tripwire

WHITE PAPER

The Tripwire Vulnerability Scoring System

FOUNDATIONAL CONTROLS FOR
SECURITY, COMPLIANCE & IT OPERATIONS

Vulnerability and Risk Analysis

Measuring and managing the security risk associated with information and information technology remains one of the most challenging and debated problems faced by all levels of an organization. While scoring standards designed to assist with solving this problem have been developed over the past decade, a select few have accomplished this and those that have are still in their infancy. While there are several options available today, most do not provide a suitable metric nor do they improve an organization's ability to effectively manage risk.

Historical Context

The Tripwire® IP360™ vulnerability scoring formula was published in 2001, at a time when vulnerability and risk management was still being defined. Around the same time, a paper detailing the formula and comparing the alternatives was published; this paper provides updated information and comparison data. After nearly two decades, the Tripwire IP360 vulnerability score continues to prove itself as the leading measurement metric in vulnerability and risk management.

With 19 years of industry application, the Tripwire IP360 score is a viable and tested model of risk assessment with unparalleled longevity. This paper explains in detail the thought behind the metric, as well as the actual formula used for calculating Tripwire IP360 vulnerability scores. This model is presented as an "open-source" method of risk analysis; anyone who finds the method useful is encouraged to utilize it. While there are many other methods of risk analysis available, this paper argues that other common industry models are either inherently ambiguous or yet untested, limiting their applicability to real world situations.

Details and Definitions

Vulnerability management and vulnerability scoring do not exist in a vacuum. In order to provide a cohesive and comprehensible paper, this section aims to set a common vocabulary and context.

Vulnerability Management

Vulnerability management is the process of assessing the existence and severity of vulnerability conditions within an organization, including the workflow and process for making mitigation decisions about the vulnerabilities.

Vulnerability Scoring

Vulnerability scoring is the process or method for describing the risk that a specific vulnerability presents.

Vulnerability management is a component of risk management, which encompasses the areas of disaster recovery, business continuity, policy and physical security. Vulnerability scoring, in turn, is a tool used to make vulnerability management more effective. In the domain of information security, with which this paper is most concerned, all of these practices deal explicitly with the protection of data.

Data refers to the information stored on and passed between the hosts within an organization. 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 an information security standpoint, security objectives are applied to information and information systems that have been categorized (800-53 - FIPS 1991). These security objectives are most commonly discussed as confidentiality, integrity, and availability. The loss of any one of these characteristics constitutes an incident. With this in mind, threats to network security fall into three general classes:

- Threats to data *confidentiality*
- Threats to data *integrity*
- Threats to data *availability*

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, e.g. the realization of a threat to confidentiality, integrity or availability. In other words, a vulnerability is an opportunity for threat to be realized. Vulnerabilities in a system can be attributed to many factors, which include, but are not limited to:

- Software bugs
- System architecture flaws
- Weaknesses in user access control
- System configuration
- Information the network resources make available to users
- Physical organization of a network

Risk, then, is the potential that the threat will be realized for a particular vulnerability. The relationship of vulnerability, threat, risk and exploit is important to understand. These are terms that often get misused, and whose definitions have changed over time.

Vulnerability Analysis

Vulnerability analysis involves the systematic detection of vulnerabilities in network resources. This is distinct from the process of vulnerability research, which involves the discovery and documentation of vulnerability conditions. It is also distinct from the process of vulnerability management, which addresses the larger process of reacting to vulnerability analysis. Vulnerabilities can exist at multiple layers of the network infrastructure. It's important to keep in mind the common levels and their differences, as the analytics apply in different ways at different levels.

Endpoint Conditions

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

Network Conditions

The context in which an endpoint functions in a network environment further defines its risk level. Improperly configured access control, routing conditions and network points of failure constitute network vulnerability conditions.

Often, these two types of conditions are analyzed separately, though the analysis of either one clearly has bearing on the analysis of the other. A major reason for this analysis gap is the lack of a common, usable metric for measuring risk.

As any organization's resources are limited, the number of vulnerabilities discovered is generally too numerous for complete remediation to be achieved. Given a network of ten thousand hosts, with a conservative average of ten to twenty vulnerabilities each, the full list of conditions would easily reach into the hundreds of thousands. Clearly, it is not sufficient merely to catalogue the risks to the network—the organization must have a set of criteria by which to categorize the discovered conditions and aid in making effective risk mitigation decisions. These requirements implicitly involve a model of risk analysis specific to the condition of network resources, of which several have been developed. The vulnerability score, then, is a risk metric or categorization applied to a specific vulnerability.

Vulnerability Scoring Models

The process of assessing a host for vulnerabilities and reporting on the data found is not new. There are a number of established means of ranking vulnerabilities.

Method 1: Keyword Model (aka Severity Rating)

This method has been around longer than the Tripwire IP360 scoring system, although there have been a number of variations. Historically, this method can be traced back to CyberNotes, a monthly security update started in 1999 by the National Infrastructure Protection Center (NIPC). CyberNotes used a three-tier severity rating system that included Low-, Medium- and High-risk categories. Prior to the establishment of this system, there had been little effort to establish an industry standard.

CyberNotes defined these categories in the following manner:

- **High** – A high-risk vulnerability is defined as one that will allow an intruder to immediately gain privileged access (e.g., sysadmin or root) to the system or allow an intruder to execute code or alter arbitrary system files. An example of a high-risk vulnerability is one that allows an unauthorized user to send a sequence of instructions to a machine and the machine responds with a command prompt with administrator privileges.
- **Medium** – A medium-risk vulnerability is defined as one that will allow an intruder immediate access to a system with less than privileged access. Such vulnerability will allow the intruder the opportunity to continue the attempt to gain privileged access. An example of medium-risk vulnerability is a server configuration error that allows an intruder to capture the password file.
- **Low** – A low-risk vulnerability is defined as one that will provide information to an intruder that could lead to further compromise attempts or a Denial of Service (DoS) attack. It should be noted that 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 to be a "High" threat.

Although NIPC has stopped issuing CyberNotes, the use of severity as a scoring model continues to be prevalent today. Numerous vulnerability management products still utilize this approach and the latest version of the Common Vulnerability Score System, CVSSv3, even provides a CVSS to Severity translation table.

Method 2: Numeric Model

Another popular method of measuring risk is to use a numeric ranking. This system works similarly to severity-based ratings but provides a clear benefit via vulnerability scoring aggregation.

Version 1.1 of the PCI DSS Security Scanning Procedures2 is one example of a numeric rank. The document provides 5 levels, also mapped to severity ratings, used to score vulnerabilities.

It's easy to see how vulnerability information could be aggregated across hosts to present a clear view of the risk on a network. Under this system, a host with two critical vulnerabilities and a high vulnerability could be aggregated to a score of 11, while a host with a single urgent vulnerability would continue to score a 5. This aggregation capability is a great improvement over the keyword model.

Method 3: Common Vulnerability Scoring System (CVSS)

The National Infrastructure Advisory Council launched CVSS in 2004. FIRST (Forum of Incident Response and Security Teams) currently maintains CVSS and was responsible for the launch of CVSSv2 in 2007 and CVSSv3 in 2015.

CVSS is broken into three score components:

- **Base** – This scores the vulnerability itself. Base scores should not change over time.
- **Temporal** – Items that impact the vulnerability that change over time. This includes patch availability, known exploits and vendor confirmation.

Level	Severity	Description
5	Urgent	Trojan Horses; file read and writes exploit; remote command execution
4	Critical	Potential Trojan Horses; file read exploit
3	High	Limited exploit of read; directory browsing; DoS
2	Medium	Sensitive configuration information can be obtained by hackers
1	Low	Information can be obtained by hackers on configuration

» **Environmental** – A selection of items to help organizations tailor scores for their environment.

CVSSv3 has not yet seen wide adoption but presents a solid base with clear definitions and guidance, which could lead to a valued scoring system. Unfortunately, this has not been true of previous CVSS implementations. CVSSv2 was plagued by ambiguous definitions and a lack of guidance. This ultimately lead to analysts making judgment calls regarding scoring which lead to a disparity among scores for similar items. Additionally, some vendors felt that CVSSv2 left holes in the various ranking scales, creating their own modifications to the system to more accurately reflect their interpretation of issues.

While CVSSv2 was widely adopted and recognized as an industry standard, it was never the complete solution that many hoped it could be. With the release of the greatly improved CVSSv3, the first viable, true industry standard may be possible. It is not without flaws but it's a big step in the right direction.

Model Methodology

It's important to note the methods utilized by the various models mentioned above. There are key aspects of vulnerability scoring models that are used to varying degrees.

Depth of Access Principle

The first method is the depth of access principle, in which vulnerabilities are ranked based on the level (depth) of access that a successful exploit provides. The logic is that the greater the depth of access, the greater the risk. The Keyword model detailed above relies solely on this principle, which also plays a role in the Numeric model. Depth of Access is reflected within CVSS via the measurement of confidentiality, integrity, and availability.

Quality of Information Concept

This method is not applied to the Keyword model but can be seen as a secondary consideration in the Numeric model detailed above. Levels 1, 2, and 3 all detail the availability of information; it's the quality of that information that adjusts the ranking. The more sensitive the information provided by successful exploitation, the higher the level.

Class of System Concept

This final concept is applicable to classifying vulnerabilities within one's organization. The Numeric example attempts to work this method into a generic scoring system, however the application is rather limited. This method is used quite well in the CVSS environmental score, which measures the value of a system and it's distribution within the network.

Limitations of Current Models

All of the above models (both the model itself and the examples used) have limitations. These limitations prevent them from completely addressing the needs of a proper risk management system. Three major limitations include subjectivity, ambiguity and inaccuracy due to a lack of contextual reference.

Limitation 1: Subjectivity

In order for a risk analysis metric or ranking to be used in any common way, the methodology for applying the rankings to vulnerability conditions must be objective. The high/medium/low model clearly fails to provide an objective methodology. There is nothing inherent in the term "high" to define to which conditions that category might apply. An external definition must be applied.

The Numeric model suffers from the same limitation, though it is somewhat masked by the use of a number instead of a text-based label. A number as category, however, contains no inherent definition of its contents either.

Subjectivity creates a significant problem in implementation. The value of these risk assessment models increases in proportion to the number of vulnerabilities that have been classified by the method in a customer's environment. As customers try to make sense of the vulnerability information reported to them, the requirement to use the vulnerability rankings systematically increases in importance. The subjective nature of these models makes that difficult.

CVSS has worked to avoid this by providing values for the various components but the metric still suffers from some subjectivity problems. These issues have surfaced when different organizations produce different CVSS scores for the same condition. CVSSv3 has done a great job of clearing up much of the subjectivity that existed within CVSSv2.

Limitation 2: Ambiguity

There is an important distinction between a ranking and a true metric. A ranking provides only a relative distinction between conditions. Its accuracy decreases in proportion to the number of conditions being ranked, or alternatively, the number of available rankings must be increased. A metric provides an atomic measurement of a condition, regardless of the other conditions that exist. In other words, a ranking is meaningful only relative to the ranking of other conditions, whereas a metric is meaningful in isolation or amidst n number of conditions.

The importance of this distinction is directly related to the number of

vulnerabilities that exist and to the number of organizations performing vulnerability assessments. If an assessment of ten hosts is performed, the pool of possible vulnerabilities is fairly small, so distinguishing between hosts is possible with rankings. If, however, an assessment contains thousands of hosts, each with tens of vulnerability conditions, then the ability to distinguish conditions and their risk based on a ranking becomes challenging without a dramatic increase in the number of available rankings.

The move from text-based categories to numeric rankings may appear to address this limitation through aggregation, but it does not. There is no more repeatable or objective logic behind the numeric ranking than the text-based categories. The input data that produces the numbers suffers from the same underlying limitation of subjectivity, making the result of any aggregation ultimately subjective as well.

This is a limitation that applies to the first two methods, but not to CVSS. CVSS makes the move to deliver a metric, rather than a ranking. The metric, however, is limited to delivering a finite scale of 1–10, which inherits some of the problems with relative values that the other methods exhibit.

Limitation 3: Contextual Reference

There are actually two distinct limitations in the sense of contextual relevance for the example methods.

Both the high/medium/low method and the Numeric ranking method suffer from the fact that they do not account for the context of time. A condition, once labeled, does not change. This means that as a vulnerability ages and the exploit techniques become more widely distributed, the vulnerability ranking remains the same, although the risk that vulnerability presents does not.

While the ranking methods ignore the time parameter, CVSS and Tripwire have both similar and opposing views of time. The similarity exists in measuring the maturity of the exploit code or skill required to successfully exploit the vulnerability. The opposing view is the measure of time. While the Tripwire Vulnerability Scoring System, as noted below, measures the age of the vulnerability, CVSS looks at the confidence of the report and the availability of a fix.

When temporal scores are generated, CVSS looks to measure the risk presented to the overall user base. If a vulnerability has a patch available, it is less of a risk than a vulnerability without a patch. While this is true in the generic sense, if a vulnerability exists within an organization, it is not patched and the availability of a patch does not reduce the current risk to the organization. This is a great example of a common metric (CVSS) versus an operational metric (Tripwire Vulnerability Scoring System).

Tripwire Vulnerability Scoring System

Context of Vulnerability Scoring

Vulnerability scoring does not equate completely with security. In fact, there is no "formula" for security, no a priori method for determining if a company's security is tight enough to keep criminals out. The vulnerability score itself does not account for the context in which that vulnerability has been discovered. Tripwire IP360 also uses host "Asset Values" to provide business context to the vulnerability scores. Asset Values are provided by the customer and are integers (typically dollar values) that denote the value of a particular host in the enterprise. Representing the value of the asset is an important component of prioritization; if a host has a high vulnerability score but a low asset value, the security administrator may choose not to focus on it, whereas if its asset value was high, it would clearly be a priority.

The Tripwire IP360 vulnerability score is, ultimately, a mathematical abstraction based on the results of an assessment. The results of Tripwire IP360 assessments are thoroughly described in the reports available through the Tripwire VnE Manager, including recommendations for how site security can be improved and risks to the network minimized. Having acknowledged these considerations, it is now important to discuss the details of how vulnerabilities in the customer network are identified and scored.

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. When 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.

The Vulnerability Scoring Equation

The Tripwire IP360 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 thousands of security audits. The primary components of the vulnerability score for a condition (n) are:

t_n : The number of days that have elapsed since information concerning vulnerability n was first made available via major security sources.

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:

$$V_n = \sqrt{t_n} \times \frac{r_n!}{{s_n}^2}$$

Analysis 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.

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 from major security sources. These sources may change over time, but the concept of public availability remains consistent. One can consider such sources as CERT Advisories, vendor alerts, mailing lists or news feeds as current examples of such sources. 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.

CVE ID: CVE-2005-1983
Date Posted: 8/9/2005
Current date: 11/18/2015
t : 3753 days
$\sqrt{t}$: 61.3

A Risk-based Approach to Vulnerabilities

The Tripwire scoring system classifies risk into seven distinct categories. It uses a system of six risk classes to categorize vulnerabilities and an additional exposure level to classify information disclosure. Base risk is calculated based on the highest level of access obtained when a vulnerability is successfully exploited. One assumption in selecting the appropriate risk class is that the principle of least privilege is applied to the system. This means that, where possible, the assumption is made that software is running with user level privileges.

Here is an example using CVE-2005-1983 from MS05-039:

Exploit: CVE-2005-1983
Class: 6
Risk (r) ***r!*** **:** 720

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 Tripwire model avoids these difficulties by using a "tool oriented" method of quantifying how difficult it is to perform certain attacks. The vulnerabilities that require the least skill to exploit are those for which there exist sophisticated applications that do all of the hard work for the user—the user 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 the vulnerability requires advanced knowledge, patience, research and genuine innovation.

For the MS05-039 example from above, there is exploit source code easily available. The value of S^2, therefore, is calculated as follows:

Type of Tool Available:
Automated Exploit
Class: 1
s^2 **:** 1

Calculating a Sample Vulnerability Score

By fitting the values derived in the above sections into the vulnerability score formula, the result is a vulnerability value for the vulnerability under consideration as of the date 11/18/2015.

$$V_n = \sqrt{3753} \times \frac{6!}{1^2}$$

$$V_n = 61.3 \times 720$$

$$V_n = 44136$$

It is important to consider that the Tripwire Vulnerability Score provides a metric for a vulnerability at a point in

Label	Description	Risk *(r)*	*r!*
Exposure	Information Disclosure	0	0
Local Availability	Local attacks against availability (e.g. DoS)	1	1
Local Access	Local methods for obtaining or increasing user-level privileges	2	2
Local Privilege	Local methods for obtaining complete administrative privileges.	3	6
Remote Availability	Remote attacks against availability (e.g. DoS)	4	24
Remote Access	Remote methods of obtaining or increasing user-level privileges	5	120
Remote Privilege	Remote methods for obtaining complete administrative privileges	6	720

*Local can be considered Authenticated. Remote can be considered Unauthenticated.

time. In practice, this point in time is almost always "now," but the metric can be used to predict risk increases. For example, in an additional 365 days, the MS05-039 condition will score like this:

$$V_n = 64.1 \times 720$$

$$V_n = 46152$$

Extending Vulnerability Scoring

Calculating the Vulnerability Score of a Host

To this point, the vulnerability score has been applied to a condition. While the condition may be the atomic unit in vulnerability management, enterprises do not think of their environments as collections of conditions. Hosts and their applications are also valid targets for vulnerability assessment and scoring.

The vulnerability score for a host 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 on that host. MS05-039 preceded MS05-041. A host exhibiting CVE-2005-1983 is also likely to exhibit CVE-2005-2303. Calculating the vulnerability score for that host, then, is a matter of summing up all of the individual vulnerability scores for the conditions discovered on that host. To calculate the vulnerability score for the host S, the formula presented below is used, where V1 is the first vulnerability discovered in S and Vn is the last:

$$V_1 + \cdots V_n = V_s$$

It's useful to calculate the combined vulnerability score for a single host for multiple reasons. First, examining the vulnerability scores of various network resources provides a basis for making effective remediation decisions based on host, rather than vulnerability. The ability to prioritize hosts is important in the enterprise. Additionally, the metric can be used as a means of comparison and for change detection. Hosts that have the same configuration and function should produce the same score. Tripwire IP360 displays several views of vulnerability information, including per resource and consolidated.

Label	Description	Skill *(s)*	s^2
Automated Exploit	An exploit is available in an exploit kit, exploit framework, or malware (e.g. Worm).	1	1
Easy	Fully functional exploit code is available, likely in an exploit repository.	2	4
Moderate	Exploit Code is available but may not be fully functional.	3	9
Difficult	A proof of concept is available.	4	16
Extremely Difficult	Minimal details are available—perhaps a technical write-up with no proof of concept.	5	25
No Known Exploit	No exploits are available.	6	36
Remote Privilege	Remote methods for obtaining complete administrative privileges	6	720

Calculating the Vulnerability Score of a Network

Tripwire IP360 represents IP space as collections of user defined networks. It is trivial, given the explanation for a host vulnerability score, to produce a network vulnerability score in the same manner. The network score is the sum of all of the vulnerability scores from each of the network systems, where Sn is the value of the combined scores of the individual vulnerabilities discovered in a resource S, as discussed in section 3.1.

The network score is calculated in the following way:

$$S_{n1} + S_{n2} + S_{n3} \ \ldots = V_n$$

Further, the network score can be used for trending risk posture over time. In a large enterprise, the ability to effectively track and trend risk posture is invaluable.

Logical Consequences of the Tripwire Scoring System

Any metric applied consistently to an environment has consequences. The act of measuring in a consistent manner produces changes in behavior. The behavioral changes vary based on what that metric actually measures, of course. Thus, what you measure directly affects what change eventually occurs. An inappropriate method of measuring risk will result in equally inappropriate actions to reduce risk. Only by applying a valid metric can reasonable risk reduction actions be taken.

This paper demonstrates that the Tripwire IP360 vulnerability score provides a valid and relevant base metric for measuring IT risk in the corporate environment. Adoption of the vulnerability score as a foundational metric for measuring risk in the enterprise, in conjunction with host Asset Values and other Tripwire solutions, enables organizations to more effectively:

- Measure network security risk using objective metrics
- Manage network security risk through dashboard reporting and integration with existing enterprise systems
- Reduce network security risk by focusing IT resources on the highest priority risks

References

1 http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf

2 https://www.pcisecuritystandards.org/pdfs/pci_scanning_procedures_v1-1.pdf

Tripwire is the trusted leader for establishing a strong cybersecurity foundation. We protect the world's leading organizations against the most damaging cyberattacks, keeping pace with rapidly changing tech complexities to defend against ever-evolving threats for more than 20 years. On-site and in the cloud, our diverse portfolio of solutions find, monitor and mitigate risks to organizations' digital infrastructure—all without disrupting day-to-day operations or productivity. Think of us as the invisible line that keeps systems safe. **Learn more at tripwire.com**

***The State of Security*: News, trends and insights at tripwire.com/blog**
Connect with us on LinkedIn, Twitter and Facebook

WPTVSS2k 1601