vulnerability Scoring System

Version 1.2

nCircle Network Security
685 Market St, Suite 300
San Francisco, CA 94105
Phone: 415-625-5900
Fax: 415-625-5982
Email: info@ncircle.com

TABLE OF CONTENTS

1.0	<i>Background and Motivation</i>	3
1.1	Current Models of Risk Management	3
1.2	Observations	4
1.3	Explanation of Data	6
1.4	An Overview of Vulnerability Scoring	6
1.5	Definition of Risk and Vulnerability	7
1.6	Vulnerability Analysis.....	7
1.7	Threat and Risk Assessment.....	7
2.0	<i>nCircle IP360</i>	8
2.1	Comments on Vulnerability Scoring.....	8
2.2	Heuristic Approach to Estimating the Penetrability of a Network	8
2.3	Vulnerability Scoring.....	9
2.4	Breakdown of the Vulnerability Score	9
3.0	<i>Comparative Risk Assessment</i>	11
3.1	Calculating the Vulnerability of a Single Resource.....	11
3.2	Calculating the Vulnerability of a Network	11
3.6	Calculating Average Vulnerability	12
3.7	Calculating Average Vulnerability per System.....	12
3.8	Identifying High Vulnerabilities.....	12
4.0	<i>Logical Consequences of the IP360 Scoring System</i>	12
4.1	An Overview of Exploit Containment.....	13
4.2	Exploit Eradication.....	13
4.3	Six Common Vulnerabilities	1
	<i>End Notes and References</i>	14

1.0 Background and Motivation

When enterprises invest time, effort, and tools in tightening network security, they need to know what is at greatest risk, what problems to fix first, and how to address those problems. Unfortunately, the tools for vulnerability assessment do not make it easier to prioritize and fix security problems. After a security audit, in-house security staff are left with a list of issues ranked low, medium, and high, and from that they must perform damage control, installations, and endless upgrades. The amount of work following in the wake of a routine security audit is substantial, and it usually takes a company several months to make all of the necessary changes.

Security is a strategic discipline. Enterprises should set criteria specific to their priorities and needs to aid security personnel in making important decisions quickly. These criteria provide a road map for determining which types of vulnerabilities pose the greatest threat to the network. Furthermore, the criteria should give a clear indication of the margins of risk associated with delays in the repair process.

There are currently many methods available to help companies perform risk analysis. It will be argued in this paper that common industry models of “risk assessment” are inherently ambiguous and unhelpful in real world situations. To address this problem, an alternative model of risk assessment is offered and explained in detail. The model is presented as an “open-source” method of risk analysis – anyone who finds the method useful is encouraged to utilize it.

This paper:

1. Details the most commonly used method of assessing risks and vulnerabilities.
2. Discusses the shortcomings of the most common vulnerability “scoring” model.
3. Presents an alternative method that addresses the limitations in the vulnerability scoring techniques in other software and service offerings.

1.1 Current Models of Risk Management

A “risk analysis model” is a set of criteria that aid security personnel in performing two tasks:

1. Estimating the “risks” associated with specific network vulnerabilities.
2. Measuring the “threat” of various kinds of attacks against a network.

Many existing methods for risk analysis do not distinguish between these concepts. Furthermore, a variety of different “risk” and “threat” analysis methodologies are currently being employed, and the classification schemes that any two approaches use for rating the seriousness of vulnerabilities or measuring the threat inherent in certain attacks often differ widely.

Even though the present lack of standardization may not adversely affect the quality of work performed by security professionals themselves, it may have negative ramifications for companies with serious security problems. Because many companies lack an in-house team of security experts, it would be beneficial for these companies to have a fairly uniform set of risk assessment strategies on which to rely. The present lack of these standards very likely results in inefficient risk management practices on the part of many companies. However, efforts are currently underway to achieve a measure of standardization in the security industry, and there has been some work toward the development of a common framework for risk analysis.

In 1999, the National Infrastructure Protection Center (NIPC) began publishing CyberNotes, a monthly security update that includes a framework for “risk analysis.” The NIPC’s method for classifying risks is noteworthy because prior to 1999 there has been little effort to establish uniform industry standards, or “best practices” for risk diagnostics. According to their classification scheme, all attacks against a network fall into one of three risk categories: Low, Medium, or High.

Low Risks: Denial of Service (DoS) attacks are generally thought to pose a low risk to a network. Other rather innocuous methods of exploiting network resources for the purpose of information gathering are also considered low threats by many risk analysis methodologies. The NIPC defines a low risk as “A vulnerability that

provides information to an intruder that could lead to further compromise attempts or a Denial-of-Service attack.”

The definition, however, does include the following caveat: “... while the DoS attack is deemed low from a threat potential, the frequency of this type of attack is very high. DoS attacks against mission-critical nodes are not included in this rating, and any attack of this nature should instead be considered as a ‘High’ Threat.”¹ Terry Escamillia, in *Intrusion Detection: Network Security Beyond the Firewall* categorizes both local and remote Denial-of-Service attacks as “Annoying.”²

Moderate risks: Vulnerabilities that allow local or remote users to increase their privileges on a system or access confidential information such as company financial records or user passwords are usually considered moderate risks. According to the NIPC, “Any vulnerability that will allow an intruder immediate access to the system that is not privileged access,” is a medium risk.

High risks: Any vulnerability that could potentially allow a user to gain privileged access to a system is almost always regarded as a “high risk.” According to the NIPC, a high risk is “A vulnerability that will allow an intruder to immediately gain privileged access (e.g. Sysadmin, root) to the system. An example of this would be a vulnerability in which a sequence of instructions is sent to a machine by an unauthorized user and the machine responds with a command prompt.”³ Likewise, Terry Escamillia considers vulnerabilities that local users can exploit to obtain privileged access to be “serious risks,” while any weakness in a system allowing remote users to acquire root-privileges he ranks as “Disastrous.”⁴

Before examining the adequacy of current risk classification schemes, it should be noted that almost every existing method presupposes a **depth of access principle**. Attacks are assigned to a risk class on the basis of an implicit rule, which stipulates that the more access to a system an attack facilitates, the greater the risk or seriousness of the attack.

The influence of the principle upon risk assessment models is pervasive. Denial-of-Service type attacks are usually considered “low threats,” whereas clear-text password “sniffing,” which is an effective method for directly increasing one’s access to a system, is generally labeled as a “medium” threat. The most serious threats are generally considered those that facilitate the highest degree of access.

1.2 Observations

This section discusses shortcomings in the “Generic Model” of risk assessment presented in section 1.1. These criticisms of existing methods focus on no one method in particular.

The three major problem areas in the status quo of risk analysis are the subjectivity, ambiguity, and inaccuracy of existing methods. The arguments are presented in a series of three observations. This discussion begins with the terminology, which will be employed in the subsequent paragraphs.

All attacks of a specific type (e.g. all Denial-of-Service exploits) will be referred to as a “class.” With this in mind, risk assessment, then, involves two functions: inter-class and intra-class risk analysis. Inter-class risk analysis involves ranking various “classes” of attacks on a risk scale ranging from low (annoying) to high (disastrous).

Intra-class analysis involves ranking attacks *within a class* in the order of their seriousness. This involves determining which of the members of a class of attacks are more serious than others of the same kind, such as whether one denial-of-service attack is more threatening than another.

Observation one: Subjectivity. Many methods currently employ the “low,” “medium,” and “high” ranking system discussed in section 1.1. A major flaw in this approach is that these risk categories often become uninformative and subjective in practice.

The vagueness of these risk assessment schemes increases in proportion to the number of vulnerabilities that have been classified by the method. For instance, twenty exploits can be classified on a “low,” “medium,” and “high,” risk scale in a meaningful way; however, if one attempts to rank 1500 vulnerabilities, the system becomes questionable and ceases to be informative. If of these 1500 vulnerabilities, 600 of them can be exploited by Denial-of-Service attacks, it seems to be uninformative to claim that all 600 belong to the same threat class.

Most approaches to risk analysis avoid this difficulty by allowing some kinds of vulnerabilities (which would ordinarily qualify as low threats) to qualify as medium or high threats depending upon special circumstances. Some vulnerabilities are then assigned a higher-than-ordinary risk factor without regard for the depth of access principle. This is evidence that either (1) unstated principles, or (2) the classifiers' opinions are influencing the process. As a consequence, no two people are likely to be able to employ the method on a large scale and obtain the same results.

Observation two: Ambiguity. Many of the serious flaws in existing methods concern difficulties that arise when inter-class or intra-class rankings of risk are performed. Ambiguity arises when one attempts to perform detailed Inter-class threat analysis. If the system is firmly grounded on the depth of access principle then threat classes such as "low," "medium" or "high" correspond to different or degrees of levels of system access. There is no basis for claiming that one Denial-of-Service attack is a greater threat than all others, if the class of Denial-of-Service attacks, as a whole, is rated as "low" or "annoying," on a risk index.

There are two ways around this problem, and both introduce ambiguity into the system of classification. The first solution is to stipulate rules, (i.e. *ceteris paribus* clauses) which define the conditions under which an attack of class A can become associated with a higher risk class. This sort of strategy is present in the NIPC definition for "Low Risks." DoS attacks are low, unless they target mission critical servers or routers, and then they are high risks. This solution is problematic for the reason that once a large number of vulnerabilities have been assigned to risk categories usually reserved for attacks of a different class, the interpretation and coherence of the low/medium/high risk index begins to break down.

The problem arises because the of depth of access principle, the very principle upon which the low/medium/high ranking system was constructed, has itself been replaced by a myriad of complex conditional statements which determine how a new vulnerability gets ranked. When one makes an exception to the low/medium/high ranking system, this requires an explanation of how this exception deviates from the general rule and requires manual intervention, preventing the system from being automated and forcing an arbitrary methodology around the scoring system.

Furthermore, it is no longer the case that classes of vulnerabilities can be generally associated with one of the three classes of risk, because there exist a large number of rules to the contrary. This method, as a result, becomes more difficult to use and far less informative. One is forced to examine each vulnerability on a case-by-case basis, thus undermining the very purpose of having a "general" system for risk assessment.

The second solution to the problem of ambiguity is no better than the first. This solution involves separating each class into sub-classes in order to accommodate more detailed intra-class analysis. By doing this, one makes an attempt at distinguishing the more innocuous attacks from those which represent a greater threat to the vulnerable system, thus avoiding the ambiguity of having a tremendous number of vulnerabilities lumped into a risk class such as "Low" It is assumed then, that by dividing each class into subclasses one can attain a measure of intra-class granularity without having to assign any vulnerability a higher than normal risk for members of its class. For example, it is common to see the "medium" risk class sub-divided in the following way: "low/medium," "medium" and "medium/high".

The underlying problem with this approach is that each of the sub-classes is "undefined." In other words, there are no criteria, which establish the class membership conditions for a sub-class such as "low/medium." As a consequence, classifying exploits once again becomes arbitrary. Worse still, the problem that the creation of sub-classes intended to solve is merely recreated at a lower level. Carried to its logical conclusion, it is foreseeable to have 250 vulnerabilities of class "low/medium" and the question then arises concerning which of the "low/medium" risks are more serious than others of the same class. Once again, ambiguity is introduced into the system.

Observation three: Inaccuracy. Apart from the conceptual problems inherent in the model, the system itself seems to be too limited to serve as a realistic method of weighing risks to a system or network. This is true for several reasons that are discussed briefly here:

1. Most existing methods of risk assessment are not sensitive to the types of tools that are publicly available on the Internet. This is an important factor because the creation of a new tool which facilitates the exploitation of a vulnerability which was previously difficult to exploit puts all networks

vulnerable to this type of attack at a higher risk.

2. In many current methods, the risk associated with vulnerabilities is not time dependent. Because knowledge diffuses rapidly over the Internet, the risk associated with being vulnerable to any new method of attack should increase over time as knowledge of how the vulnerability can be exploited becomes more widely available.

1.3 Explanation of Data

Commonly, the information passed between the machines on a network is referred to as “data”. On a practical, day-to-day, level, most employees using a company network are concerned only with the privacy and integrity of data in this sense. Network security personnel and system administrators must be concerned, however, with a company’s data security in a broader sense. “Network data,” so understood, encompasses not only the information passed from machine to machine, but also any information about the network’s structure, composition, or configuration.

From a security standpoint, network data – in both senses of the term – should remain confidential. Any information a potential intruder or cracker can discover about a network’s topology and composition can be used as a stepping stone for compromising the existing security measures of the network. Once an intruder has bypassed these measures, he or she can get access to restricted or sensitive information, or stop the company’s mission critical operations. With this in mind, threats to network security fall into two general classes:

1. Threats to network “Data Confidentiality.”
2. Threats to network “Resource Availability.”

1.4 An Overview of Vulnerability Scoring

Before proceeding to define the general classes of threats just mentioned, it is important to distinguish between remote and local users of a system, and remote and local attacks. An internal user is someone who executes a command, starts a process, etc., on one or more hosts or servers on a company *intranet*. In other words, they are persons with interactive login access to a system on the network. By this definition, the group persons that can be described as “local users” of a network, includes:

- Employees who are using the network
- Anyone who walks up to a terminal connected to the company network and interacts with the system
- Anyone who telnets into an exposed server on the company network and establish an interactive login session with the system

An internal attack is one that requires a successful login session prior to the initiation of the attack. The following examples of attacks are considered “local” by this definition:

- An authorized user compiles and runs a binary, which exploits a vulnerability in Sendmail version 8.8.6, which fills up the disk partition where *user/bin/queue* is stored, causing the network mail server to refuse all incoming connections.
- A malicious ex-employee knows that the network firewall filters out incoming UDP traffic, so one day, during lunch-hour, he walks into the building and goes over to an unattended terminal, loads a binary from of a disk he was carrying, and compiles it. He then executes the program, which continuously thrashes a number of NT servers on the company network with a high-bandwidth stream of UDP datagrams.
- Angry with her manager, an employee compiles and executes a script, which she downloaded from a hacker web site. The program causes the TERM variable to write past the end of the **err_buf** in the DOS emulator shipped with the operating system the company purchased.

Remote users, by contrast, are individuals who do not have access to a local system. They launch remote attacks from systems outside the company network. Remote attacks include viruses and worms, Denial of Service and

Distributed Denial of Service, Man in the Middle, enhanced or elevated user privileges, and privileged access to system resources from a location other than a local account.

1.5 Definition of Risk and Vulnerability

A vulnerability is some aspect of a network resource's functioning, configuration, or architecture that makes the resource a target of potential misuse, exploitation, or denial of service. Vulnerabilities in a system can be attributed to many factors, which include, but are not limited to:

- (1) Software bugs
- (2) System architecture flaws
- (3) Weaknesses in user access control
- (4) System configuration
- (5) Information the network resources make available to users
- (6) Physical organization of a network

Essentially, a vulnerability is any property of a system or network which could be exploited by either by a remote or local user to gain unauthorized access to, or use of, the network's resources or result in the failure of a critical system from performing its proper function.

1.6 Vulnerability Analysis

Vulnerability analysis involves the systematic detection of vulnerabilities in network resources as well as determining potential weaknesses in a company's automated security measures. It is useful to speak of vulnerability analysis as involving two conceptually distinct phases.

Host-level vulnerabilities. Host-based vulnerability analysis involves determining whether settings on the hardware, the configuration of an operating system, or flaws or limitations in the software, produce vulnerabilities in a specific network resource. Buffer overflows in FTP services and weaknesses in user authentication protocols are common examples of Host-based vulnerabilities.

Network-level vulnerabilities. Analyzing network traffic and network-resource performance for the purpose of determining the conditions under which various network resources will fail to meet industry security standards carries out Network-based vulnerability analysis. Network vulnerabilities exist when a specific network resource can be exploited in a manner that causes it to fail in its proper function. Denial of Service (DoS) attacks involve exploiting these vulnerabilities, but this is only one of the ways in which network resources can exhibit vulnerabilities.

Certain vulnerabilities arise from improperly configured hardware or software, which allows unauthorized individuals to learn too much about the layout of a network and the kinds of hardware that provides for its connectivity. Yet another class of network vulnerabilities concerns ways unauthorized persons can gain access to private user and company information by exploiting weaknesses in security measures which provide for the confidentiality of data being transmitted along the company intranet.

The vulnerability score for a network is calculated from evidence gathered during the network discovery and vulnerability analysis phase of an external penetration assessment. The vulnerability score is comprised by the combined results of host-based and network-based vulnerability analysis. In order to gauge the significance of a network's vulnerability score, it is necessary to understanding the assumption underlying the nCircle IP360 scoring method. It is important to look at those assumptions in detail.

IP360 does not currently score an environment based on physical access conditions or use any method of cost analysis for the correction of system vulnerabilities.

1.7 Threat and Risk Assessment

Any flaw, fault, or vulnerability in a network that – if exploited – would adversely affect the network's "data confidentiality" or "resource availability" is considered a threat.

Based on this model, "Data Confidentiality" can be negatively impacted in a number of ways: when an intruder

or unauthorized person

1. Can or does compromise the privacy of sensitive company documents, financial records, user passwords, etc.
2. Employs (or could potentially employ) methods such as traceroute or ICMP broadcasts, to gain confidential information about the network's topology, hardware or software composition, server names and IP addresses, etc.

“Resource Availability” can also be compromised in a number of ways. Based on this scoring system, threats to Resource Availability fall into two general classes:

1. Attack or potential attack that could result in single-system or network-wide Internet connectivity.
2. Attack or potential attack that could halt mission critical operations of hosts or servers on the company network.

These classifications do not characterize the full range of attacks that hostile persons may launch against a company.

Psychological Operations (PSYOPS) and other methods of social programming/engineering can be equally damaging to an organization's data confidentiality or resource availability, though a discussion of the threat and impact of PSYOPS upon customer networks is beyond the scope of this document.⁵

Since company resources are limited and the number of vulnerabilities discovered in networks is often too numerous to be fixed all at once, it is not sufficient merely to catalogue the risks to the network. Ideally, a company needs to have a set of criteria based on their priorities and needs which aid in making decisions as to which vulnerabilities should be assigned the highest priority and which should be fixed last. This methodology implicitly involves measuring and weighing risks to the system.

2.0 nCircle IP360

2.1 *Comments on Vulnerability Scoring*

The IP360 method for scoring the seriousness of a network's vulnerabilities involves far more than a system of equations. In fact, there is no “formula” for security, no a priori method for determining if a company's security is tight enough to keep hackers out. To this extent, even if a company scores very low on the IP360 vulnerability index (indicating overall, that their network security is excellent), this exceptional score is no substitute for caution and common-sense. A very low score on the vulnerability index does not indicate that a company's network is invulnerable. In the same respect, scoring very highly does not indicate that a company's site security is poor, but merely reflects that based on the IP360 security audit findings, there is substantial room for improvement in the company's automated security model.

Finally, the “vulnerability score” for a network is a mathematical abstraction based on the results of a full security audit. The results of this audit are thoroughly described in the reports available in the IP360 console, with recommendations for how site security can be improved and risks to the network minimized. The overall vulnerability score for the customer network and the breakdown of these scores is a metric, not a complete picture of network security. Having acknowledged these considerations, it is now important to discuss the details of how vulnerabilities in the customer network are identified, and scored.

2.2 *Heuristic Approach to Estimating the Penetrability of a Network*

Each vulnerability in a system or network is associated with a specific “risk” value, but this value should not be thought of as an absolute measurement of the threat, which the vulnerability, if left unchecked, poses to the network. This “risk value” changes over time based on factors that are entirely independent of the system or network that exhibits the vulnerability. In interpreting the vulnerability score of a network, there are two very important considerations to keep in mind. Both considerations have to do with the vulnerability score being a heuristic measurement rather than an absolute metric that is not subject to change.

2.3 Vulnerability Scoring

The Vulnerability Score has been developed to address concerns inherent in existing vulnerability rating systems. The model, its mathematical structure and variables, were developed over several years using data collected from over two hundred security audits.

The primary components of the vulnerability score for an exploit (n) are:

- t_n : The number of days that have elapsed since information concerning vulnerability n was first made available on major newsgroups and security-related web-sites.
- r_n : The “class risk” factor, which represents the threat inherent in having vulnerability n on a system S
- S_n : A measurement of the “skill set” required to successfully carry out an attack, which exploits vulnerability n .

Let V_n represent the vulnerability score, which is calculated in the following manner:

$$V_n = \sqrt{t_n} \cdot \frac{r_n!}{S_n^2}$$

2.4 Breakdown of the Vulnerability Score

This section examines how numerical values are assigned to each of the variables employed in the vulnerability score formula. Lastly, a few comments will be offered concerning the formula itself.

2.4.1 A Time-based Approach to Vulnerabilities

The variable t in the “vulnerability score” formula represents the amount of time that information concerning a vulnerability has been available to the public on major newsgroups or websites. One can use CERT Advisories, Vendor Alerts, or any Internet discussion list such as Bugtraq as a base line. To calculate the value for t for a given vulnerability, simply determine how many days have elapsed since news of the vulnerability was first published in an advisory or posted to a discussion group.

Exploit	Date Posted	Current date	t	\sqrt{t}
Palmetto	January 1, 1999	Jan 25 1999	25 days	5

2.4.2 A Risk-based Approach to Vulnerabilities

Based on the IP360 scoring system, it is very easy to determine the “risk” for an exploit. First, the “class” of the exploit must be determined. As previously discussed, the IP360 risk assessment model uses a system of 6 “risk classes” to categorize vulnerabilities (in the order of increasing severity):

- 1) Local attacks against resource availability (e.g. various local DoS attacks)
- 2) Local methods for increasing user privileges
- 3) Local methods for obtaining complete administrative privileges
- 4) Remote attacks against resource availability
- 5) Remote methods for increasing user privileges
- 6) Remote methods for obtaining complete administrative privileges

Calculating the “base risk” for the exploit involves plugging the highest applicable “class risk” value for an exploit into the formula above. Here is the system can be used to calculate the severity or risk for a common vulnerability:

Exploit	Type\Strategy\Impact	class risk (r)	$r!$
Palmetto	Remote\BufferOverflow\Root	6	$6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$

2.4.3 Understanding Skill and the Vulnerability Score

At first glance, measuring or determining the “skill” prerequisites for performing various kinds of attacks

presents a number of difficulties. Even the very idea of numerically quantifying skill levels is nebulous at best, and almost any numerical scheme one could use to represent the degree of difficulty associated with effectively exploiting a vulnerability can be criticized as being uninformative and arbitrary.

The method suggested here avoids these difficulties by using a “tool oriented” method of quantifying how difficult it is to perform certain attacks. This method involves extensive research concerning the types of tools that are publicly available on the Internet. The vulnerabilities that require the least skill to exploit are those for which there exist sophisticated Windows applications that do all of the hard work for the hacker: the would-be hacker is able to install the program, pull up a graphical-user interface, then point, click and root! On the opposite side of the skill-spectrum, vulnerabilities that require the greatest skill are those that are highly “theoretical.” Occasionally, an exploit is referenced in a public newsgroup or advisory but there is no publicly available source code, scripts, or binaries that could be used to automate or facilitate an effective attack on the vulnerability. To effectively exploit this vulnerability requires advanced knowledge, patience, research, and genuine innovation.

The following table describes the IP360 classification scheme for skill in greater detail:

Type of Tools Available	Description of the Tool, Script or Source Code	Class (S)	S^2
GUI	An application that consists of an installation program and an executable Graphical User Interface	1	1
Binary	A non-UNIX binary application, typically containing an installation script, batch file, or other simple installation mechanism. A binary is a precompiled exploit that does not require specific operating system or networking knowledge.	2	4
Non-Windows	A non-Windows binary application, typically a binary containing an installation script, batch file, or other simple installation mechanism. A binary is a precompiled exploit that does require operating system or networking knowledge.	3	9
Script	A non-Windows shell, perl, or interpreted script program that requires limited knowledge of operating systems, shell code, interpreters or networking	4	16
Source Code	An uncompiled set of source files, typically compressed in some way that requires specific knowledge of operating systems compilers, and advanced system experience.	5	25
Not Available	Typically, this category describes an exploit or tool vulnerability that has been referenced in a public forum or advisory and does not include source code, an exploit script, or a reference to predefined exploit source.	6	36

In the FTP buffer overflow example, assume that there exist certain attack scripts that are widely available on the Internet. The value of S^2 is calculated as follows:

Type of Tool Available	Class	S^2
FTP attack script	4	16

2.4.4 Calculating a Sample Vulnerability Score

By fitting the values derived in sections 2.4.1, 2.4.2, and 2.4.3 into the vulnerability score formula, the result is a vulnerability value for the exploit under consideration.

$$V_{\text{FTP Buffer-Overflow}} = \sqrt{25} \cdot \frac{6!}{4^2}$$

$$\text{OR } 5 \cdot \frac{720}{16}$$

$$V_{\text{FTP Buffer Overflow}} = 225$$

A few comments are in order to put the score for the FTP buffer overflow in perspective. First, the value derived for the exploit is relative to January 25, 1999. If, however, another 75 days pass and the exploit remains unpatched on the target system, the vulnerability score for this exploit will have risen accordingly. In fact, after a total of 100 days, the vulnerability score for the FTP buffer overflow will have doubled:

$$V_{\text{FTP Buffer Overflow}} = \sqrt{100} \cdot \frac{6!}{4^2}$$

$$\text{OR } 10 \cdot \frac{720}{16}$$

$$V_{\text{FTP Buffer Overflow}} = 450$$

3.0 Comparative Risk Assessment

3.1 Calculating the Vulnerability of a Single Resource

The “vulnerability score” for a single resource on the network (e.g. a firewall, an individual computer, a router, etc.) is the sum of the risk values (i.e. vulnerability scores) for each of the vulnerabilities discovered in the resource. Assume that the FTP buffer overflow vulnerability discussed in the previous sections is one of several vulnerabilities discovered on a particular sever. Calculating the vulnerability score for that server is a matter of summing up all of the individual vulnerability scores for the exploits discovered in that machine. To calculate the vulnerability score for the Server S , the formula presented below is used, where V_1 is the first vulnerability discovered in S and V_n is the last:

$$S_n = \sum_{i=1}^n V_i = V_1 + V_2 + V_3 \dots V_{n-1} + V_n$$

It is useful to calculate the combined vulnerability score for a single resource because this is a quick way for identifying highly vulnerable systems on the company network. Examining the vulnerability scores of various network resources provides a basis for making quick comparisons in vulnerability levels across disparate systems along the company network. The IP360 Console displays several views of vulnerability information, including per resource and consolidated.

3.2 Calculating the Vulnerability of a Network

The raw score is the sum of all of the vulnerability scores from each of the network systems and resources, where S_n is the value of the combined scores of the individual vulnerabilities discovered in a resource S , as discussed in section 3.1.

The raw score is calculated in the following way:

$$T_n = \sum_{i=1}^n S_i = S_1 + S_2 + S_3 \dots S_{n-1} + S_n$$

The total vulnerability score will be higher for larger networks.

3.6 Calculating Average Vulnerability

The “average vulnerability score per system” is the total vulnerability score for the network divided by the total number of systems audited.

Where f is the frequency of a vulnerability

Where x is the base vulnerability score of a single exploit

Where a is the total number of systems audited

Where n is the number of vulnerabilities identified

$$AV = \sum_{i=1}^n \frac{f_i x_i}{a_i} = \frac{f_1 x_1 + f_2 x_2 + \dots + f_n x_n}{a_1 + a_2 \dots a_n} = \frac{T_n}{a}$$

3.7 Calculating Average Vulnerability per System

The “average vulnerability score per vulnerable system” is calculated by dividing the total vulnerability score of the network, as discussed in section 3.2, by the number of systems that exhibited at least one vulnerability.

Where f is the frequency of a vulnerability

Where x is the base vulnerability score of a single exploit

Where s is the number of systems in which at least one vulnerability was discovered

Where n is the number of vulnerabilities identified

$$AV_s = \sum_{i=1}^n \frac{f_i x_i}{s_i} = \frac{f_1 x_1 + f_2 x_2 + \dots + f_n x_n}{s_1 + s_2 \dots s_n} = \frac{T_n}{s}$$

3.8 Identifying High Vulnerabilities

Any systems with vulnerability scores above a certain threshold are flagged and presented in a report from the IP360 Console.

4.0 Logical Consequences of the IP360 Scoring System

With the information available from IP360, enterprises can reduce their vulnerability to attack by:

- Exploit containment – changing application or network configuration to make vulnerable condition less accessible to attackers.
- Exploit eradication – applying vendor patches, shutting down services, or otherwise eliminating the vulnerability, so the exploit cannot succeed.

Six Common Vulnerabilities

1. Hosts running unnecessary services (i.e. finger, anonymous ftp, tenet, portmapper)
2. Unpatched, outdated operating systems and vulnerable RPM packages or software or firmware that should be upgraded.
3. Numerous instances of services and servers needlessly display sensitive information to remote users (i.e. DNS version numbers, tenet\ftp banners, systat, finger, netstat, open NetBIOS nameable, open DNS listings, etc.)
4. Easily exploitable trust-relationships relying on vulnerable services such as rlogin, rsh, exec, etc.
5. Misconfigured firewalls and routers (i.e. firewalls which allow ICMP traffic through; external routers that allow for remote access and configuration but fail to perform any kind of authentication mechanisms.)
6. Default passwords on routers, firewalls, etc.

4.1 An Overview of Exploit Containment

A company can decrease its vulnerability score by practicing a containment policy in regard to vulnerabilities. If attack A on vulnerability v can be launched remotely, and the company reconfigures their existing security devices to disallow remote attacks against v , their vulnerability index score will drop. Even if the company does not install the vendor-specific patches to correct vulnerability v , their score will drop simply because they have eliminated an attacker's ability to exploit v remotely.

Before containment:

Exploits

Remote UDP
Diagnostic scans
DoS

Vulnerability Score

$$\sqrt{60days} \bullet \frac{4!}{1^2} = \mathbf{185.9 \text{ per vulnerable host}}$$

After a thorough containment policy in regard the remote UDP diagnostic DoS their score will be lowered significantly. Every system which was previously vulnerable to a remote UDP DoS is now only vulnerable to a Local UDP DoS. Let's say it takes them 10 days to make the necessary changes.

Exploits

Local UDP
Diagnostic scans
DoS

Vulnerability Score

$$\sqrt{70days} \bullet \frac{1!}{1^2} = \mathbf{8.36 \text{ per vulnerable host}}$$

As is apparent, if the 25 hosts were vulnerable to remote DoS before containment, the raw score of the network would be rather high. After containment, the raw score would be much lower.

4.2 Exploit Eradication

Companies that practice exploit eradication in addition to exploit containment will score the best on the IP360 model. Although a strategy of "perfect eradication" is not feasible, to maximize a company's security does require a balance of containment and eradication efforts. Some vulnerabilities will result from normal business operations, so it will not be surprising that some types of companies will generally tend to score higher than others, depending upon their business model.

End Notes and References

1. cf. NIPC Cybernotes #2-99, pg. 4.
2. cf. Escamillia, Terry, *Intrusion Detection: Network Security Beyond the Firewall* (John Wiley & Sons Inc, New York, 1998, pg. 182.)
3. cf NIPC CyberNotes, #2-99, pg. 4.
4. cf. Escamillia, 1998, pg. 182.
5. Companies who are concerned with the threat posed by PSYOPS to the integrity of their organization should e-mail info@ncircle.com for more information. Private consultations concerning this service are available.
6. The team members of nCircle are indebted to the ISSO at the National Security Agency for its help in compiling these industry-standard definitions.