# The Center for Internet Security

# The CIS Security Metrics

May 11

2009

Organizations struggle to make cost-effective security investment decisions; information security professionals lack widely accepted and unambiguous metrics for decision support. CIS established a consensus team of one hundred (100) industry experts to address this need. The result is a set of standard metrics and data definitions that can be used across organizations to collect and analyze data on security process performance and outcomes.

This document contains twenty (20) metric definitions for six (6) important business functions: Incident Management, Vulnerability Management, Patch Management, Application Security, Configuration Management and Financial Metrics. Additional consensus metrics are currently being defined for these and additional business functions.

## Consensus Metric Definitions v1.0.0

# Contents

# Terms of Use Agreement

The nonprofit Center for Internet Security ("**CIS**") provides consensus-oriented information security products, services, tools, metrics, suggestions, and recommendations (the "**CIS Products**") as a public service to Internet users worldwide. **Downloading or using any CIS Product in any way signifies and confirms your acceptance of and your binding agreement to these CIS Terms of Use**.

## CIS Terms of Use

### Both CIS Members and non-Members may:

- Download, install, and use each of the CIS Products on a single computer, and/or
- Print one or more copies of any CIS Product that is in a .txt, .pdf, .doc, .mcw, or .rtf format, but only if each such copy is printed in its entirety and is kept intact, including without limitation the text of these CIS Terms of Use.

### Under the Following Terms and Conditions:

- **CIS Products Provided As Is.** CIS is providing the CIS Products "as is" and "as available" without: (1) any representations, warranties, or covenants of any kind whatsoever (including the absence of any warranty regarding: (a) the effect or lack of effect of any CIS Product on the operation or the security of any network, system, software, hardware, or any component of any of them, and (b) the accuracy, utility, reliability, timeliness, or completeness of any CIS Product); or (2) the responsibility to make or notify you of any corrections, updates, upgrades, or fixes.
- **Intellectual Property and Rights Reserved**. You are not acquiring any title or ownership rights in or to any CIS Product, and full title and all ownership rights to the CIS Products remain the exclusive property of CIS. All rights to the CIS Products not expressly granted in these Terms of Use are hereby reserved.
- **Restrictions.** You acknowledge and agree that you may not: (1) decompile, dis-assemble, alter, reverse engineer, or otherwise attempt to derive the source code for any software CIS Product that is not already in the form of source code; (2) distribute, redistribute, sell, rent, lease, sublicense or otherwise transfer or exploit any rights to any CIS Product in any way or for any purpose; (3) post any CIS Product on any website, bulletin board, ftp server, newsgroup, or other similar mechanism or device; (4) remove from or alter these CIS Terms of Use on any CIS Product; (5) remove or alter any proprietary notices on any CIS Product; (6) use any CIS Product or any component of a CIS Product with any derivative works based directly on a CIS Product or any component of a CIS Product; (7) use any CIS Product or any component of a CIS Product with other products or applications that are directly and specifically dependent on such CIS Product or any component for any part of their functionality; (8) represent or claim a particular level of compliance or consistency with any CIS Product; or (9) facilitate or otherwise aid other individuals or entities in violating these CIS Terms of Use.
- **Your Responsibility to Evaluate Risks**. You acknowledge and agree that: (1) no network, system, device, hardware, software, or component can be made fully secure; (2) you have the sole responsibility to evaluate the risks and benefits of the CIS Products to your particular circumstances and requirements; and (3) CIS is not assuming any of the liabilities associated with your use of any or all of the CIS Products.
- **CIS Liability.** You acknowledge and agree that neither CIS nor any of its employees, officers, directors, agents or other service providers has or will have any liability to you whatsoever (whether based in contract, tort, strict liability or otherwise) for any direct, indirect, incidental, consequential, or special damages that arise out of or are connected in any way with your use of any CIS Product.
- **Indemnification.** You agree to indemnify, defend, and hold CIS and all of CIS's employees, officers, directors, agents and other service providers harmless from and against any liabilities, costs and expenses incurred by any of them in connection with your violation of these CIS Terms of Use.
- **Jurisdiction**. You acknowledge and agree that: (1) these CIS Terms of Use will be governed by and construed in accordance with the laws of the State of Maryland; (2) any action at law or in equity arising out of or relating to these CIS Terms of Use shall be filed only in the courts located in the State of Maryland; and (3) you hereby consent and submit to the personal jurisdiction of such courts for the purposes of litigating any such action.

**Special Rules for CIS Member Organizations:**

CIS reserves the right to create special rules for: (1) CIS Members; and (2) Non-Member organizations and individuals with which CIS has a written contractual relationship. CIS hereby grants to each CIS Member Organization in good standing the right to distribute the CIS Products within such Member's own organization, whether by manual or electronic means. Each such Member Organization acknowledges and agrees that the foregoing grants in this paragraph are subject to the terms of such Member's membership arrangement with CIS and may, therefore, be modified or terminated by CIS at any time.

# Background

## Consensus Guidance

This guide was created using a consensus process comprised of volunteer and contract subject matter experts. Consensus participants provide perspective from a diverse set of backgrounds including consulting, software development, audit and compliance, security research, operations, government and legal.

## Intent and Scope

This initial set comprises metrics and business function selected as a starting point by the metrics community, both in terms of the scope of the metrics across business functions and the depth of the metrics in assessing security outcomes and performance. Once these foundational datasets and metrics are in place, additional metrics can and will be developed by the community covering additional functions and topics in each function.

## Management Perspective and Benefits

The immediate objective of these definitions is to help enterprises improve their overall level of security and reduce costs by providing a set of standard metrics that can be implemented in a wide range of organizations. A future objective is to provide standard metrics as a basis for inter-enterprise benchmarking. These security control metrics were selected for common security functions and concepts based on the availability of data, value provided for security management, and their ability to communicate the state of security performance. Organizations can create a foundation for a metrics program by first selecting metrics from the business management areas of immediate interest and then implement one or more of the metrics based on the definitions provided in this document. This well-defined set of standard metrics will enable the use of metrics in the security community by providing:

- **Clear Guidance for Organizations on Implementing Metrics**. Practical definitions of security metrics based on data most organizations are already collecting. This will make it easier, faster, and cheaper to implement a metrics program that supports effective decision-making. Metrics provide a means of communicating security performance and can be used to guide resource allocation, identify best practices, improve risk management effectiveness, align business and security decision-making, and demonstrate compliance.

- **Defined Metric Framework for Security Products and Services**. A clear set of data requirements and consensus-based metric definitions will enable vendors to efficiently incorporate and enhance their security products with metrics. Consensus-driven metric standards will provide ways to demonstrate the effectiveness of vendor products, processes, and services assist the state of their customers.

- **Common Standards for Meaningful Data Sharing and Benchmarking**. Metric results will be calculated uniformly enabling meaningful benchmarking among business partners and industry sectors. A shared metric framework and the ability to track and compare results will leverage the capabilities of the entire security community, leading to best practice identification and improvements in overall information security practices.

## Business Functions

This initial document provides twenty consensus metrics definitions for six important business functions. Organizations can adopt the metrics based on the business functions of highest priority. More metrics will be defined in the future for these and additional business functions.

Table 1: Business Functions

| Business Functions | | |
|---|---|---|
| **Function** | **Management Perspective** | **Defined Metrics** |
| **Incident Management** | How well do we detect, accurately identify, handle, and recover from security incidents? | • Mean-Time to Incident Discovery<br>• Number of Incidents<br>• Mean-Time Between Security Incidents<br>• Mean-Time to Incident Recovery |
| **Vulnerability Management** | How well do we manage the exposure of the organization to vulnerabilities by identifying and mitigating known vulnerabilities? | • Vulnerability Scanning Coverage<br>• Percent of Systems with No Known Severe Vulnerabilities<br>• Mean-Time to Mitigate Vulnerabilities<br>• Number of Known Vulnerabilities |
| **Patch Management** | How well are we able to maintain the patch state of our systems? | • Patch Policy Compliance<br>• Patch Management Coverage<br>• Mean-Time to Patch |
| **Application Security** | Can we rely on the security model of business applications to operate as intended? | • Number of Applications<br>• Percent of Critical Applications<br>• Risk Assessment Coverage<br>• Security Testing Coverage |
| **Configuration Management** | How do changes to system configurations affect the security of the organization? | • Mean-Time to Complete Changes<br>• Percent of Changes with Security Reviews<br>• Percent of Changes with Security Exceptions |
| **Financial Metrics** | What is the level and purpose of spending on information security? | • IT Security Spending as % of IT Budget<br>• IT Security Budget Allocation |
| **Future Functions** | Community recommendations for additional business functions include: | • Data / Information<br>• Software Development Life-Cycle<br>• Configuration Management<br>• Third Party Risk Management<br>• Additional Financial and Impact Metrics<br>• Authentication and Authorization<br>• Data and Network Security<br>• Remediation Efforts<br>• Anti-Malware Controls |

# Incident Management

This section describes metrics for measuring the processes for detecting, handling, and recovering from security incidents.

As described in the *Glossary* section of this document, a *security incident* results in the actual outcomes of a business process deviating from expected outcomes for confidentiality, integrity, and availability resulting from people, process, or technology deficiencies or failures[1]. Incidents that should not be considered "security incidents" include disruption of service due to equipment failures.

## Data Attributes

The following is a list of attributes that should be populated as completely as possible for each security incident.

**Table 2: Security Incidents Table**

### Security Incidents Table

| Name | Type | De-Identified | Required | Description |
|---|---|---|---|---|
| Incident ID | Number | No | No | Unique identifier for the incident. Generally auto-generated. |
| Date of Occurrence | Date / Time | No | Yes | Date and time the incident occurred. |
| Date of Discovery | Date / Time | No | Yes | Date and time the incident was discovered |
| Discovered By | Text | Yes | No | The name of the person or system that first discovered the incident. |
| Detected by Internal Controls | Boolean | No | No | Whether the incident was detected by a control operated by the organization. |
| Date of Verification | Date / Time | No | No | Date and time the incident was verified, by an Incident Handler |
| Verified By | Text | Yes | No | The name of the person or system that verified the incident. |
| Date of Containment | Date / Time | No | Yes | Date and time the incident was contained. |
| Date of Recovery | Date / Time | No | Yes | Date and time the affected systems were brought back to a fully operational state. |
| Level of Effort | Number | No | No | Staff-hours for containment and recovery |
| Goss Loss Amount | Number | No | No | Quantifiable, direct financial loss verified by management |
| Business System Downtime | Number | No | No | The number of hours that a business system was unavailable or non-operational (if any); on a per-business system (not per-host) basis. |
| Scope of Incident | Text | No | No | Free-form text comment indicating the scope and size of the incident; for example, the number of hosts, networks, or business units affected by the incident. |
| Affected Systems | Text/Numeric | Yes | No | One-to-many list of the technologies (Technology Reference) and applications (Application ID) directly affected by the incident. These values |

[1] Source: Operational Risk Exchange. <http://www.orx.org/reporting/>

| | | | | |
|---|---|---|---|---|
| | | | | may be reference to application or technology tables. |
| Affected Organizations | Text | Yes | No | One-to-many list of the parts of the organization affected by the incident, for example named business units or functions. These values will vary between organizations. |
| Classifications | Text | Yes | No | One-to-many list of values used to categorize or classify the incident. Can use NIST SP800-61 or other classifications. Incidents may include more than one tag. |
| Root Cause | Text | No | No | Text description of the root cause of the incident. |
| Priority | Text | Yes | No | One-to-many list of values used to indicate the severity or priority of the incident for each affected organization, using a priority classification (links below). Priorities may vary by affected organization. |
| Country of Origination | Text | No | No | The ISO code of the country where the source of the incident resides. |
| Country of Destination | Text | Yes | No | One to many list of the ISO codes of the country where the target company/server(s) reside. |

### Classifications

Tagging of information is a very valuable way to provide context to collected data records. Classification tags provide a way to group incidents. A single incident might fall into one or more categories, so the security incident records management system must support one-to-many tagging capabilities.

Classification tags for security incidents may include NIST incident categories as defined in Special Publication 800-61[2], for example:

- **Denial of service** — an attack that prevents or impairs the authorized use of networks, systems, or applications by exhausting resources
- **Malicious code** — a virus, worm, Trojan horse, or other code-based malicious entity that infects a host
- **Unauthorized access** — a person gains logical or physical access without permission to a network, system, application, data, or other resource
- **Inappropriate usage** — a person violates acceptable computing use policies

### Priority

Priorities for security incidents may include CERT severity levels or priorities as summarized in CERT publication "State of the Practice of Computer Security Incident Response Teams (CSIRTs)"[3]. For example:

- [Kruse 02] — Highest (e-commerce, authentication/billing) to Low (network switch, chat, shell server)
- [Schultz 01] — Level 4 (high-impact affecting many sites) to Level 1 (affects one location)
- [ISS 01] — Severity 5 (penetration or DoS with signification impact on operations) to Severity 1 (low-level probes/scans, known virus)

[2] Scarfone, Grance and Masone. Special Publication 800-61 Revision 1:Computer Security Incident Handling Guide. US National Institute of Standards and Technology, 2004. <http://csrc.nist.gov/publications/nistpubs/800-61-rev1/SP800-61rev1.pdf>

[3] Killcrece, Kossakowski, Ruefle and Zajicek. State of the Practice of Computer Security Incident Response Teams (CSIRTs). Carnegie-Mellon Software Engineering Institute, 2003: p94-96. <http://www.cert.org/archive/pdf/03tr001.pdf>

- [Schultz 90] — Priority 1 (human life, human safety) to Priority 5 (minimize disruption to computing processes)
- [Schiffman 01] —Devilish (extremely skilled, able to cover tracks, leave covert channels) to Low (script kiddie attacks, low innovation)
- [McGlashan 01] — Priority 5 (life and health) to Priority 1 (preservation of non-critical systems)

### Sources

Sources for incident data can come from a variety of sources including incident tracking systems, help desk ticket systems, incident reports, and SIM/SEM systems.

### Dimensions

This metric may include additional dimensions for grouping and aggregation purposes. These dimensions should be applied or tagged at the level of the underlying incident record as described in *Security Incident Metrics: Data Attributes*. For example:

- **Priority** dimension allows metrics to be computed for high, medium, or low severity incidents
- **Classifications** for characterizing types of incidents, such as denial of service, theft of information, etc.
- **Affected Organization** for identifying the affected part of the organization
- **Cause** dimension, such as Missing Patch, Third-Party Access, etc. could be used to improve mitigation effort

### Automation

The ability to automate source data collection for these metrics is **low**, because humans, rather than machines, declare when an incident occurs, is contained and is resolved. Calculation of these metrics on an ongoing basis, after source data has been obtained, lends itself to a **high** degree of automation.

### Visualization

These metrics may be visually represented in several ways:

**Simple visualizations** may include a table showing the metric result for the organization with each row displaying the value as of selected time periods (each week or each month). Columns may be used for different incident classes (e.g. Denial of Service, Unauthorized Access, etc.)

**Graphical visualizations** may include time-series charts where the metric result is plotted on the vertical axis and time periods displayed on the horizontal axis. To provide maximum insight, plotted values for each period may include stacked series for the differing incident classifications.

**Complex visualizations** should be used for displaying the metric result for cross-sections by organization, incident classification, or incident priority. For example, small multiples could be used to compare the number of high priority incidents of unauthorized access across business units or regions.

# Defined Metrics

## Mean-Time-To-Incident-Discovery

### *Objective*

Mean-Time-To-Incident-Discovery (MTTID) characterizes the efficiency of detecting incidents, by measuring the average elapsed time between the initial occurrence of an incident and its subsequent discovery. The MTTID metric also serves as a leading indicator of resilience in organization defenses because it measures detection of attacks from known vectors *and* unknown ones.

Table 3: Mean Time to Incident Discovery

| | |
|---|---|
| **Metric Name** | Mean time to Incident Discovery |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | Mean-Time-To-Incident-Discovery (MTTID) measures the effectiveness of the organization in detecting security incidents. Generally, the faster an organization can detects an incident, the less damage it is likely to incur. MTTID is the average amount of time, in hours, that elapsed between the Date of Occurrence and the Date of Discovery for a given set of incidents. The calculation can be averaged across a time period, type of incident, business unit, or severity. |
| **Audience** | Operations |
| **Question** | What is the average (mean) number of hours between the occurrence of a security incident and its discovery? |
| **Answer** | A positive decimal value that is greater than or equal to zero. A value of “0” indicates hypothetical instant detection. |
| **Formula** | For each record, the time-to-discovery metric is calculated by subtracting the Date of Occurrence from the Date of Discovery. These metrics are then averaged across a scope of incidents, for example by time, category or business unit:<br>$MTTID = \frac{\sum(Date\_of\_Discovery - Date\_of\_Occurrence)}{Count(Incidents)}$ |
| **Units** | Hours per incident |
| **Frequency** | Weekly, Monthly, Quarterly, Annually |
| **Targets** | MTTID values should trend lower over time. The value of “0 hours” indicates hypothetical instant detection times. There is evidence the metric result may be in a range from weeks to months (2008 Verizon Data Breach Report). Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for MTTIDs exist. |
| **Sources** | Since humans determine when an incident occurs, when the incident is contained, and when the incident is resolved, the primary data sources for this metric are manual inputs as defined |

| | |
|---|---|
| | in Security Incident Metrics: Data Attributes. However, these incidents may be reported by operational security systems, such as anti-malware software, security incident and event management (SIEM) systems, and host logs. |

### *Usage*

Mean-Time-To-Incident-Discovery is a type of security incident metric, and relies on the common definition of "security incident" as defined in *Terms in Definitions*.

Optimal conditions would reflect a low value in the MTTID. The lower the value of MTTID, the healthier the security posture is. The higher the MTTID, the more time malicious activity is likely to have occurred within the environment prior to containment and recovery activities. Given the current threat landscape and the ability for malicious code to link to other modules once entrenched, there may be a direct correlation between a higher MTTID and a higher level-of-effort value (or cost) of the incident.

MTTIDs are calculated across a range of incidents over time, typically per-week or per-month. To gain insight into the relative performance of one business unit over another, MTTIDs may also be calculated for cross-sections of the organization, such as individual business units or geographies.

### *Limitations*

This metric measures incident detection capabilities of an organization. As such, the importance of this metric will vary between organizations. Some organizations have much higher profiles than others, and would thus be a more attractive target for attackers, whose attack vectors and capabilities will vary. As such, MTTIDs may not be directly comparable between organizations.

In addition, the ability to calculate meaningful MTTIDs assumes that incidents are, in fact, detected and reported. A lack of participation by the system owners could cause a skew to appear in these metrics. A higher rate of participation in the reporting of security incidents can increase the accuracy of these metrics.

The date of occurrence of an incident may be hard to determine precisely. The date of occurrence field should be the date that the incident could have occurred no later than given the best available information. This date may be subject to revision and more information becomes known about a particular incident.

Mean values may not provide a useful representation of the time to detect incidents if distribution of data exhibits significantly bi-modal or multi-model. In such cases additional dimensions and results for each of the major modes will provide more representative results.

### *References*

Scarfone, Grance and Masone. Special Publication 800-61 Revision 1:Computer Security Incident Handling Guide. US National Institute of Standards and Technology, 2004. <http://csrc.nist.gov/publications/nistpubs/800-61-rev1/SP800-61rev1.pdf>

Killcrece, Kossakowski, Ruefle and Zajicek. State of the Practice of Computer Security Incident Response Teams (CSIRTs). Carnegie-Mellon Software Engineering Institute, 2003. <http://www.cert.org/archive/pdf/03tr001.pdf>

Baker, Hylender and Valentine, 2008 Data Breach Investigations Report. Verizon Business RISK Team, 2008. <http://www.verizonbusiness.com/resources/security/databreachreport.pdf>

## Incident Rate

### *Objective*

Incident Rate indicates the number of detected security incidents the organization has experienced during the metric time period. In combination with other metrics, this can indicate the level of threats, effectiveness of security controls, or incident detection capabilities.

Table 4: Number of Incidents

| | |
|---|---|
| **Metric Name** | Incident Rate |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | Incident Rate measures the number of security incidents for a given time period. |
| **Audience** | Management, Operations |
| **Question** | What is the number of security incidents that occurred during the time period? |
| **Answer** | A non-negative integer value. A value of "0" indicates that no security incidents were identified. |
| **Formula** | To calculate Incident Rate (IR), the number of security incidents are counted across a scope of incidents, for example a given time period, category or business unit:<br>IR = Count(Incidents) |
| **Units** | Incidents per period; for example, incidents per week or incidents per month |
| **Frequency** | Weekly, Monthly, Quarterly, Annually |
| **Targets** | IR values should trend lower over time – assuming perfect detection capabilities. The value of "0" indicates hypothetical perfect security since there were no security incidents. Because of the lack of experiential data from the field, no consensus on range of acceptable goal values for Incident Rate exists. |
| **Sources** | Since humans determine when an incident occurs, when the incident is contained, and when the incident is resolved, the primary data sources for this metric are manual inputs as defined in Security Incident Metrics: Data Attributes. However, these incidents may be reported by operational security systems, such as anti-malware software, security incident and event management (SIEM) systems, and host logs. |

### *Usage*

Number of Incidents is a type of security incident metric and relies on the common definition of "security incident" as defined in *Glossary*.

Optimal conditions would reflect a low number of incidents. The lower the number of incidents, the healthier the security posture would be assuming perfect detection. However, a low number of incidents might also indicate a weak capability to detect incidents. This metric can also indicate the effectiveness of security controls. Assuming similar threat levels and detection capabilities, fewer incidents could indicate better performance of one set of security controls.

The Incident Rate metric is calculated over time, typically per-week or per-month. Not all incidents are easily detected, so the trend of incidents can be useful for indicating patterns in the environment.

To gain insight into the relative performance of one business unit over another, the number of incidents may also be calculated for cross-sections of the organization such as individual business units or geographies.

### *Limitations*

A security program may or may not have direct control over the number of incidents that occur within their environment. For instance, if all the incidents that occur are due to zero-day or previously unidentified attack vectors then there are not many options left to improve posture. However, this metric could be used to show that improving countermeasures and processes within operations to reduce the number of incidents that occur. Thus, Number of Incidents must be considered in the context of other metrics, such as MTTID.

The definition of "Incident" may not be consistently applied across organizations.  For meaningful comparisons, similar definitions are necessary.

The importance of this metric will vary between organizations. Some organizations have much higher profiles than others and would be a more attractive target for attackers whose attack vectors and capabilities will vary. The Number of Incidents may not be directly comparable between organizations.

### *References*

Scarfone, Grance and Masone. Special Publication 800-61 Revision 1:Computer Security Incident Handling Guide. US National Institute of Standards and Technology, 2004. <http://csrc.nist.gov/publications/nistpubs/800-61-rev1/SP800-61rev1.pdf>

Killcrece, Kossakowski, Ruefle and Zajicek. State of the Practice of Computer Security Incident Response Teams (CSIRTs). Carnegie-Mellon Software Engineering Institute, 2003. <http://www.cert.org/archive/pdf/03tr001.pdf>

## Mean Time Between Security Incidents

### *Objective*

Mean Time Between Security Incidents (MTBSI) identifies the relative levels of security incident activity.

Table 5: Mean Time Between Security Incidents

| | |
|---|---|
| **Metric Name** | Mean Time Between Security Incidents |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | Mean Time Between Security Incidents (MTBSI) calculates the average time, in days, between security incidents. This metric is analogous to the Mean Time Between Failure (MTBF) metric found in break-fix processes for data center. |
| **Audience** | Operations |
| **Question** | For all security incidents that occurred within a given time period, what is the average (mean) number of days between incidents? |
| **Answer** | A floating-point value that is greater than or equal to zero. A value of “0” indicates instantaneous occurrence of security incidents. |
| **Formula** | For each record, the mean time between incidents is calculated by dividing the number of hours between the time on the Date of Occurrence for the current incident from the time on the Date of Occurrence of the previous incident by the total number of incidents prior to the current incident:<br>$MTBSI = \frac{\sum(Date\_of\_Occurence[Incident_n] - Date\_of\_Occurence[Incident_{n-1}])}{Count(Incidents)}$ |
| **Units** | Hours per incident interval |
| **Frequency** | Weekly, Monthly, Quarterly, Annually |
| **Targets** | MTBSI values should trend higher over. The value of “0” indicates hypothetical instantaneous occurrence between security incidents. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Mean Time Between Security Incidents exists. |
| **Sources** | Since humans determine when an incident occurs, when the incident is contained, and when the incident is resolved, the primary data sources for this metric are manual inputs as defined in Security Incident Metrics: Data Attributes. However, these incidents may be reported by operational security systems, such as anti-malware software, security incident and event management (SIEM) systems, and host logs. |

### Usage

This metric provides an indication of activity within the environment. A higher value for this metric might indicate a less-active landscape. However, an inactive landscape might be caused by a lack of reporting or a lack of detection of incidents.

### Limitations

The date of occurrence of an incident may be hard to determine precisely.  The date of occurrence field should be the date that the incident could have occurred.  This date may be subject to revision as more information becomes known about a particular incident.

### References

Scarfone, Grance and Masone. Special Publication 800-61 Revision 1:Computer Security Incident Handling Guide. US National Institute of Standards and Technology, 2004. <http://csrc.nist.gov/publications/nistpubs/800-61-rev1/SP800-61rev1.pdf>

Killcrece, Kossakowski, Ruefle and Zajicek. State of the Practice of Computer Security Incident Response Teams (CSIRTs). Carnegie-Mellon Software Engineering Institute, 2003. <http://www.cert.org/archive/pdf/03tr001.pdf>

## Mean Time to Incident Recovery

### *Objective*

Mean Time to Incident Recovery (MTIR) characterizes the ability of the organization to return to a normal state of operations. This is measured by the average elapse time between when the incident occurred to when the organization recovered from the incident.

Table 6: Mean Time to Incident Recovery

| | |
|---|---|
| **Metric Name** | Mean Time to Incident Recovery |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | Mean Time to Incident Recovery (MTIR) measures the effectiveness of the organization to recovery from security incidents. The sooner the organization can recover from a security incident, the less impact the incident will have on the overall organization. This calculation can be averaged across a time period, type of incident, business unit, or severity. |
| **Audience** | Management, Operations |
| **Question** | What is the average (mean) number of hours from when an incident occurs to recovery? |
| **Answer** | A positive integer value that is greater than or equal to zero. A value of “0” indicates instantaneous recovery. |
| **Formula** | Mean time-to-incident recovery is calculated by dividing the difference between the Date of Occurrence and the Date of Recovery for each incident recovered in the metric time period, by the total number of incidents recovered in the metric time period<br>$MTIR = \frac{\sum(Date\_of\_Recovery - Date\_of\_Occurrence)}{Count(Incidents)}$ |
| **Units** | Hours per incident |
| **Frequency** | Weekly, Monthly, Quarterly, Annually |
| **Targets** | MTIR values should trend lower over time. There is evidence the metric result will be in a range from days to weeks (2008 Verizon Data Breach Report). The value of “0” indicates hypothetical instantaneous recovery. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Mean Time to Incident Recovery exists. |
| **Sources** | Since humans determine when an incident occurs, when the incident is contained, and when the incident is resolved, the primary data sources for this metric are manual inputs as defined in Security Incident Metrics: Data Attributes. However, these incidents may be reported by operational security systems, such as anti-malware software, security incident and event management (SIEM) systems, and host logs. |

### *Usage*

MTIR is a type of security incident metric and relies on the common definition of "security incidents" as defined in Glossary.

Optimal conditions would reflect a low value in the MTIR. A low MTIR value indicates a healthier security posture as the organization quickly recovered from the incident. Given the impact that an incident can have on an organization's business processes, there may be a direct correlation between a higher MTIR and a higher incident cost.

### *Limitations*

This metric measures incident recovery capabilities of an organization. As such, the importance of this metric will vary between organizations. Some organizations have much higher profiles than others and would be a more attractive target for attackers whose attack vectors and capabilities vary. MTIRs may not be directly comparable between organizations.

The date of occurrence of an incident may be hard to determine precisely. The date of occurrence field should be the date that the incident could have occurred. This date may be subject to revision and more information becomes known about a particular incident.

### *References*

Scarfone, Grance and Masone. Special Publication 800-61 Revision 1:Computer Security Incident Handling Guide. US National Institute of Standards and Technology, 2004. <http://csrc.nist.gov/publications/nistpubs/800-61-rev1/SP800-61rev1.pdf>

Killcrece, Kossakowski, Ruefle and Zajicek. State of the Practice of Computer Security Incident Response Teams (CSIRTs). Carnegie-Mellon Software Engineering Institute, 2003. <http://www.cert.org/archive/pdf/03tr001.pdf>

Baker, Hylender and Valentine, 2008 Data Breach Investigations Report. Verizon Business RISK Team, 2008. <http://www.verizonbusiness.com/resources/security/databreachreport.pdf>

# Vulnerability Management

This section describes metrics for measuring the process used for the identification and management of vulnerabilities within an organization's environment.

As described in the *Glossary* section of this document, a *vulnerability* is a flaw or misconfiguration that causes a weakness in the security of a system that could be exploited. Sources of vulnerabilities include new systems or applications introduced to the organization's environment or the discovery of new vulnerabilities on existing systems and applications. Vulnerability management is a vital part of keeping an organization's assets safe; identifying and mitigating weaknesses found on systems and applications reduces the risk of negatively impacting the business should these vulnerabilities be exploited. It consists of the following high-level process steps:

- Vulnerability Notification through becoming aware of disclosed vulnerabilities and performing security assessments.
- Vulnerability Identification through manual or automated scanning of technologies throughout the organization.
- Vulnerability Remediation & Mitigation through application of patches, adjustment of configurations, modification of systems, or acceptance of risk.

The primary question this activity is concerned with is: "Are my systems safe?" In vulnerability management terms this question can be decomposed to: "*Are there vulnerable systems? Have systems been checked, and if so, what was found?*"

## Data Attributes

Vulnerability metrics are comprised of the following datasets:

**Technologies**. Contains information about the technologies in the organization's environment. Technologies should be identified and named according to the Common Product Enumeration Dictionary maintained by NIST (http://nvd.nist.gov/cpe.cfm).

**Vulnerability Information**. Contains information about the vulnerability, such as its severity and classification, as denoted by the National Vulnerability Database (http://nvd.nist.gov/) or other source.

**Identified Vulnerabilities**. Contains the set of vulnerability instances identified in the organization's environment for the metric time period (this can be a larger set that is filtered by scan date).

### Technologies

The following is a list of attributes that should be populated as completely as possible for each technology within the organization:

Table 7: Technologies Table

Technologies Table

| Name | Type | De-identified | Required | Description |
|---|---|---|---|---|
| Technology ID | Number | No | Yes | Unique identifier for the technology. |

| | | | | |
|---|---|---|---|---|
| | | | | Generally auto-generated. |
| Name | Text | No | No | [CPE Field] Name from CPE Dictionary. |
| Vendor | Text | No | No | [CPE Field] Vendor from CPE Dictionary. |
| Product | Text | No | No | [CPE Field] Product from CPE Dictionary |
| Version | Text | No | No | [CPE Field] Version from CPE Dictionary. |
| Technology Value | Text | No | Recommended | Impact from the loss of this technology (C/I/A) to the organization. Uses value *Low, Medium, High,* or *Not Defined*. [4] |
| Under Patch Management | Boolean | No | Yes | This is a flag to use with Patch Management metrics. |
| Dimensions/Tags | Text/Drop-Down List | No | No | Business Unit, Technology class, geographical area |

## Vulnerability Information

This is a table of information about known vulnerabilities, such as affected versions, severities, and references. The NVD will be the reference database, and CVSS v2 the reference severity rating system. Vendors of vulnerability identification systems may also enhance or expand both the listing and specifications of known vulnerabilities. The following is a list of attributes that should be populated as completely as possible for each vulnerability:

**Table 8: Vulnerability Information Table**

| Vulnerability Information Table | | | | |
|---|---|---|---|---|
| **Name** | **Type** | **De-identified** | **Required** | **Description** |
| Vulnerability ID | Number | No | Yes | Unique identifier for the vulnerability. Generally auto-generated. This can be an organization-specific identifier for the vulnerability. |
| CVE id | Number | No | No | Common Vulnerability Enumeration identifier for this vulnerability. |
| cwe-id | Number | No | No | Common Weakness Enumeration id for the weakness associated with this vulnerability |
| Description | Number | No | No | Text description of the vulnerability (from |

[4] This is adopting 2.3.3 Security Requirements Scoring Evaluation from CVSS v2, http://www.first.org/cvss/cvss-guide.html#i2.3.

| | | | | |
|---|---|---|---|---|
| | | | | NVD or elsewhere) |
| CVSS Base Score | Number | No | Recommended | [CVSS Field] CVSS Base Score |
| CVSS Base Score String | Text | No | No | [CVSS Field] CVS Base Score string (optional) |
| Impact Subscore | Number | No | No | [CVSS Field] CVSS Impact Subscore |
| Exploitability Subscore | Number | No | No | [CVSS Field] CVSS Exploitability Subscore |
| Access Vector | Text | No | No | [CVSS Field] CVSS classification of how the vulnerability is exploited. Uses values of *Local*, *Adjacent Network*, or *Network*. |
| Access Complexity | Text | No | No | [CVSS Field] CVSS rating of the complexity of the attack required to exploit the vulnerability. Uses values of *High*, *Medium*, or *Low*. |
| Authentication | Text | No | No | [CVSS Field] CVSS rating of the number of times an attacker must authenticate to exploit a vulnerability. Uses value s of *Single, Multiple,* or *None*. |
| Impact Type | Text | No | No | [CVSS Field] Description text of vulnerability impact |
| Vulnerability Type | Text | No | No | [CVSS Field] Type of vulnerability (CWE cross section used by NVD) |
| Original Release Date | Date / Time | No | No | [CVSS Field] Date that the vulnerability was made publicly known. |
| Exploitability | Text | No | No | [CVSS Field] CVSS current state of exploit techniques. Uses value *Unproven*, *Proof-of-Concept*, *Functional*, *High*, or *Not Defined*. |
| Remediation Level | Text | No | No | [CVSS Field] CVSS stage of the remediation lifecycle. Uses value *Official Fix*, *Temporary Fix*, *Workaround*, *Unavailable*, or *Not Defined*. |
| Report Confidence | Text | No | No | [CVSS Field] CVSS degree of confidence in the existence of the vulnerability. Uses value *Unconfirmed*, *Uncorroborated*, *Confirmed*, or *Not Defined*. |

| | | | | |
|---|---|---|---|---|
| Collateral Damage Potential | Text | No | No | [CVSS Field] Potential for loss through damage or theft of the asset. Uses values of *None*, *Low*, *Low-Medium*, *Medium-High*, *High*, or *Not Defined*. |
| Target Distribution | Text | No | No | [CVSS Field] Proportion of vulnerable systems. Uses value *None*, *Low*, *Medium*, *High*, or *Not Defined*. |
| cvss-generated-on | Date/Time | No | No | [CVSS Field] Date and time the cvss score was generated |

## Identified Vulnerabilities

This table represents information regarding vulnerability instances on technologies. The following is a list of attributes that should be populated as completely as possible for the current set of vulnerability instances identified on technologies within the organization:

**Table 9: Identified Vulnerabilities Table**

### Identified Vulnerabilities Table

| Name | Type | De-identified | Required | Description |
|---|---|---|---|---|
| Vulnerability Instance ID | Number | No | Yes | Unique identifier for the vulnerability instance. Generally auto-generated |
| Vulnerability Reference ID | Number | No | Yes | Reference to the Vulnerability in the Vulnerability Information Table |
| Technology Reference | Number | Yes | Recommended | Reference in the Technologies Table to the specific technology with this vulnerability instance. |
| Date of Detection | Date/Time | No | Yes | Date and time when the vulnerability was initially detected |
| Detection Method | Text | No | No | *Vulnerability Scanner Name / Version* or *Manual Detection* |
| Date of Remediation | Date/Time | No | No | Date and time when the vulnerability was remedied. |
| Vulnerability Status | Text | No | Yes | Current status of the vulnerability instance. Uses values of *Open, Not Valid* or *Mitigated*.<br>Vulnerabilities should be flagged Open by default. |

| | | | | |
|---|---|---|---|---|
| Instance Impact Subscore | Number | No | No | CVSS Impact Subscore of this vulnerability instance. An instance-specific value can be used by the organization for internal metrics. |
| Instance Exploitability Subscore | Number | No | No | CVSS Exploitability Subscore of this vulnerability instance. An instance-specific value can be used by the organization for internal metrics. |
| Instance Collateral Damage Potential | Text | No | No | Potential for loss through damage or theft of the asset resulting from this instance of the vulnerability. Uses values of *None*, *Low*, *Low-Medium*, *Medium-High*, *High*, or *Not Defined*. An instance-specific value can be used by the organization for internal metrics. |

### Classifications and Dimensions

Tagging of information is a very valuable way to provide context to collected data records. Classification tags provide a way to group vulnerabilities. Currently, the only classification used is the severity of the vulnerability. In the future, vulnerabilities can be grouped by other categories, such as vulnerability type or source of the vulnerability.

It is expected that dimensions will be added to these tables to provide the ability to view metric results that address key questions and concerns. Examples of dimensions include:

- **Technologies**: business unit, geography, business value, or technology category by technology
- **Vulnerability Information**: vulnerability severity, classification, or vendor
- **Identified Vulnerabilities**: remediation status, identification date, environment-specific severity

Within an organization, the combination of dimensions can provide key insight into their concentrations of risk.

### Severity of Vulnerabilities

Severity ratings are determined by the CVSS v2 scoring system and can commonly be found in reference systems such as the National Vulnerability Database (NVD). Severity ratings for vulnerabilities are along several dimensions with Base Scores derived from exploitability factors (such as attack complexity) and impact factors (such as integrity impact). CVSS Base scores can be expressed in a 0-10 range, commonly summarized as:

- "Low" severity if they have a CVSS base score of 0.0-3.9
- "Medium" severity if they have a CVSS base score of 4.0-6.9
- "High" severity if they have a CVSS base score of 7.0-10.0

The severity of a specific vulnerability instance in an organization can be more accurately determined by combining environment and temporal factors with the base score. Metrics can be generated using organization-specific values in place of external values for fields such as vulnerability impact or exploitability scores to account for an organization's specific environment. These calculations are beyond the current scope of these metrics.

### Technology Value (CTV, ITV, ATV)

Technology values will be rated by adopting the Common Vulnerability Scoring System (v2) section 2.3.3 Security Requirements Scoring Evaluation ratings. These Technology Value scores can be used independently as well as used for the complete scoring of a vulnerability that affected the technology. Each technology is assigned one of three possible values, "Low", "Medium", "High" (or Not Defined) depending on the impact from loss of confidentiality (CTV), integrity (ITV), or availability (ATV). These ratings are reproduced here:

- Low (L). Loss of [confidentiality | integrity | availability] is likely to have only a limited adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).
- Medium (M). Loss of [confidentiality | integrity | availability] is likely to have a serious adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).
- High (H). Loss of [confidentiality | integrity | availability] is likely to have a catastrophic adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).
- Not Defined (ND). Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric.

As described in CVSS v2, these values should be based on network location, business function, and the potential for loss of revenue of life. No specific methodology is defined to assign these values.

### Sources

The primary data source for both systems scanned and vulnerabilities identified on systems will be network scanning and vulnerability identification tools. Generally a list of all discovered and scanned systems can be extracted from vulnerability scanning systems and compared to reports of all systems with identified vulnerabilities. The totals of all systems in the organization can come from asset management systems and/or network discovery scans.

### Dimensions

These metrics may include additional dimensions for grouping and aggregation purposes. These dimensions should be applied or tagged at the level of the technology record as described in *Vulnerability Management Metrics: Data Attributes.* For example:

- **Technology Value** allows the metric to be computed for high, medium, or lower value technologies.
- **Remediation Level** of the vulnerability allows metrics to be computed around the current state of vulnerabilities and remediation efforts
- **Tags** for characterizing types of technologies, such as coverage by vendor, etc.
- **Business Units** for identifying the concentrations of risk across different parts of the organization
- **Severity** of the vulnerabilities is a dimension that should be used. While CVSS Base Score uses a scale of 1-10, this is generally summarized into low, medium, and high severity vulnerabilities. Generally many low severity vulnerabilities are found.

### Automation

The ability to automate source data collection for these metrics is **high** because most automated vulnerability identification systems can provide the necessary reports in combination with asset tracking and/or discovery scans providing counts of all technologies. Calculation of these metrics is on an ongoing basis. Once source data has been obtained, it lends itself to a **high** degree of automation.

### Visualization

These metrics may be visually represented in several ways:

**Simple visualizations** may include a table showing the metric result for the organization with each row displaying the value as of selected time periods (each week or each month). Columns may be used for different vulnerability severities (e.g. Low, Medium, High).

**Graphical visualizations** may include time-series charts where the metric result is plotted on the vertical axis and time periods displayed on the horizontal axis. To provide maximum insight, plotted values for each period may include stacked series for the differing severity values.

**Complex visualizations** should be used for displaying the metric result for cross-sections by organization, vulnerabilities, or technology values. For example, small multiples could be used to compare the number of high severity vulnerabilities across business units or technology values.

# Defined Metrics

## Vulnerability Scan Coverage

### *Objective*

Vulnerability Scan Coverage (VSC) indicates the scope of the organization's vulnerability identification process. Scanning of systems known to be under the organization's control provides the organization the ability to identify open known vulnerabilities on their systems. Percentage of systems covered allows the organization to become aware of areas of exposure and proactively remediate vulnerabilities before they are exploited.

Table 10: Vulnerability Scan Coverage

| | |
|---|---|
| **Metric Name** | Vulnerability Scan Coverage |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | Vulnerability Scanning Coverage (VSC) measures the percentage of the organization's systems under management that were checked for vulnerabilities during vulnerability scanning and identification processes. This metric is used to indicate the scope of vulnerability identification efforts. |
| **Audience** | Management, Operations |
| **Question** | What percentage of the organization's total systems has been checked for known vulnerabilities? |
| **Answer** | Positive integer value that is greater than or equal to zero but less than or equal to 100%. A value of "100%" indicates that all systems are covered by the vulnerability scanning process. |
| **Formula** | Vulnerability Scanning Coverage is calculated by dividing the total number of systems scanned by the total number of systems within the metric scope such as the entire organization:<br>$VSC = \frac{Count(Scanned\_Systems)}{Count(All\_Systems\_Within\_Organization)} * 100$ |
| **Units** | Percentage of systems |
| **Frequency** | Weekly, Monthly, Quarterly, Annually |
| **Targets** | VSC values should trend higher over time. Higher values are obviously better as it means more systems have been checked for vulnerabilities. A value of 100% means that all the systems are checked in vulnerability scans. For technical and operational reasons, this number will likely be below the theoretical maximum. |

### *Usage*

This metric provides information about how much of the organization's environment is checked for known vulnerabilities. Organizations can use this metric to evaluate their risk position in terms of concentrations of unknown vulnerability states of systems. In combination with other vulnerability metrics, it provides insight on the organization's exposure to known vulnerabilities.

The results of the coverage metric indicate the:

- Scope of the vulnerability scanning activities
- Applicability of other metric results across the organization
- Relative amount of information known about the organization's vulnerability

### *Limitations*

Due to technical or operational incompatibility certain systems may be excluded from scanning activities while other systems such as laptops and guest systems may be intermittently present for network scans, resulting in variability of metric results. In addition, scanning activities can vary in depth, completeness, and capability.

This metric assumes that systems scanned for vulnerabilities are systems known to and under full management by the organization. These systems do not include partial or unknown systems. Future risk metrics may account for these to provide a clearer view of all system ranges.

### *References*

ISO/IEC 27002:2005

Mell, Bergeron and Henning. Special Publication 800-40: Creating a Patch and Vulnerability Management Program. US National Institute of Standards and Technology, 2005.

## Percent of Systems Without Known Severe Vulnerabilities

### *Objective*

Percent of Systems Without Known Severe Vulnerabilities (PSWKSV) measures the organization’s relative exposure to known severe vulnerabilities. The metric evaluates the percentage of systems scanned that do not have any known high severity vulnerabilities.

Table 11: Percentage of Systems Without Known Severe Vulnerabilities

| | |
|---|---|
| **Metric Name** | Percent of Systems Without Known Severe Vulnerabilities |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | Percent of Systems Without Known Severe Vulnerabilities (PSWKSV) measures the percentage of systems that when checked were not found to have any known high severity vulnerabilities during a vulnerability scan. Vulnerabilities are defined as “High" severity if they have a CVSS base score of 7.0-10.0<br>Since vulnerability management involves both the identification of new severe vulnerabilities and the remediation of known severe vulnerabilities, the percentage of systems without known severe vulnerabilities will vary over time. Organizations can use this metric to gauge their relative level of exposure to exploits and serves as a potential indicator of expected levels of security incidents (and therefore impacts on the organization).<br>This severity threshold is important, as there are numerous informational, local, and exposure vulnerabilities that can be detected that are not necessarily material to the organization’s risk profile. Managers generally will want to reduce the level of noise to focus on the greater risks first. This metric can also be calculated for subsets of systems, such as by asset criticality of business unit |
| **Audience** | Management, Operations |
| **Question** | Of the systems scanned, what percentage does not have any known severe vulnerabilities? |
| **Answer** | A positive integer value that is greater than or equal to zero. A value of “100%” indicates that none of the organization’s systems have any known high severity vulnerabilities. |
| **Formula** | Percent of Systems Without Known Severe Vulnerabilities is calculated by counting those systems that have no open high severity level vulnerabilities (Vulnerability Status !=“Open” & CVSS Base Score >= 7.0). This result is then divided by the total number of systems in the scanning scope.<br>$PSWKSV = \frac{Count(Systems\_Without\_Known\_Severe\_Vulnerabilities)}{Count(Scanned\_Systems)} * 100$ |
| **Units** | Percentage of systems |

| | |
|---|---|
| **Frequency** | Weekly, Monthly, Quarterly, Annually |
| **Targets** | PSWKSV values should trend higher over time. It would be ideal to have no known severe vulnerabilities on systems; therefore, an ideal target value would be 100%. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Percent of Systems Without Known Severe Vulnerabilities exists. |

### *Usage*

Percent of Systems Without Known Severe Vulnerabilities is a type of vulnerability management metric and relies on the common definition of "vulnerability" as defined in the Glossary. Due to the number of vulnerabilities and exposures found by most scanning tools, this metric should be calculated for "High" severity vulnerabilities.

Optimal conditions would reflect a high value in the metric. A value of 100% would indicate that none of the organizations systems are known to possess severe vulnerabilities. The lower the value, the greater the risk that systems are exploited. Since many attacks are designed to exploit known severe vulnerabilities there may be a direct correlation between a higher percentage of vulnerable systems and the number of security incidents.

Percent of Systems Without Known Severe Vulnerabilities can be calculated over time, typically per-week or per-month. To gain insight into the relative performance and risk to one business unit over another, the metric may also be calculated for cross-sections of the organization such as individual business units or geographies.

### *Limitations*

Due to technical or operational incompatibility certain systems may be excluded from scanning activities while other systems such as laptops may be intermittently present for network scans. Systems not scanned, even if they possess severe vulnerabilities will not be included in this metric result. In addition, scanning activities can vary in depth, completeness, and capabilities.

This metric assumes that systems scanned for vulnerabilities are systems known to and under full management by the organization. These systems do not include partial or unknown systems. Future risk metrics may account for these to provide a clearer view of all system ranges.

### *References*

ISO/IEC 27002:2005

Mell, Bergeron and Henning. Special Publication 800-40: Creating a Patch and Vulnerability Management Program. US National Institute of Standards and Technology, 2005.

## Mean-Time to Mitigate Vulnerabilities

### *Objective*

Mean-Time to Mitigate Vulnerabilities (MTTMV) measures the average amount of time required to mitigate an identified vulnerability. This metric indicates the performance of the organization in reacting to vulnerabilities identified in the environment. It only measures the time average times for explicitly mitigated vulnerabilities, and not mean time to mitigate any vulnerability, or account for vulnerabilities that no longer appear in scanning activities.

Table 12: Mean-Time to Mitigate Vulnerabilities

| | |
|---|---|
| **Metric Name** | Mean-Time to Mitigate Vulnerabilities |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | Mean-Time to Mitigate Vulnerabilities measures the average time taken to mitigate vulnerabilities identified in an organization's technologies. The vulnerability management processes consists of the identification and remediation of known vulnerabilities in an organization's environment. This metric is an indicator of the performance of the organization in addressing identified vulnerabilities. The less time required to mitigate a vulnerability the more likely an organization can react effectively to reduce the risk of exploitation of vulnerabilities.<br>It is important to not that only data from vulnerabilities explicitly mitigated are is included in this metric result. The metric result is the mean time to mitigate vulnerabilities that are actively addressed during the metric time period, and not a mean time to mitigate based on the time for all known vulnerabilities to be mitigated. |
| **Audience** | Operations |
| **Question** | How long does it take the organization to mitigate a vulnerability? |
| **Answer** | A positive floating-point value that is greater than or equal to zero. A value of "0" indicates that vulnerabilities were instantaneously mitigated. |
| **Formula** | Mean-Time to Mitigate Vulnerabilities is calculated by determining the number of hours between the date of detection and the Date of Mitigation for each identified vulnerability instance in the current scope, for example, by time period, severity or business unit. These results are then averaged across the number of mitigated vulnerabilities in the current scope:<br>$MTTMV = \frac{\sum(Date\_of\_Mitigation - Date\_of\_Detection)}{Count(Mitigated\_Vulnerabilities)}$ |
| **Units** | Hours per vulnerability |

| | |
|---|---|
| **Frequency** | Weekly, Monthly, Quarterly, Annually |
| **Targets** | MTTMV values should trend lower over time. Lower levels of MTTMV are preferred. Most organizations put mitigation plans through test and approval cycles prior to implementation. Generally, the target time for MTTMV will be a function of the severity of the vulnerability and business criticality of the technology. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Mean Time to Mitigate Vulnerabilities exists. |

### *Usage*

Mean-Time to Mitigate Vulnerabilities is a type of vulnerability management metric and relies on the common definition of "vulnerability" as defined in the Glossary. Due to the number of vulnerabilities and exposures found by most scanning tools, this metric should generally be calculated for "High" and "Medium" severity vulnerabilities. Combined with the number of identified vulnerabilities this metric can provide visibility into the time and effort required to manage the known vulnerabilities in the organization.

Optimal conditions would reflect a low value in the metric. The lower the value the more quickly the organization is able to react to and mitigate identified vulnerabilities. Since many attacks are designed to exploit known vulnerabilities there may be a direct correlation between a lower time to mitigate vulnerabilities and the number of security incidents.

MTTV can be calculated over time, typically per-month. To gain insight into the relative performance and risk , this metric can be calculated for vulnerabilities with differing severity levels, as well as calculated for cross-sections of the organization such as individual business units or geographies.

### *Limitations*

Only data from mitigated vulnerabilities are included in this calculation. Therefore it is an indicator of the organization's ability to mitigate vulnerabilities as they are identified, but not necessarily a true representation of the average time taken to mitigate all vulnerabilities that may exist in the organization's environment. Other indicators of the scale of scope of unmitigated vulnerabilities should also be used to assess the performance of the vulnerability management function.

Mitigation effort can vary depending on the scope and depth of the mitigation solution, modification of firewall rules or other changes to the environment may be less effort than directly addressing vulnerabilities in an application's code. It is possible that the vulnerabilities that are easier to mitigate are the ones completed in the metric scope, and the remaining vulnerabilities represent the most challenging to mitigate. Therefore the metric result could be biased low compared the to mean time to mitigate remaining known vulnerabilities.

### *References*

ISO/IEC 27002:2005

Mell, Bergeron and Henning. Special Publication 800-40: Creating a Patch and Vulnerability Management Program. US National Institute of Standards and Technology, 2005.

## Number of Known Vulnerability Instances

### *Objective*

Number of Known Vulnerability Instances (NKVI) measures the total number of instances of known vulnerabilities within an organization among scanned assets based on the scanning process at a point in time.

**Table 13: Number of Known Vulnerability Instances**

| | |
|---|---|
| **Metric Name** | Number of Known Vulnerability Instances |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | Number of Known Vulnerability Instances (NKVI) measures the number of known vulnerabilities have been found on an organization's systems during the vulnerability identification process. |
| **Audience** | Operations |
| **Question** | How many open vulnerability instances were found during the scanning process? |
| **Answer** | A positive integer value that is greater than or equal to zero. A value of "0" indicates that no instances of known vulnerabilities were found. |
| **Formula** | This metric is calculated by counting the number of open vulnerability instances identified. This count should also be done for each severity value (Low, Medium, and High):<br>*Number of Known Vulnerabilities* = *Count(Vulnerability Status=Open)* |
| **Units** | Number of Vulnerabilities |
| **Frequency** | Weekly, Monthly, Quarterly, Annually |
| **Targets** | NKVI values should trend lower over time. In the ideal case, there would be no known vulnerability instances on any technologies in the organization. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Number of Known Vulnerability Instances exists. |

### *Usage*

By understanding the number of instances of known exploitable vulnerabilities, the organization can assess relative risk levels across the organization of time, estimate and management remediation efforts, and correlate and predict the volume of security incidents.

The vulnerability scanning process can consist of a number of vulnerability scanning activities occurring over a set time period in cases where multiple scans are necessary to cover all of an organization's technologies or potential vulnerability types.

This metric should be used in conjunction with other vulnerability metrics to provide context around the magnitude of known vulnerabilities in an organization.  Since other metrics are expressed as ratios, this metric quantifies the volume of known vulnerabilities the organization is managing.  Combined with the mean time to mitigate vulnerabilities this metric can provide visibility into the time and effort required to manage the known vulnerabilities in the organization.

When comparing performance over time and between organizations, this metric can be normalized across the total number of systems.  This and additional vulnerability metrics are an area noted for further development by the CIS metrics community.

### *Limitations*

The vulnerability scans may not be comprehensive, instead only attempting to identify a subset of potential vulnerabilities.  Different scanning sessions and products can be checking for different numbers and types of vulnerabilities, some may consist of thousands of checks for vulnerabilities, while other products or sessions may only check for hundreds of known vulnerabilities.

The scope of the scanning effort may not be complete and may also not be representative of the organizations overall systems.  Those systems out of scope may potentially be areas of risk.  In some cases key servers or production systems may be excluded from scanning activities.

This metric only reports on known vulnerabilities.  This does not mean that there are no “unknown” vulnerabilities. Severe vulnerabilities that the organization is unaware of can exist, and potentially be exploited, for years before any public disclosure may occur.

When reporting a total number of vulnerabilities, severe vulnerabilities are considered equal to informational vulnerabilities. Reporting this metric by the dimension of Vulnerability Severity will provide more actionable information.

### *References*

ISO/IEC 27002:2005

Mell, Bergeron and Henning. Special Publication 800-40: Creating a Patch and Vulnerability Management Program. US National Institute of Standards and Technology, 2005.

# Patch Management

This section describes metrics for measuring the effectiveness of patch management processes.

Many security incidents are caused by exploitation of known vulnerabilities for which patches are available. Patches are released by vendors on regular and ad-hoc schedules and the cycle of testing and deploying patches is a regular part of an organization's IT activities. Many patches are released to directly address security issues in applications and operating systems and the performance of the patch management process will directly affect the security posture of the organization.

These metrics are based upon a patching management process with the following structure:

1. Security and Patch Information Sources
2. Patch Prioritization and Scheduling
3. Patch Testing
4. Configuration (Change) Management
5. Patch Installation and Deployment
6. Patch Verification and Closure

## Data Attributes

Patch metrics are comprised of the following datasets:

**Technologies**. Contains information about the technologies in the organization's environment. Technologies should be identified and named according to the Common Product Enumeration Dictionary maintained by NIST (http://nvd.nist.gov/cpe.cfm).

**Patch Information**. This table contains information about the patch, such as the release date, vendor references, vulnerability references, etc. The Open Vulnerability and Assessment Language (OVAL) Repository[5] provides a structured data source of patch information that can be used for this purpose.

**Patch Activity**. This table contains local information about specific patch deployments in an environment, such as the number of systems patched, patch installation date, etc.

### Technologies

The following is a list of attributes that should be populated as completely as possible for each technology:

Table 14: Technologies Table

Technologies Table

| Name | Type | De-identified | Required | Description |
|---|---|---|---|---|

[5] http://oval.mitre.org/repository/index.html

| | | | | |
|---|---|---|---|---|
| Technology ID | Number | No | Yes | Unique identifier for the technology. Generally auto-generated. |
| Name | Text | No | No | [CPE Field] Name from CPE Dictionary. |
| Vendor | Text | No | No | [CPE Field] Vendor from CPE Dictionary. |
| Product | Text | No | No | [CPE Field] Product from CPE Dictionary |
| Version | Text | No | No | [CPE Field] Version from CPE Dictionary. |
| Technology Value | Text | No | Recommended | Impact from the loss of this technology (C/I/A) to the organization. Uses value *Low, Medium, High,* or *Not Defined*. [6] |
| Under Patch Management | Boolean | No | Yes | Indicator flag that the technology is managed by a patch management system. This is a flag for use with Patch Management metrics. |
| Patch Policy Compliance | Boolean | No | Yes | Indicator flag that the technology is compliant with patch policy (required patches installed). This is a flag for use with Patch Management metrics. |
| Dimensions/Tags | Text/Drop-Down List | No | No | Business Unit, Technology class, geographical area |

## Patch Information

The following is a list of attributes that should be populated as completely as possible for each patch:

**Table 15: Patch Information Table**

### Patch Information Table

| Name | Type | De-identified | Required | Description |
|---|---|---|---|---|
| Patch ID | Number | No | Yes | Unique identifier for the patch. Generally auto-generated. This can be an organization-specific identifier for the patch. |
| Patch Source | Text | No | No | The name of the vendor or group issuing the patch |

[6] This is adopting 2.3.3 Security Requirements Scoring Evaluation from CVSS v2, http://www.first.org/cvss/cvss-guide.html#i2.3.

| | | | | |
|---|---|---|---|---|
| Patch Name | Text | No | No | The name of the patch. |
| Technology | Text/Numeric | No | No | Name or Reference ID of the technology the patch applies to. |
| Vulnerability References | Number | No | No | One to many references to vulnerabilities in NVD addressed by this patch |
| Criticality Level | Text | No | Yes | Level of criticality as determined by the classification process, typically High, Medium, or Low. |
| Organization-Specific Criticality Level | Text | No | Yes | Level of criticality as determined by the organization. This may be distinct from a vendor or community determined patch criticality. |
| Date of Notification | Date/Time | No | No | Date and time when the patch notification was first received. Generally this should be the release date of the patch. |
| Date of Availability | Date/Time | No | Yes | Date and time when the patch was released. |
| Date of Patch Approval | Date/Time | No | No | Date and time when the patch was approved by the organization for deployment. |

## Patch Activity

The following is a list of attributes that should be populated as completely as possible for each patch deployed in the environment. Some organizations may wish to track patch activity with greater granularity, at the level of each patch instance. In this case, the same table structure can be used, with the number of "Technology Instances" and "Patch Instances" being '1' for each row.

**Table 16: Patch Activity Table**

| Patch Activity Table | | | | |
|---|---|---|---|---|
| **Name** | **Type** | **De-identified** | **Required** | **Description** |
| Patch Instance ID | Number | No | Yes | Unique identifier for the patch instance. Generally auto-generated |
| Patch Reference ID | Number | No | Yes | Reference to the Patch in the Patch Information Table |
| Technology | Numeric | Yes | Yes | Number of instances of a specific technology. This is |

| | | | | |
|---|---|---|---|---|
| Instances | | | | a count of all the technologies to which this patch applies. |
| Patch Instances | Numeric | Yes | Yes | Number of instances of the patch installed. This is a count of the number of successful patch attempts made on the technology instances. |
| Date of Installation | Date/Time | No | Yes | Date and time when the patch was installed (including any rebooting or reloading process). |
| Patch Complete | Boolean | No | Yes | Flag indicating whether or not the patch was installed. |
| Patch Cost | Numeric | No | No | Cost of the patch deployment (USD) |
| Patch Effort | Numeric | No | No | Total person-hours of effort for the patch deployment. |

### Classifications

Tagging of information is a very valuable way to provide context to collected data records. Classification tags provide a way to group patches. While currently the only classification is the criticality of the patch, in the future, patches may fall into one or more categories, so the patch management record system should support one-to-many tagging capabilities.

### Criticality of Patches

Criticality ratings for patches are usually provided by vendors, although alternate ratings may be provided by security companies. An example of such a scale is Microsoft's Severity Rating System[7]:

- Critical – A vulnerability whose exploitation could allow the propagation of an Internet worm without user action.
- Important – A vulnerability whose exploitation could result in compromise of the confidentiality, integrity, or availability of users' data, or of the integrity or availability of processing resources.
- Moderate – Exploitability is mitigated to a significant degree by factors such as default configuration, auditing, or difficulty of exploitation.
- Low – A vulnerability whose exploitation is extremely difficult, or whose impact is minimal.

### Technology Value (CTV, ITV, ATV)

Technology values will be rated by adopting the Common Vulnerability Scoring System (v2) section 2.3.3 Security Requirements Scoring Evaluation ratings. These Technology Value scores can be used independently as well as used for the complete scoring of a vulnerability that affected the technology. Each technology is assigned one of three possible values, "Low", "Medium", "High" (or Not Defined) depending on the impact from loss of confidentiality (CTV), integrity (ITV), or availability (ATV). These ratings are reproduced here:

[7] http://www.microsoft.com/technet/security/bulletin/rating.mspx

- Low (L) – Loss of [confidentiality | integrity | availability] is likely to have only a limited adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).
- Medium (M) – Loss of [confidentiality | integrity | availability] is likely to have a serious adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).
- High (H) – Loss of [confidentiality | integrity | availability] is likely to have a catastrophic adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).
- Not Defined (ND) – Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric.

As described in CVSS v2, these values should be based on network location, business function, and the potential for loss of revenue of life, although no specific methodology is defined to assign these values.

### Sources

The primary data source for patch deployments, systems under management, and time to patch can be found in automated patch management systems and processes. The primary source for data about those systems not under management can be derived from asset management systems or network discovery activities. Generally, a list of all assets under management can be extracted from patch management systems and compared to lists of all assets generated from asset management systems and/or network discovery scans.

### Dimensions

These metrics may include additional dimension for grouping and aggregation purposes. These dimensions should be applied or tagged at the level of the technology record as described in *Patch Management Metrics: Data Attributes*. For example:

- **Technology Value** dimension allows Coverage to be computed for high, medium, or lower value technologies.
- **Patch Criticality** could be a dimension if data with sufficient granularity is available.
- **Business Units** for identifying the coverage by parts of the organization.
- **Asset Value** dimension allows Coverage to be computed for high, medium, or lower value assets.
- **Tags** for characterizing types of assets, such as coverage by vendor, etc.

### Automation

The ability to automate source data collection for this metric is **high** because most automated patch management systems can provide the necessary reports in combination with assets tracking and discovery across networks providing counts of all technologies. Calculation of this metric is an ongoing basis. Once source data has been obtained, it lends itself to a **high** degree of automation.

### Visualization

These metrics may be visually represented in several ways:

**Simple visualizations** may include a table showing the metric result for the organization with each row displaying the value as of selected time periods (each week or each month).

**Graphical visualizations** may include time-series charts where the metric result is plotted on the vertical axis and time periods displayed on the horizontal axis. To provide maximum insight, plotted values for each period may include stacked series for the differing severity values.

**Complex visualizations** should be used for displaying the metric result for cross-sections of dimensions to expose concentrations of risk, such as patch criticality, business units, or technology value. For example, small multiples could be used to compare the number of high severity vulnerabilities across business units or technology values.

# Defined Metrics

## Patch Policy Compliance

### *Objective*

Patch Policy Compliance (PPC) indicates the scope of the organization's patch level for supported technologies as compared to their documented patch policy. While specific patch policies may vary within and across organizations, performance versus stated patch state objectives can be compared as a percentage of compliant systems.

Table 17: Patch Policy Compliance

| | |
|---|---|
| **Metric Name** | Patch Policy Compliance |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | Patch Policy Compliance (PPC) measures an organization's patch level for supported technologies as compared to their documented patch policy.<br>"Policy" refers to the patching policy of the organization, more specifically, which patches are required for what type of computer systems at any given time. This policy might be as simple as "install the latest patches from system vendors" or may be more complex to account for the criticality of the patch or system.<br>"Patched to policy" reflects an organization's risk/reward decisions regarding patch management. It is not meant to imply that all vendor patches are immediately installed when they are distributed. |
| **Audience** | Management, Operations |
| **Question** | What percentage of the organization's technologies is not in compliance with current patch policy? |
| **Answer** | A positive integer value between zero and 100 inclusive. A value of "100%" indicates that all technologies are in compliance to the patch policy. |
| **Formula** | Patch Policy Compliance (PPC) is calculated by dividing the sum of the technologies currently compliant by the sum of all technologies under patch management (where the current patch state is known). This metric can be calculated for subsets of technologies such as by technology value or business unit:<br>$PPC = \frac{Count(Compliant\_Instances)}{Count(Technology\_Instances)} * 100$ |
| **Units** | Percentage of technology instances |
| **Frequency** | Weekly, Monthly, Quarterly, Annually |

| | |
|---|---|
| **Targets** | PPC values should trend higher over time. An ideal result would be 100% of technologies. The expected trend for this metric over time is to remain stable or increase towards 100%. There will be variations when new patches are released for large number of technologies (such as a common operating system) that could cause this value to vary significantly. Measurement of this metric should take such events into consideration. Higher values would generally result in less exposure to known security issues. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Patch Policy Compliance exists. |

### *Usage*

Patch Management Coverage is a type of patch management metric and relies on the common definition of "patch" as defined in *Glossary*.

Patch Policy Compliance can be calculated over time typically per-week or per-month. To gain insight into the relative risk to one business unit over another, Compliance may also be calculated for cross-sections of the organization, such as individual business units or geographies or technology values and types.

### *Limitations*

This metric is highly dependent upon the current set of patch policy requirements. When patches are released that affect large numbers of technologies (such as common operating systems), this number can vary greatly with time if the lack of new patches makes a system non-compliant.

### *References*

Mell, Bergeron and Henning. Special Publication 800-40: Creating a Patch and Vulnerability Management Program. US National Institute of Standards and Technology, 2005.

## Patch Management Coverage

### *Objective*

Patch Management Coverage (PMC) characterizes the efficiency of the patch management process by measuring the percentage of total technologies that are managed in a regular or automated patch management process. This metric also serves as an indicator of the ease with which security-related changes can be pushed into the organization's environment when needed.

Table 18: Patch Management Compliance

| | |
|---|---|
| **Metric Name** | Patch Management Coverage |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | Patch Management Coverage (PMC) measures the relative amount of an organization's systems that are managed under a patch management process such as an automated patch management system. Since patching is a regular and recurring process in an organization, the higher the percentage of technologies managed under such a system the timelier and more effectively patches are deployed to reduce the number and duration of exposed vulnerabilities. |
| **Audience** | Management, Operations |
| **Question** | What percentage of the organization's technology instances are not part of the patching process and represent potential residual risks for vulnerabilities? |
| **Answer** | A positive integer value that is greater than or equal to zero. A value of "100%" indicates that all technologies are under management. |
| **Formula** | Patch Management Coverage is calculated by dividing the number of the technology instances under patch management by the total number of all technology instances within the organization. This metric can be calculated for subsets of technologies such as by asset criticality or business unit.<br>$$PMC = \frac{Count(Technology\_Instances\_Under\_Patch\_Management)}{Count(Technology\_Instances)} * 100$$ |
| **Units** | Percentage of technology instances |
| **Frequency** | Weekly, Monthly, Quarterly, Annually |
| **Targets** | PMC values should trend higher over time. Given the difficulties in manually managing systems at scale, having technologies under patch management systems is preferred. An ideal result would be 100% of technologies. However, given incompatibilities across technologies and systems this is unlikely to be attainable. Higher values would generally result in more efficient use of security resources. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for PMC exists. |

### *Usage*

Patch Management Coverage is a type of patch management metric and relies on the common definition of "patch" as defined in *Glossary*.

Optimal conditions would reflect a high value in the metric. A value of 100% would indicate that every technology in the environment falls under the patch management system. The lower the value, the greater the degree of "ad-hoc" and manual patch deployment and the longer and less effective it will be. Given that many known vulnerabilities result from missing patches, there may be a direct correlation between a higher level of Patch Management coverage and the number of known vulnerabilities in an environment. Patch Management Coverage can be calculated over time, typically per-week or per-month. To gain insight into the relative performance and risk to one business unit over another, Coverage may also be calculated for cross-sections of the organization, such as individual business units or geographies.

### *Limitations*

Not all technologies within an organization may be capable of being under a patch management system, for technical or performance reasons, so the results and interpretation of this metric will depend on the specifics of an organizations infrastructure.

### *References*

Mell, Bergeron and Henning. Special Publication 800-40: Creating a Patch and Vulnerability Management Program. US National Institute of Standards and Technology, 2005.

## Mean Time to Patch

### *Objective*

Mean Time to Patch (MTTP) characterizes the effectiveness of the patch management process by measuring the average time taken from date of patch release to installation in the organization for patches deployed during the metric time period. This metric serves as an indicator of the organization's overall level of exposure to vulnerabilities by measuring the time the organization takes to address systems known to be in vulnerable states that can be remediated by security patches. This is a partial indicator as vulnerabilities may have no patches available or occur for other reasons such as system configurations.

Table 19: Mean Time to Patch

| | |
|---|---|
| **Metric Name** | Mean Time to Patch |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | Mean Time to Patch (MTTP) measures the average time taken to deploy a patch to the organization's technologies. The more quickly patches can be deployed, the lower the mean time to patch and the less time the organization spends with systems in a state known to be vulnerable. |
| **Audience** | Operations |
| **Question** | How long does it take the organization to deploy patches into the environment? |
| **Answer** | A positive floating-point value that is greater than or equal to zero. A value of "0" indicates that patches were theoretically instantaneously deployed. |
| **Formula** | Mean Time to Patch is calculated by determining the number of hours between the Date of Availability and the Date of Installation for each patch completed in the current scope, for example by time period, criticality or business unit. These results are then averaged across the number of completed patches in the current scope:<br>$$MTTP = \frac{\sum(Date\_of\_Installation - Date\_of\_Availability)}{Count(Completed\_Patches)}$$ |
| **Units** | Hours per patch |
| **Frequency** | Weekly, Monthly, Quarterly, Annually |
| **Targets** | MTTP values should trend lower over time. Most organizations put patches through test and approval cycles prior to deployment. Generally, the target time for MTTP will be a function of the criticality of the patch and business criticality of the technology. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Mean Time to Patch exists. |

### Usage

Mean Time to Patch is a type of patch management metric, and relies on the common definition of "patch" as defined in *Glossary*.

Given that many known vulnerabilities result from missing patches, there may be a direct correlation between lower MTTP and lower levels of Security Incidents. MTTP can be calculated over time, typically per-week or per-month. To gain insight into the relative performance and risk to one business unit over another, MTTP may also be calculated for different patch criticalities and cross-sections of the organization, such as individual business units or geographies.

### Limitations

Critical Technologies. This metric assumes that the critical technologies are known and recorded. If the critical technologies are unknown, this metric cannot be accurately measured. As new technologies are added their criticality needs to be determined and, if appropriate, included in this metric.

Vendor Reliance. This metric is reliant upon the vendor's ability to notify organization of updates and vulnerabilities that need patching. If the vendor does not provide a program for notifying their customers then the technology, if critical, will always be a black mark on this metric.

Criticality Ranking. This metric is highly dependent upon the ranking of critical technologies by the organization. If this ranking is abused then the metric will become unreliable.

Patches in-Progress. This metric calculation does not account for patch installations that are incomplete or on-going during the time period measured. It is not clear how this will bias the results, although potentially an extended patch deployment will not appear in the results for some time.

### References

Mell, Bergeron and Henning. Special Publication 800-40: Creating a Patch and Vulnerability Management Program. US National Institute of Standards and Technology, 2005.

# Configuration Change Management Metrics

This section describes metrics for measuring security around the configuration and change management in an organization's environment.

Configuration management is important to organizations for both the deployment and ongoing management of systems. It encompasses the creation of initial configurations of systems and the continuous management of changes to these configurations.

The initial set of metrics for configuration management will cover changes to existing systems and configurations. Other specific processes may be covered by other sets of security metrics, such as security patch management or vulnerability management.

Changes are likely to be constantly occurring in large and complex environments. Managers will want to know how these changes impact the security of their systems and need metrics that answer questions such as:

- How much change is happening?
- How frequently are we making changes?
- How quickly can changes be implemented?
- Do we know the security impacts of these changes?
- Are we deviating from existing security policies?

The following initial set of metrics for Configuration Management are designed to provide managers with information the organization's ability to implement change, to understand the security impacts of those changes, and how these changes affect their overall risk profile.

1. **Mean time to Complete Change**. The average time taken to complete change requests.
2. **Percent of Security Reviews**. The percentage of completed change requests that had a review of the security impacts.
3. **Percentage of Security Exceptions**. The percentage of completed changes that did received an exception to current security policy.

## Data Attributes

The following is a list of attributes that should be populated as completely as possible for each configuration data record. These attributes were derived from the *ITIL v3 –Request for Change* data record.[8] Please note that some fields in the Request for Change record are documented here because they are not needed for configuration metrics calculations.

[8] S. Kempter and A. Kempter, ITIL V3 Checklist Request for Change RFC, 2008. <http://wiki.en.it-processmaps.com/index.php/Checklist_Request_for_Change_RFC>

## Configuration Change Request

Configuration Request contains information regarding the approval of configuration change requests.

Table 20: Configuration Change Request Table

### Configuration Change Request Table

| Name | Type | De-identified | Required | Description |
|---|---|---|---|---|
| Change Request ID | Number | No | Yes | Unique identifier for the change request. Generally auto-generated. |
| Submission Date | Date/Time | No | No | Date and time the change item was submitted |
| Change Owner | Text | Yes | No | Unique identifier of the person that owns the change. |
| Initiator | Text | Yes | No | Unique identifier of the person that submitted the change |
| Priority | Text | No | Recommended | How soon the request should take place. Uses values *High*, *Medium*, and *Low*. |
| Cost | Text | No | No | Estimated cost of the change in Level of Effort or actual dollar amounts |
| Approved | Boolean | No | Yes | Whether or not request was approved. Uses values *Yes* or *No*. |
| Approval Date | Date/Time | No | No | Date and time the request was approved or disapproved |
| Approved By | Text | Yes | No | Unique identifier of the person who approved the change |
| Security Reviewed | Boolean | No | Yes | Flag indicating if a security review of the change was performed. Uses values *Yes* or *No*. |
| Technology ID | Text/Number | No | No | One-to-many reference to technologies that should undergo configuration change. |
| Exempt Technology ID | Text/Number | No | No | One-to-many references to technologies that are exempt from undergoing configuration change. |

## Configuration Change Item

This table displays configuration changes that occurred on technologies within organizations.

Table 21: Configuration Change Item Table

**Configuration Change Item Table**

| Name | Type | De-identified | Required | Description |
|---|---|---|---|---|
| Configuration Change ID | Number | No | Yes | Unique identifier for the configuration change. Generally auto-generated. |
| Change Request ID | Number | No | Yes | Unique identifier for the change request. Generally auto-generated. |
| Changed By | Text | Yes | No | Unique identifier of the individual that performed the configuration change. |
| Cost | Text | No | No | Actual cost of the change in Level of Effort or actual dollar amounts |
| Technology Reference | Text/Number | No | No | One-to-many reference to the technologies that underwent configuration change. |
| Scheduled Date | Date/Time | No | No | Suggested date and time for the change |
| Completion Date | Date/Time | No | Yes | Date and time the change was completed. |
| Configuration Change Cost | Numeric | No | No | Cost of the configuration change (USD) |
| Configuration Change Effort | Numeric | No | No | Total person-hours of effort of the configuration change. |

## Classifications

Tagging of information is a valuable way to provide context to collected data records. Classification tags provide a way to group change requests, requesting parties, affected business applications or technologies, implementation teams, and change approval and review methods.

Within an organization, the combination of dimensions can provide key insight into concentrations of risks for an organization such as urgent requests on critical applications or changes to critical applications without security review.

## Sources

The primary data source for these metrics is a configuration management system or a change-control tracking system.

## Dimensions

This metric may include additional dimensions for grouping and aggregation purposes. These dimensions should be applied or tagged at the level of the underlying change record as described in *Configuration Management Metrics: Data Attributes*. For example:

- Priority of the change request
- Group requesting the change

- Whether or not security reviews were involved
- Location or business unit of the changed technology
- Results of the security review
- Importance of the technology to the organization requiring the change request

### Automation

The ability to automate the source data collection for these metrics is **medium** because most organizations maintain a tracking system for configuration changes, although these systems may vary in their degree of automation. Once the initial dataset has been collected, use of the dataset can be automated for metric calculation purposes.

### Visualization

Configuration change metrics may be visually represented in several ways:

- Simple visualizations may include a table showing metric results for the organization with each row displaying the value as of selected time periods (each week or each month). Columns may be used for different request priority levels (e.g. Low, Medium, High).
- Graphical visualizations may include time-series charts where metric results are plotted on the vertical axis and the time periods displayed on the horizontal. To provide maximum insight, plotted values for each period may include stacked series for the differing request priorities.
- Complex visualizations should be used for displaying metric results for cross-sections such as by organization or request priority. For example, small multiples could be used to compare the number of urgent change requests across business units or values of the target technologies or applications.

# Defined Metrics

## Mean Time to Complete Changes

### *Objective*

The goal of this metric is to provide managers with information on the average time it takes for a configuration change request to be completed.

Table 22: Mean Time to Complete Changes

| | |
|---|---|
| **Metric Name** | Mean Time to Complete Changes |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | The average time it takes to complete a configuration change request. |
| **Audience** | Operations |
| **Question** | What is the mean time to complete a change request? |
| **Answer** | A positive integer value that is greater than zero. A value of "0" indicates that the organization immediately implements changes. |
| **Formula** | The mean time to complete a change request is calculated by taking the difference between the date the request was submitted and the date the change was completed for each change completed within the time period of the metric. This number is then divided by the total number of changes completed during the metric's time period:<br>$MTCC = \frac{Sum(Completion\_Date - Submission\_Date)}{Count(Completed\_Changes)}$ |
| **Units** | Days per configuration change request |
| **Frequency** | Weekly, Monthly, Quarterly, Annually |
| **Targets** | MTCC values should generally trend lower over time provided operational system uptime is very high. This number will depend on the organization's business, structure, and use of IT. While a lower value indicates greater effectiveness at managing the IT environment, this should be examined in combination with the use of approval and change review controls. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Mean Time to Complete Changes exists. |

### *Usage*

Managers can use this metric to understand their ability to react to changing needs in their environment. The faster the approval cycle, the shorter the response time will be. The exact value that reflects a healthy environment will be subjective for the type of company. However, values should be similar for companies of the same size and business focus.

By focusing on high-value applications or urgent change requests they can improve their understanding of risk management capabilities. It is useful to pair this metric with data on the absolute number of changes in order to understand the effectiveness of the change management capabilities of the organization.

### *Limitations*

Only completed changes. This metric only calculates the result for changes that have been completed during the time period. Changes that have not occurred will not influence the metric results until they are completed, perhaps several reporting periods later. This may over-report performance while the changes are not completed and under-report performance after the changes has been completed.

Scheduled changes. Changes that have been submitted with a scheduled change date may result in metric values that do not provide material information. The time taken for the change request to be approved and any delays due to the work queue volumes should be considered, but not time a change request is not being processed in some manner.

Variations in the scale of changes. All changes are weighted equally for this metric regardless of the level of effort required or priority of the request and are not taken into account by the current metric definition. Organizations wanting increased precision could group results by categories of change size (e.g. Large, Medium, Small) or normalize based on level of effort.

### *References*

S. Kempter and A. Kempter, ITIL V3 Checklist Request for Change RFC, 2008. <http://wiki.en.it-processmaps.com/index.php/Checklist_Request_for_Change_RFC>

S. Kempter and A. Kempter, ITIL V3 Configuration Management Process, 2008. <http://wiki.en.it-processmaps.com/index.php/Change_Management>

A. Riley *et al*. Open Guide ITIL Configuration Management, 2008. <http://www.itlibrary.org/index.php?page=Configuration_Management>

## Percent of Changes with Security Review

### *Objective*

The goal of this metric is to provide managers with information about the amount of changes and system churn in their environment that have unknown impact on their security state.

Table 23: Percent of Change with Security Review

| | |
|---|---|
| **Metric Name** | Percent of Changes with Security Review |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | This metric indicates the percentage of configuration or system changes that were reviewed for security impacts before the change was implemented. |
| **Audience** | Management, Operations |
| **Question** | What percentage of changes received security reviews? |
| **Answer** | A positive integer value between zero and one hundred that represents a percentage. A value of "100%" indicates that all changes received security reviews during the metric time period. |
| **Formula** | The Percent of Changes with Security Review (PCSR) metric is calculated by counting the number of completed configuration changes that had a security review during the metric time period divided by the total number of configuration changes completed during the metric time period.<br>$PCSR = \frac{Count(Completed\_Changes\_with\_Security\_Reviews)}{Count(Completed\_Changes)} * 100$ |
| **Units** | Percentage of configuration changes |
| **Frequency** | Weekly, Monthly, Quarterly, Annually |
| **Targets** | PCSR values should trend higher over time. Generally speaking, change management processes should contain review and approval steps that identify potential business and security risks. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Percent of Changes with Security Review exists. |

### *Usage*

Managers can use this metric to understand the degree to which changes with unknown security impacts are occurring in their environment. The metric results indicate the amount of churn that has a known impact on the intended security model of the organization. As changes with unknown security implications accumulate, it would be expected that the security model of these systems would degrade.

By focusing on changes to high-value applications and technologies or key business units, managers can understand the degree to which security risks may be introduced to these systems.

### *Limitations*

Only completed changes. This metric is only calculating the results for changes that have been completed during the time period. Changes in security review policies may not be included in this metric if the changes have not been completed in the metric time period.

Variations in the scale of changes. All changes are weighted equally for this metric regardless of the level of effort required or priority of the request and are not taken into account by the current metric definition. Organizations wanting increased precision could group results by categories of change size (e.g. Large, Medium, Small) or normalize based on level of effort.

### *References*

S. Kempter and A. Kempter, ITIL V3 Checklist Request for Change RFC, 2008. <http://wiki.en.it-processmaps.com/index.php/Checklist_Request_for_Change_RFC>

S. Kempter and A. Kempter, ITIL V3 Configuration Management Process, 2008. <http://wiki.en.it-processmaps.com/index.php/Change_Management>

A. Riley *et al*. Open Guide ITIL Configuration Management, 2008. <http://www.itlibrary.org/index.php?page=Configuration_Management>

## Percent of Changes with Security Exceptions

### *Objective*

The goal of this metric is to provide managers with information about the potential risks to their environment resulting from configuration or system changes exempt from the organization's security policy.

Table 24: Percent of Changes with Security Exceptions

| | |
|---|---|
| **Metric Name** | Percent of Changes with Security Exceptions |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | This metric indicates the percentage of configuration or system changes that received an exception to existing security policy. |
| **Audience** | Operations |
| **Question** | What percentage of changes received security exceptions? |
| **Answer** | A positive integer value between zero and one, reported as a percentage. A value of "100%" indicates that all changes are exceptions. |
| **Formula** | This Percentage of Security Exception (PCSE) metrics are calculated by counting the number of completed configuration changes that received security exceptions during the metric time period divided by the total number of configuration changes completed during the metric time period:<br>$$PCSE = \frac{Count(Completed\_Changes\_with\_Security\_Exceptions)}{Count(Completed\_Changes)} * 100$$ |
| **Units** | Percentage of configuration changes |
| **Frequency** | Weekly, Monthly, Quarterly, Annually. |
| **Targets** | PCSE values should trend lower over time. Generally speaking, exceptions made to security policies increase the complexity and difficulty of managing the security of the organization. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Percent of Changes with Security Exceptions exists. |

### *Usage*

Manager can use this metric to understand their exposure in terms of the percentage of change exceptions to their security policy. While exceptions may be granted based on negligible risk or additional controls, it is possible that accumulated change exceptions could degrade their security posture.

By focusing on exceptions granted to changes to high-value applications and technologies, or key business units, managers can focus their attention and resources and increase their understanding of the degree to which security risks may be introduced to these systems.

### *Limitations*

Only completed changes. This metric is only calculating the results for changes that have been completed during the time period. Changes in-progress will not be included in this metric if they have not been completed in the metric time period.

Variations in the scale of changes. All changes are weighted equally for this metric and do not take into account the amount of effort required. For a better understanding of the scale of exceptions, organizations should group results by categories of change size (Large, Medium, Small) or normalize based on scale of the change.

Dependency on security reviews. Security exceptions may only have been granted for systems that received security reviews. Changes implemented without security reviews may include unknown and untracked exceptions to security policies.

### *References*

S. Kempter and A. Kempter, ITIL V3 Checklist Request for Change RFC, 2008. <http://wiki.en.it-processmaps.com/index.php/Checklist_Request_for_Change_RFC>

S. Kempter and A. Kempter, ITIL V3 Configuration Management Process, 2008. <http://wiki.en.it-processmaps.com/index.php/Change_Management>

A. Riley *et al*. Open Guide ITIL Configuration Management, 2008. <http://www.itlibrary.org/index.php?page=Configuration_Management>

# Application Security Metrics

This section describes metrics for measuring security around the business applications in an organization's environment.

Business applications perform many functions from order processing to inventory management. Organizations are increasingly dependent on business applications, especially applications connected to the Internet for transactions between customers, suppliers, business units and employees.

While a individual applications may be more or less critical than another, all managers want to understand if they can rely on their business applications to reliably function as intended. Security issues with business applications can put both information assets as well as the capability to operate at risk.

The security of these business applications depends on several factors:

- Design of the underlying security model
- Selection and incorporation of component technologies
- Development of the applications, through software development and integration processes
- Underlying infrastructure such as the operating systems and applications

The following initial set of metrics for Application Security are designed to provide managers with information on the distribution by types of applications they are managing, what the known risks to those applications are, and how well their applications have been examined for weaknesses:

1. **Number of Applications**. The absolute number of applications provides a useful measure that allows an organization to understand "what they have" and to interpret the results provided by other metrics. As a key indicator of risk, the number of critical and high value applications should be viewed.
2. **Percentage of Critical Applications**. This metric identifies the percentage of an organization's applications that are critical to its operations. This helps the organization understand their relative level of exposure to application security risks.
3. **Risk Assessment Coverage**. This metric examines the percentage of applications that have undergone a risk assessment. Understanding the percentage of applications that have had a risk assessment performed provides managers with a better understanding of their risks among their applications. A key risk indicator is the Risk Assessment Coverage for High Value applications.
4. **Security Testing Coverage**. The percentage of post-deployment applications that have experienced material security testing for weaknesses is a key indicator of the level of application security risk.

## Data Attributes

The following is a list of attributes that should be populated as completely as possible for each application security data record.

### Technologies

The following is a list of attributes that should be populated as completely as possible for each technology

Table 25: Technologies Table

Technologies Table

| Name | Type | De-identified | Required | Description |
|---|---|---|---|---|
| Technology ID | Number | No | Yes | Unique identifier for the technology. Generally auto-generated. |
| Name | Text | No | No | Fields from CPE Dictionary. |
| Vendor | Text | No | No | Fields from CPE Dictionary. |
| Version | Text | No | No | Fields from CPE Dictionary. |
| Technology Value | Text | No | Recommended | Impact from the loss of this technology (C/I/A) to the organization. Uses value Low, Medium, High, and Not Defined. [9] |
| Under Patch Management | Boolean | No | Yes | This is a flag to use with Patch Management metrics. |
| Dimensions/Tags | Text/Drop-Down List | No | No | Business Unit, Technology class, geographical area |

### Business Applications

This table contains information regarding an organization's business applications.

Table 26: Business Applications Table

Business Applications Table

| Name | Type | De-identified | Required | Description |
|---|---|---|---|---|
| Application ID | Number | No | Yes | Unique identifier for the application. Generally auto-generated. |
| Business Application Name | Text | Yes | No | The name of the business application to which this technology belongs (if any). |

[9] This is adopting 2.3.3 Security Requirements Scoring Evaluation from CVSS v2, http://www.first.org/cvss/cvss-guide.html#i2.3.

| | | | | |
|---|---|---|---|---|
| Business Application Value | Text | No | Recommended | A value that indicates the impact from the loss of this business system to the organization. Use values Low, Medium, High, and Not Defined. |
| Business Unit | Text | Yes | No | Fields indicating the applications business unit and owner. |
| Application Status | Text | No | Yes | Indicator of the application's current status. Uses values: In Testing, In Development, and Production. |
| Date of Last Status Change | Date / Time | No | No | Date and time when application status was last changed. |
| Date of Deployment | Date / Time | No | No | Date and time when application was deployed. Additional dates (for development, testing, and deployment milestones can also be included). |
| Technology Reference | Text | No | No | References to the components of this applications found in the technologies table |
| Dimensions/Tags | Text/Drop-Down List | No | No | Business Unit, Applications Scope, Technology class, geographical area, commercial vs. custom software, managed in-house or externally hosted |

### Risk Assessments

This table contains information on the risk assessments performed in the organization. Currently for the initial set of metrics, relatively few fields are required. Organizations can include additional fields to enhance their ability to measure and understand their risks.

Table 27: Risk Assessments Table

Risk Assessments Table

| Name | Type | De-identified | Required | Description |
|---|---|---|---|---|
| Assessment ID | Number | No | Yes | Unique identifier for the assessment. Generally auto-generated. |
| Date of Assessment | Date / Time | No | No | Date that risk assessment was completed. |
| Application ID | Number | Yes | Yes | Reference identifier for the application. |
| Assessment Type | Text | No | No | Methodology or process used for the Risk Assessment, such as: FAIR, FRAP, OCTAVE, SOMAP, ISO 27005, NIST 800-30 |
| Assessment Effort | Numeric | No | No | Total person-hours of the assessment |

| | | | | |
|---|---|---|---|---|
| Assessment Cost | Numeric | No | No | Total cost of the assessment |
| Assessment Scope | Text | No | No | Scope of the risk assessment covering this application: *Organization*, *system*, or *application* |
| Assessment Results | Text | No | No | Results of the assessment. |

## Security Testing

This table contains information about security tests, such as manual penetration tests, static or dynamic binary analysis, and other application security testing. Organizations can include additional fields to enhance their ability to measure and understand their risks.

**Table 28: Security Testing Table**

### Security Testing Table

| Column Name | Type | De-identified | Required | Column Description |
|---|---|---|---|---|
| Security Test ID | Number | No | Yes | Unique identifier for the test. Generally auto-generated. |
| Date of Testing | Date / Time | No | No | Date that security testing was performed or completed. |
| Application ID | Number | Yes | Yes | Reference identifier for the application. |
| Test Type | Text | No | No | Methodology or process used for the security testing such as: *Source Code Analysis, Static Binary Analysis, Dynamic Analysis, Fuzzing, Penetration Testing* |
| Test Method | Text | No | No | *Manual* or *Automated* |
| Test Results | Text | No | No | Results of the testing. |
| Security Test Effort | Numeric | No | No | Total person-hours of test effort |
| Security Test Cost | Numeric | No | No | Cost of the Security Test (USD). |

## Classifications

Tagging of information is a very valuable way to provide context to collected data records. Classification tags provide a way to group change requests, requesting parties, affected business applications or technologies, implementation teams, and change approval and review methods.

It is expected that dimensions will be added to these tables to provide the ability to view metric results that address key questions and concerns. Examples of dimensions that can be added to the metric datasets include:

- **Technologies**: Application status, business unit, geography, business value, or technology category by technology
- **Risk Assessments**: Assessment method or development stage
- **Security Testing**: Testing effort, testing team, or test duration

Within an organization, the combination of dimensions can provide key insight into concentrations of risks for an organization such as the percent of critical applications without risk assessments or security testing.

### Business Application Value

Business Applications will be rated for their value by adopting a simplified version of the Common Vulnerability Scoring System (v2) section 2.3.3 Security Requirements Scoring Evaluation ratings. Each Business Applications is assigned one of three possible values, “Low”, “Medium”, “High” (or Not Defined) depending on the impact from loss of this system to the business. These ratings are reproduced here:

- Low (L). Loss of [confidentiality | integrity | availability] is likely to have only a limited adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).
- Medium (M). Loss of [confidentiality | integrity | availability] is likely to have a serious adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).
- High (H). Loss of [confidentiality | integrity | availability] is likely to have a catastrophic adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).
- Not Defined (ND). Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric.

As described in CVSS v2, these values should be based on network location, business function, and the potential for loss of revenue or life, although no specific methodology is defined to assign these values.

## Sources

The data sources for these metric are application tracking systems that containing application and values, risk assessment tracking systems that contain the dates and results of assessments, and security testing histories.

## Dimensions

This metric may include additional dimensions for grouping and aggregation purposes. These dimensions should be applied or tagged at the level of the underlying application record as described in Application Security Metrics: Data Attributes. For example:

- Value of applications allows for analysis of the volume of applications that are of high, medium, or low value to the organization
- Location or business unit in the organization allows for the identification of concentrations of risk

- Assessment types and scope
- Development stage of the application
- Testing type, such as manual penetration, automated testing, binary analysis
- Testing organizations (e.g. in-house or external consultants)

### Automation

The ability to automate the source data collection for this metric is **medium**. While most organizations maintain tracking systems for business applications, risk assessments and security testing, these systems are generally maintained manually. Once the initial dataset has been collected, the potential for ongoing automation is **high**.

### Visualization

Application security metrics may be visually represented in several ways:

- Simple visualizations may include a table showing the number of applications for the organization with each row displaying the value for selected time periods (each week or each month). Columns may be used for different application value levels (e.g. Low, Medium, High).
- Graphical visualizations may include time-series charts where the number of applications is plotted on the vertical axis and the time periods displayed on the horizontal. To provide maximum insight, plotted values for each period may include stacked series for the differing values of applications.
- Complex visualizations should be used for displaying the number of applications for cross-sections such as by organization or asset value. For example, small multiples could be used to compare the number of high value applications across business units.

# Defined Metrics

## Number of Applications

### *Objective*

The goal of this metric is to provide managers with the number of applications in the organization and to help translate the results of other metrics to the scale of the organization's environment.

Table 29: Number of Applications

| | |
|---|---|
| **Metric Name** | Number of Applications |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | This metric counts the number of applications in the organization's environment. |
| **Audience** | Operations |
| **Question** | What is the number of applications in the organization? |
| **Answer** | A positive integer value that is greater than or equal to zero. A value of "0" indicates that the organization does not have any applications. |
| **Formula** | The number of applications (NOA) is determined by simply counting the number of applications in the organization:<br>$NOA = Count(Applications)$ |
| **Units** | Number of applications |
| **Frequency** | Weekly, Monthly, Quarterly, Annually. |
| **Targets** | NOA values generally should trend lower over time although this number will depend on the organization's business, structure, acquisitions, growth and use of IT. This number will also help organizations interpret the results of other applications security metrics. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Number of Applications exists. |

### *Usage*

Managers can use this metric to understand and monitor changes to their application environment. This metric provides a reference point for metrics around the organization's applications.

### *Limitations*

Variations in application scope. Different organizations might count as a "single" application a system that another organization may consider several distinct applications, resulting in significantly different numbers of applications between organizations.

Variations in application scale. Applications within or across organizations might be significantly different in size, so the level of effort required to assess, test or fix vulnerabilities may vary between applications.

### *References*

Web Application Security Consortium. Web Application Security Statistics Project., http://www.webappsec.org/projects/statistics/

## Percentage of Critical Applications

### *Objective*

This metric tracks the percentage of applications that are critical to the business.

Table 30: Percentage of Critical Applications

| | |
|---|---|
| **Metric Name** | Percentage of Critical Applications |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | The percentage of critical applications measures the percent of applications that are critical to the organization's business processes as defined by the application's value rating. |
| **Audience** | Operations |
| **Question** | What percentage of the organization's applications is of critical value? |
| **Answer** | Positive integer value that is equal to or greater than zero and less than or equal to one hundred, reported as a percentage. A value of "100%" indicates that all applications are critical. |
| **Formula** | The Percentage of Critical Applications (PCA) metric is calculated by dividing the number of applications that have high value to the organization by the total number of applications in the organization:<br>$PCA = \frac{Count(Critical\_Applications)}{Count(Applications)} * 100$ |
| **Units** | Percent of applications |
| **Frequency** | Weekly, Monthly, Quarterly, Annually. |
| **Targets** | Because of the lack of experiential data from the field, no consensus on goal values for the percentage of critical applications. The result will depend on the organization's business and use of IT. |

### *Usage*

Managers can use this metric to gain a better understanding of the quantity of applications that are critical to their organization. This metric provides a reference to the scale of the organization's use of applications and assists managers with better understanding of the scope and scale of their application security risk.

### *Limitations*

Variations in application scope. Different organizations might count as a "single" application a system that another organization may consider several distinct applications, resulting in significantly different numbers of applications between organizations.

Variations in application scale. Applications within or across organizations might be significantly different in size, so the level of effort required to assess, test or fix vulnerabilities may vary between applications.

## Risk Assessment Coverage

### *Objective*

This metric reports the percentage of applications that have been subjected to risk assessments.

Table 31: Risk Assessment Coverage

| | |
|---|---|
| **Metric Name** | Risk Assessment Coverage |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | Risk assessment coverage indicates the percentage of business applications that have been subject to a risk assessment at any time. |
| **Audience** | Management, Operations |
| **Question** | What percentage of applications have been the subjected to risk assessments? |
| **Answer** | A positive value between zero and one hundred, reported as a percentage. A value of "100%" would indicate that all applications have had risk assessments. |
| **Formula** | The metric is calculated by dividing the number of applications that have been subject to any risk assessments by the total number of applications in the organization:<br>$$RAC = \frac{Count(Applications\_Undergone\_Risk\_Assessment)}{Count(Applications)} * 100$$ |
| **Units** | Percent of applications |
| **Frequency** | Weekly, Monthly, Quarterly, Annually. |
| **Targets** | RAC values should trend higher over time. A higher result would indicate that more applications have been examined for risks. Most security process frameworks suggest or require risk assessments for applications deployed in production environments. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Risk Assessment Coverage exists. |

### *Usage*

Managers can use this metric to evaluate their risk posture in terms of applications that have undergone a risk assessment. A better understanding of the quantity of applications that have not been exposed to a risk assessment allows the organization to evaluate their level of unknown risk associated with these applications. With metric results for different dimensions is possible to identify and evaluate concentrations of risk, such as for results for critical applications or applications containing confidential information.

### Sources

The data source for this metric is a risk assessment tracking system.

### Limitations

Variations in application scope. Different organizations might count as a "single" application a system that another organization may consider several distinct applications, resulting in significantly different numbers of applications between organizations.

Variations in application scale. Applications within or across organizations might be significantly different in size, so the level of effort required to assess, test or fix vulnerabilities may vary between applications.

Depth of Risk assessments. Risk assessments can vary in depth due to the methodology used, the amount of time spent, and the quality of the assessment team.

Stage when Assessed. Risk assessments can occur at varying times in an application's development cycle that may influence the assessment.

### References

Web Application Security Consortium. Web Application Security Statistics Project.
http://www.webappsec.org/projects/statistics/

## Security Testing Coverage

### *Objective*

This metric indicates the percentage of the organization's applications have been tested for security risks.

Table 32: Security Testing Coverage

| | |
|---|---|
| **Metric Name** | Security Testing Coverage |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | This metric tracks the percentage of applications in the organization that have been subjected to security testing. Testing can consists of manual or automated white and/or black-box testing and generally is preformed on systems post-deployment (although they could be in pre-production testing).<br>Studies have shown that there is material differences in the number and type of application weaknesses found. As a result, testing coverage should be measured separately from risk assessment coverage. |
| **Audience** | Operations |
| **Question** | What percentage of applications has been subjected to security testing? |
| **Answer** | A positive value between zero and one hundred, reported as a percentage. A value of "100%" would indicate that all applications have had security testing. |
| **Formula** | This metric is calculated by dividing the number of applications that have had post-deployment security testing by the total number of deployed applications in the organization:<br>$$STC = \frac{Count(Applications\_Undergone\_Security\_Testing)}{Count(Deployed\_Applications)} * 100$$ |
| **Units** | Percent of applications |
| **Frequency** | Weekly, Monthly, Quarterly, Annually. |
| **Targets** | STC values should trend higher over time. Generally, the higher the value and the greater the testing scope, the more vulnerabilities in the organization's application set will be identified. A value of 100% indicates that every application has been subject to post-deployment testing. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Security Testing Coverage exists. |

### *Usage*

Managers can use this metric to evaluate the degree to which applications have been tested for weaknesses during the post-development phase (dimensions could be used to expand this metric to cover various stages of the development lifecycle). Quantifying the applications not subjected to security testing allows the organization to evaluate their application risk.

### *Automation*

The ability to automate source data collection for this metric is **medium**. While the results of security testing are often maintained in a tracking system, these systems are generally maintained manually. Once the initial dataset has been collected, use of the dataset can be automated for metric calculation purposes.

### *Limitations*

Variations in application scope. Different organizations might count as a "single" application a system that another organization may consider several distinct applications, resulting in significantly different numbers of applications between organizations.

Variations in application scale. Applications within or across organizations might be significantly different in size, so the level of effort required to assess, test or fix vulnerabilities may vary between applications.

Depth of Risk assessments. Risk assessments can vary in depth due to the methodology used, the amount of time spent, and the quality of the assessment team.

### *References*

Web Application Security Consortium. Web Application Security Statistics Project.
http://www.webappsec.org/projects/statistics/

## Financial Metrics

The combination of security costs and security outcome metrics can be used to understand if security spending is optimized, if projects meet their projected goals, and if organizations are focusing on the right areas. If cost data is not available, if may be possible to use effort data instead (e.g. FTEs and time.) For instance, metrics centered around the effort involved in security processes, such as the effort to remediate a vulnerability can be used to improve efficiency. Metrics around the impact and benefits to the organization, such as reductions in the number of security incidents can improve overall security effectiveness.

When organizations consider their security costs and benefits the three questions they seek to answer are:

1. How much is being spent on information security? Companies would like to know if their security spending is in-line to other organizations with similar characteristics. If they are over- or under- spending compared to their peers and their security posture seems equivalent than they know that their spending is likely to be less or more effective than their peers. An issue with comparing "financial" metrics in isolation is that there are several unobserved values, namely the effectiveness of the security that is being purchased.

2. What is the security budget being spent on? Looking at the ways in which security budgets are allocated can help optimize spending. This can help identify if the most resources are being directed at the areas of greatest risks, and if spending is aligned with the organization's strategy.

3. What are the benefits received for this spending? Directly measuring the benefits of security spending is challenging. Currently most benefits can only be captured as reduced time spent by personnel in maintaining a level of security activity, reduced numbers of common incidents (password resets, virus clean-ups), and reduced operational downtime, but can't easily measure averted threats. It is also possible to consider the benefits of particular projects and spending segments by looking at improvements in the performance of business functions, for example, and the marginal change resulting from additional spending.

**Initial Metrics:**

1. **Percent of IT budget allocated to information security**. How much of information security spending is allocated to security, normalized as a percentage of overall IT spending.

2. **Security Budget Allocation**. What things is the security budget being spent on, such as systems, personnel, software licenses, managed services, etc. Percentage of spending on: personnel, software and hardware, services (of different types), managed services, products of various type and purpose, and training.

## Data Attributes

The following is a list of attributes that should be populated as completely as possible.

Table 33: Security Spending Table

**Information Security Spending Table**

| Name | Type | De-identified | Required | Description |
|---|---|---|---|---|
| Reference ID | Number | No | No | Unique identifier for the security spending. Generally auto-generated. |
| Time Period Start Date | Date | No | Yes | The starting date for the time period for which this spending occurred |
| Time Period End Date | Date | No | Yes | The ending date for the time period for which this spending occurred |
| IT Budget | Number | Yes | Yes* | The total IT budget (including security activities) for this time period |
| IT Actual | Number | Yes | No | The actual IT spending during this time period (including security activities). |
| IT Security Budget | Number | Yes | Yes* | The total amount budgeted for information security personnel, services, and systems during the time period. |
| IT Security Actual | Number | Yes | No | The actual spending on information security personnel, services, and systems during the time period |
| Spending Category | Text/Drop-down | Yes | No | An indicator of the purpose of the security spending, from categories: Personnel, Systems, Managed Services, Services, Training, and Other. |
| Purpose | Text/Drop-down | No | No | Purpose of the spending: Prevention, Detection, Incident Response, Auditing |
| Additional Dimensions | Text | Yes | No | Additional dimensional tags such as business unit, location, etc. These additional fields could include references to technologies or applications. |

*This table could be assembled with multiple rows for each time period, with one for the IT budget, and other rows for the budget for specific security items, summing in the rows for the relevant metric time period. For simplicity, if this is done, it is recommended that all rows provide values for the same time periods as the metric calculations.

### Security Spending and Budget

The products, procedures, and personnel (employees and contractors) that are primarily dedicated to or used for provision of IT security for the specific IT investment, such as the activities covered under ISO 27002. All capital and operational costs for IT Operational Security, IT Risk Management, IT Compliance, IT Privacy, and IT Disaster Recovery should be included even through these costs may cross organizational boundaries. Dimensions can be used to maintain information on spending by organizational units.

Following guidance presented in OMB Circular No. A-11 Section 53 (2008), security spending is defined as spending on or intended for activities and systems including:

- Risk assessment;

- Security planning and policy;
- Certification and accreditation;
- Specific management, operational, and technical security controls (to include access control systems as well as telecommunications and network security);
- Authentication or cryptographic applications;
- Security education, awareness, and training;
- System reviews/evaluations (including security control testing and evaluation);
- Oversight or compliance inspections;
- Contingency planning and testing;
- Physical and environmental controls for hardware and software;
- Auditing and monitoring;
- Computer security investigations and forensics; and
- Reviews, inspections, audits and other evaluations performed on contractor facilities and operations.
- Managed services, consulting services providing any of the above;

### Spending Categories and Purpose

Security spending can be tracked in more detail by indicating the category of item the spending is for, such as Personnel (in-house), Systems (software, appliances, and hardware), Managed Services, Security Services (such as penetration testing), Training, Auditing, and Other.

The spending can be assigned a purpose, such as prevention (on controls and hardening), detection (IDS systems, log monitoring, etc.), auditing and measurement, and incident response and recovery. These dimensions can be used to gain a more complete picture of the allocation of security spending and its impact on the performance of business functions.

### Sources

Sources for financial data include published budgets and financial management systems. In some cases manual effort will be required to separate security spending from IT budgets, or to sum security spending across multiple divisions or departments.

### Dimensions

This metric may include additional dimensions for grouping and aggregation purposes. These dimensions should be tagged at the row level, an can include :

- **Business functions** to track financial metrics on security around specific business activities
- **Business Units** owing the systems to which the security spending is directed
- **Geographic locations** for analyzing spending across multiple locations

### Automation

The ability to automate source data collection for these metrics is **medium**, because most organizations use financial management systems for budgeting activities, however these results may require additional work to determine total security spending across multiple units, group. Locations and systems. Calculation of these metrics on an ongoing basis, after source data has been obtained, lends itself to a **moderate** degree of automation, as a process can be defined, but some recurring analysis is likely to be required.

### Visualization

These metrics may be visually represented in several ways:

**Simple visualizations** may include a table showing metric results for the organization with each row displaying the value for selected time periods (each week or each month). Columns may be used for spending categories (e.g. Personnel) or purposes (e.g. Prevention).

**Graphical visualizations** may include time-series charts where the metric result is plotted on the vertical axis and time periods displayed on the horizontal axis. To provide maximum insight, plotted values for each period may include stacked series for the differing categories or purposes or business units (for Information Security Budget as % of IT Budget).

**Complex visualizations** should be used for displaying the metric result for cross-sections by organization, categories, or purposes. For example, small multiples could be used to compare the spending on systems for prevention across business units.

# Defined Metrics

## Information Security Budget as % of IT Budget

### *Objective*

Organizations are seeking to understand if their security spending is reasonable for the level of security performance and in-line with other organizations. This metric presents the IT security budget as a percentage of organizations overall IT budget, tracking the relative cost of security compared to IT operations. This result can also be used to benchmark spending against other organizations.

Table 34: Security Budget as % of IT Budget

| | |
|---|---|
| **Metric Name** | Information Security Budget as a Percentage of IT Budget |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | Security budget as a percentage of IT Budget tracks the percentage of IT spending on security activities and systems. For the purposes of this metric, it is assumed that Information Security is included in the IT budget. |
| **Audience** | Management |
| **Question** | What percentage of the IT Budget is allocated to information security? |
| **Answer** | A positive value equal to or between 0 and 1, expressed as a percentage. A value of "100%" indicates that the entire Information Technology budget is dedicated to information security. |
| **Formula** | The total budget allocated for security activities and systems for the metric time period is divided by the total information security budget.<br>$SBPITB = \frac{SecurityBudget}{ITBudget}$ |
| **Units** | Percentage of IT Budget |
| **Frequency** | Quarterly, Annually depending on budget cycle |
| **Targets** | Because of the lack of experiential data from the field, no strong consensus on the range of acceptable goal values for security spending exists In general, this value should be comparable with peer organizations with similar IT profiles and security activities. |
| **Sources** | Financial management systems and/or annual budgets |

### *Usage*

Examining and tracking the percentage of the IT budget allocated to security allows an organization to compare the costs of securing their infrastructure between an organization's division, against other organizations, as well as to observe changes over time. These results will also provide a foundation for the optimization of security spending through comparison of spending with the outcomes of other metrics such as numbers of incidents, time to detection, time to patch, etc.

The percentage of budget allocated to security should be calculated over time, typically per-quarter or per-year. To gain insight into the relative performance of one business unit over another, this result may also be calculated for cross-sections of the organization, such as individual business units or geographies where they have discrete budgets.

### *Limitations*

**Different threat profiles across organizations**. While there is systemic risk to common viruses and attacks, there is also firm specific risk based on the companies specific activities that may require higher or lower level of security spending relative to peer organizations.

**Different IT profiles across organizations**. Although in theory all organizations will make market-efficient use of IT, legacy systems and specific implementations will impact the costs of otherwise-similar IT operations as well as the costs of similar levels of security performance.

**Differences in accounting**. Different organizations may account for both IT and security spending in different ways that make it hard to compare this value across organizations. Some may leverage IT resources for security purposes that make it hard to account for such partial FTEs without significant activity-based costing exercises, others may have lump-sum outsourced IT contracts without specific information on security spending.

### *References*

Chew, Swanson, Stine, Bartol, Brown and Robinson. Special Publication 800-55: Performance Measurement Guide for Information Security (Rev 1). US National Institute of Standards and Technology, 2008

Open Web Application Security Project, Security Spending Benchmark Project <https://www.owasp.org/index.php/Category:OWASP_Security_Spending_Benchmarks>

Office of Management and Budget, OMB Circular No. A–11 (2008) , Form 300s and 53s

## Information Security Budget Allocation

### *Objective*

Looking at the ways in which security budgets are allocated can help optimize spending. This can help identify is the most resources being directed at the areas of greatest risks, and if spending is aligned with the organization's strategy.

Table 35: Information Security Budget Allocation

| | |
|---|---|
| **Metric Name** | Information Security Budget Allocation |
| **Version** | 1.0.0 |
| **Status** | Final |
| **Description** | Information security budget allocation tracks the distribution of security spending across a variety of security activities, systems, and sources, as a percentage of overall information security spending. |
| **Audience** | Management |
| **Question** | What percentage of the Information Security Budget is allocated to each category of spending? |
| **Answer** | A positive value equal to or between 0 and 1, expressed as a percentage for each spending category. A value of "100%" indicates that the entire Information Security budget is dedicated to that spending category. |
| **Formula** | For each budget category, divide the amount allocated to the category by the total information security budget. These values should be for the relevant item period only. If the category of any budget costs is unknown they should be allocated to an "unknown" category. |
| **Units** | Percentage of Information Security Budget |
| **Frequency** | Quarterly, Annually depending on budget cycle |
| **Targets** | Because of the lack of experiential data from the field, no consensus on a goal value for the allocation of security spending exists. In general, this value should be comparable with peer organizations with similar security performance across each of the sending categories, and will vary depending on the use of in-house vs. external resources, software license structures, reliance on outsourcing, etc. |
| **Sources** | Financial management systems and/or annual budgets |

### *Usage*

Examining and tracking the percentage of the IT budget allocated to security allows an organization to compare the relative costs of their various information security activities. This can help identify if security spending is being directed toward the areas of greatest risk to the organization, i.e. is security spending aligned with the results of risk assessments? It also enables organizations to start to optimize spending by observing incremental changes in business function performance correlating to changes in spending on various security activities, such as numbers of incidents, time to detection, time to patch, etc.

The percentage of information security budget allocated to security should be calculated over time, typically per-quarter or per-year.

To gain insight into the relative performance of one business unit over another, this result may also be calculated for cross-sections of the organization, such as individual business units or geographies where they have discrete budgets.

### *Limitations*

**Different threat profiles across organizations**. While there is systemic risk to common viruses and attacks, there is also firm specific risk based on the companies specific activities that may require higher or lower level of security spending relative to peer organizations.

**Different IT profiles across organizations**. Although in theory all organizations will make market-efficient use of IT, legacy systems and specific implementations will impact the costs of otherwise-similar IT operations as well as the costs of similar levels of security performance.

**Differences in accounting**. Different organizations may account for both IT and security spending in different ways that make it hard to compare this value across organizations. Some may leverage IT resources for security purposes that make it hard to account for such partial FTEs without significant activity-based costing exercises, others may have lump-sum outsourced IT contracts without specific information on security spending.

### *References*

Chew, Swanson, Stine, Bartol, Brown and Robinson. Special Publication 800-55: Performance Measurement Guide for Information Security (Rev 1). US National Institute of Standards and Technology, 2008

Open Web Application Security Project, Security Spending Benchmak Project <https://www.owasp.org/index.php/Category:OWASP_Security_Spending_Benchmarks>

Office of Management and Budget, OMB Circular No. A–11 (2008) , Form 300s and 53s

# Appendix A: Glossary

## Application Security Testing

The term *application security testing* is defined as a material test of the security of a business application after it has been developed and deployed (although it may be a pre-production test). It can consist of a combination of one or more of the following techniques:

- Source code analysis (automated and/or manual)
- Manual penetration testing (white- or black-box),
- Static or dynamic binary analysis,
- Automated testing, or
- “Fuzzing” or other techniques that identify vulnerabilities in an application.

## Bias

*Bias* is identified as a term that refers to how far the average statistic lies from the parameter it is estimating, that is, the error that arises when estimating a quantity. Errors from chance will cancel each other out in the long run, those from bias will not.[10] *Systemic Bias* is identified as the inherent tendency of a process to favor a particular outcome.[11]

## Business Application

The term *business application* can mean many things in IT systems ranging from productivity applications on individual desktop computers to complex manufacturing systems existing on multiple pieces of custom hardware. In this context, the term refers to a set of technologies that form a system performing a distinct set of business operations. Examples of this include an order processing system, online shopping cart, or an inventory tracking system.

Since applications can consist of more than one technology, the scope of an application is defined as a process or set of processes that the organization manages and makes decisions around as a single entity. Generally, this scope is not intended to include infrastructure components of the application, such as the web or application server itself, although this may not be separated for certain types of testing.

## Containment

*Containment* is identified as limiting the extent of an attack.[12] Another way to look at containment is to “stop the bleeding”. The impact of the incident has been constrained and is not increasing. Measure can now be taken to recover systems, and “effective recovery” of primary capabilities may be complete.

[10] Source: Wikipedia. <http://en.wikipedia.org/wiki/Bias>

[11] Source: Wikipedia. <http://en.wikipedia.org/wiki/Systemic_bias>

[12] G. Miles, Incident Response Part #3: Containment. Security Horizon, Inc., 2001. <http://www.securityhorizon.com/whitepapersTechnical/IncidentResponsepart3.pdf>

## Data Record

A *Data record* is a single sample of data for a particular metric. Each data record roughly approximates a row in a relational database table. Data records contain *data attributes* that describe the data that should be collected to calculate the metric. Each data attribute roughly approximates a column in the database table. Attributes contains the following characteristics:

- **Name** — a short, descriptive name.
- **Type** — the data type of the attribute. Types include Boolean, Date/Time[13], Text, Numeric and ISO Country Code.
- **De-identification** — a Boolean value describing whether the field of the data record should optionally be cleansed of personally or organizationally identifying information. If "yes," then prior to consolidation or reporting to a third-party, the data in this field should be de-identified using a privacy-preserving algorithm, or deleted. For example, severity tags for security incidents might require de-identification.
- **Description** — additional information describing the attribute in detail.

In this document, the beginning of each major section describes the attributes that should be collected in order to calculate the metric.

## De-identified

De-identified information is information from which all potentially identifying information that would individually identify the provider has been removed. For the purposes of these metrics, these are data records for which de-identification needs to occur in order to maintain the anonymity of the data provider.

## Risk Assessment

The term *risk assessment* is defined as a process for analyzing a system and identifying the risks from potential threats and vulnerabilities to the information assets or capabilities of the system. Although many methodologies can be used, it should consider threats to the target systems, potential vulnerabilities of the systems, and impact of system exploitation. It may or may not include risk mitigation strategies and countermeasures. Methodologies could include FAIR, OCTAVE or others.

## Security Incident

A *security incident* results in the actual outcomes of a business process deviating from the expected outcomes for confidentiality, integrity & availability due to deficiencies or failures of people, process or technology.[14] Incidents that should not be considered "security incidents" include disruption of service due to equipment failures.

## Security Patch

A *patch* is a modification to existing software in order to improve functionality, fix bugs, or address security vulnerabilities. *Security patches* are patches that are solely or in part created and released to address one or more security flaws, such as, but not limited to publicly disclosed vulnerabilities.

[13] Also known as a "timestamp."

[14] Source: Operational Risk Exchange. <http://www.orx.org/reporting/>

### Technology

A *technology* is an application, operating system, or appliance that supports business processes. A *critical technology* is one upon which normal business operations depend, and whose impairment would cause such operations to halt.

### Third party

An organizational entity unrelated to the organization that calculates a metric, or supplies the source data for it. Note that “third-party” is a subjective term and may be interpreted differently by each recording entity. It may denote another group within the same corporation or an independent entity outside of the corporation.

### Vulnerability

A *vulnerability* is defined as a weakness in a system that could be exploited by an attacker to gain access or take actions beyond those expected or intended by the system’s security model. According to the definition used by CVE, Vulnerabilities are mistakes in software design and execution, while exposures are mistakes in configuration or mistakes in software used as a component of a successful attack. For the purposes of these metrics, the term vulnerabilities include exposures as well as technical vulnerabilities.

# Appendix B: Acknowledgements

Over one hundred (100) industry experts contributed prioritizing business functions and guiding the development of the consensus-based, metric definitions in this document. Additionally, the following people significantly contributed to the definition of these metrics: Kip Boyle, Rodney Caudle, Anton Chuvakin, Dean Farrington, Brad Gobble, Ben Hamilton, Pat Hymes, Andrew Jaquith, Adam Kliarsky, Clint Kreitner, David Lam, Charlie Legrand, Bill Marriott, Elizabeth Nichols, Orlando Padilla, Steven Piliero, Fred Pinkett, Mike Rothman, Andrew Sudbury, Chad Thunberg, Chris Walsh, Lilian Wang, Craig Wright, Chris Wysopal, Caroline Wong and others.

# Appendix C: Examples of Additional Metrics

The datasets provided can be used to create additional metrics to suit an organizations specific needs. For example, an organization focusing on incident containment could create additional incident metrics to track their ability to detect incidents internally as well as provide additional granularity around incident recovery by measuring the time from incident discovery to containment (as well as recovery). Two new metrics, “Percentage of Incidents detected by Internal Controls” and “Mean Time from Discover to Containment” can be created using the Incidents Dataset. Another organization may wish to focus on the patching process and provide the Mean-Time to Deploy metric just for critical patches as a key indicator to management. “Mean-Time to Deploy Critical Patches” can be created from the Patch datasets, using the severity field as a dimension to focus management attention on a key risk area. The following definitions of these additional metrics defined using the CIS datasets are provided below:

## Percentage of Incidents Detected by Internal Controls

### *Objective*

Percentage of Incidents Detected by Internal Controls (PIDIC) indicates the effectiveness of the security monitoring program.

Table 36: Percentage of Incidents Detected by Internal Controls

| | |
|---|---|
| **Metric Name** | Percentage of Incidents Detected by Internal Controls |
| **Version** | 0.9.0 |
| **Status** | Reviewed |
| **Description** | Percentage of Incidents Detected by Internal Controls (PIDIC) calculates the ratio of the incidents detected by standard security controls and the total number of incidents identified. |
| **Audience** | Operations |
| **Question** | Of all security incidents identified during the time period, what percent were detected by internal controls? |
| **Answer** | Positive floating point value between zero and 100. A value of "0" indicates that no security incidents were detected by internal controls and a value of "100" indicates that all security incidents were detected by internal controls. |
| **Formula** | Percentage of Incidents Detected by Internal Controls (PIDIC) is calculated by dividing the number of security incidents for which the Detected by Internal Controls field is equal to "true" by the total number of all known security incidents:<br>$PIDIC = \frac{Count(Incident\_DetectedByInternalControls = TRUE)}{Count(Incidents)} * 100$ |
| **Units** | Percentage of incidents |
| **Frequency** | Monthly, Quarterly, Annually |
| **Targets** | PIDIC values should trend higher over time. The value of "100%" indicates hypothetical perfect internal controls since no incidents were detected by outside parties. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Percentage of Incidents Detected by Internal Controls exists. |
| **Sources** | Since humans determine when an incident occurs, when the incident is contained, and when the incident is resolved, the primary data sources for this metric are manual inputs as defined in Security Incident Metrics: Data Attributes. However, these incidents may be reported by operational security systems, such as anti-malware software, security incident and event management (SIEM) systems, and host logs. |

### *Usage*

This metric measures the effectiveness of a security monitoring program by determining which incidents were detected by the organization's own internal activities (e.g. intrusion detection systems, log reviews, employee observations) instead of an outside source, such as a business partner or agency. A low value can be due to poor visibility in the environment, ineffective processes for discovering incidents, ineffective alert signatures and other factors. Organizations should report on this metric over time to show improvement of the monitoring program.

### *Limitations*

An organization may not have direct control over the percentage of incidents that are detected by their security program. For instance, if all the incidents that occur are due to zero-day or previously unidentified vectors then there are not many options left to improve posture. However, this metric could be used to show that improving countermeasures and processes within operations could increase the number of incidents that are detected by the organization.

### *References*

Scarfone, Grance and Masone. Special Publication 800-61 Revision 1:Computer Security Incident Handling Guide. US National Institute of Standards and Technology, 2004. <http://csrc.nist.gov/publications/nistpubs/800-61-rev1/SP800-61rev1.pdf>

Killcrece, Kossakowski, Ruefle and Zajicek. State of the Practice of Computer Security Incident Response Teams (CSIRTs). Carnegie-Mellon Software Engineering Institute, 2003. http://www.cert.org/archive/pdf/03tr001.pdf

Baker, Hylender and Valentine, 2008 Data Breach Investigations Report. Verizon Business RISK Team, 2008. <http://www.verizonbusiness.com/resources/security/databreachreport.pdf>

## Mean Time from Discovery to Containment

### *Objective*

Mean Time from Discovery to Containment (MTDC) characterizes the effectiveness of containing a security incident as measured by the average elapsed time between when the incident has been discovered and when the incident has been contained.

Table 37: Mean Time from Discovery to Containment

| | |
|---|---|
| **Metric Name** | Mean Time from Discovery to Containment |
| **Version** | 0.9.0 |
| **Status** | Reviewed |
| **Description** | Mean Time from Discovery to Containment (MTDC) measures the effectiveness of the organization to identify and contain security incidents. The sooner the organization can contain an incident, the less damage it is likely to incur. This calculation can be averaged across a time period, type of incident, business unit, or severity. |
| **Audience** | Operations |
| **Question** | What is the average (mean) number of hours from when an incident has been detected to when it has been contained? |
| **Answer** | A positive integer value that is greater than or equal to zero. A value of “0” indicates instantaneous containment. |
| **Formula** | For each incident contained in the metric time period, the mean time from discovery to containment is calculated dividing the difference in hours between the Date of Containment from the Date of Discovery for each incident by the total number of incidents contained in the metric time period:<br>$MTDC = \frac{\sum (Date\_of\_Containment - Date\_of\_Discovery)}{Count(Incidents)}$ |
| **Units** | Hours per incident |
| **Frequency** | Weekly, Monthly, Quarterly, Annually |
| **Targets** | MTDC values should trend lower over time. The value of “0” indicates hypothetical instantaneous containment. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Mean Time from Discovery to Containment exists. |
| **Sources** | Since humans determine when an incident occurs, when the incident is contained, and when the incident is resolved, the primary data sources for this metric are manual inputs as defined in Security Incident Metrics: Data Attributes. However, these incidents may be reported by operational security systems, such as anti-malware software, security incident |

| | |
|---|---|
| | and event management (SIEM) systems, and host logs. |

### Usage

MTDC is a type of security incident metric and relies on the common definition of "security incidents" as defined in *Glossary*.

An incident is determined to be "contained" when the immediate effect of the incident has been mitigated. For example, a DDOS attack has been throttled or unauthorized external access to a system has been blocked, but the system has not yet been fully recovered or business operations are not restored to pre-incident levels.

Optimal conditions would reflect a low value in the MTDC. A low MTDC value indicates a healthier security posture as malicious activity will have less time to cause harm. Given the modern threat landscape and the ability for malicious code to link to other modules once entrenched, there may be a direct correlation between a higher MTDC and a higher incident cost.

### Limitations

This metric measures incident containment capabilities of an organization. As such, the importance of this metric will vary between organizations. Some organizations have much higher profiles than others, and would thus be a more attractive target for attackers, whose attack vectors and capabilities will vary. As such, MTDCs may not be directly comparable between organizations.

In addition, the ability to calculate meaningful MTDCs assumes that incidents are detected. A lack of participation by the system owners could skew these metrics. A higher rate of participation in the reporting of security incidents can increase the accuracy of these metrics.

The date of occurrence of an incident may be hard to determine precisely. The date of occurrence field should be the date that the incident could have occurred no later than given the best available information. This date may be subject to revision and more information becomes known about a particular incident.

Incidents can vary in size and scope. This could result in a variety of containment times that, depending on its distribution, may not provide meaningful comparisons between organizations when mean values are used.

### References

Scarfone, Grance and Masone. Special Publication 800-61 Revision 1:Computer Security Incident Handling Guide. US National Institute of Standards and Technology, 2004. <http://csrc.nist.gov/publications/nistpubs/800-61-rev1/SP800-61rev1.pdf>

Killcrece, Kossakowski, Ruefle and Zajicek. State of the Practice of Computer Security Incident Response Teams (CSIRTs). Carnegie-Mellon Software Engineering Institute, 2003. http://www.cert.org/archive/pdf/03tr001.pdf

Baker, Hylender and Valentine, 2008 Data Breach Investigations Report. Verizon Business RISK Team, 2008. <http://www.verizonbusiness.com/resources/security/databreachreport.pdf>

## Mean Time to Deploy Critical Patches

### *Objective*

Mean Time to Deploy Critical Patches (MTDCP) characterizes effectiveness of the patch management process by measuring the average time taken from notification of critical patch release to installation in the organization. This metric serves as an indicator of the organization's exposure to severe vulnerabilities by measuring the time taken to address systems in known states of high vulnerability for which security patches are available. This is a partial indicator as vulnerabilities may have no patches available or occur for other reasons such as system configurations.

Table 38: Mean Time to Deploy Critical Patches

| | |
|---|---|
| **Metric Name** | Mean Time to Deploy Critical Patches |
| **Version** | 0.9.0 |
| **Status** | Draft |
| **Description** | Mean Time to Patch Deploy Patches (MTPCP) measures the average time taken to deploy a critical patch to the organization's technologies. The sooner critical patches can be deployed, the lower the mean time to patch and the less time the organization spends with systems in a state known to be vulnerable.<br>In order for managers to better understand the exposure of their organization to vulnerabilities, Mean Time to Deploy Critical Patches should be calculated for the scope of patches with Patch Criticality levels of "Critical". This metric result, reported separately provides more insight than a result blending all patch criticality levels as seen in the Mean Time to Patch metric. |
| **Audience** | Management |
| **Question** | How many days does it take the organization to deploy critical patches into the environment? |
| **Answer** | A positive floating-point value that is greater than or equal to zero. A value of "0" indicates that critical patches were theoretically instantaneously deployed. |
| **Formula** | Mean Time to Deploy Critical Patches is calculated by determining the number of hours between the Date of Notification and the Date of Installation for each critical patch completed in the current scope, for example by time period or business unit. These results are then averaged across the number of completed critical patches in the current scope:<br>$MTDCP = \frac{\sum(Date\_of\_Installation - Date\_of\_Notification)}{Count(Completed\_Critical\_Patches)}$ |
| **Units** | Hours per patch |
| **Frequency** | Weekly, Monthly, Quarterly, Annually |
| **Targets** | MTDCP values should trend lower over time. Most organizations put critical patches |

through test and approval cycles prior to deployment. Generally, the target time for Mean Time to Deploy Critical Patches is within several hours to days. Because of the lack of experiential data from the field, no consensus on the range of acceptable goal values for Mean Time to Deploy Critical Patches exists.

### *Usage*

Mean Time to Deploy Critical Patches is a type of patch management metric, and relies on the common definition of “patch” as defined in *Glossary*.

Given that many known severe vulnerabilities result from missing critical patches, there may be a direct correlation between lower MTDCP and lower levels of Security Incidents. MTDCP can be calculated over time, typically per-week or per-month. To gain insight into the relative performance and risk to one business unit over another, MTDCP can be compared against MTTP by cross-sections of the organization such as individual business units or geographies.

### *Limitations*

Critical Technologies. This metric assumes that the critical technologies are known and recorded. If the critical technologies are unknown, this metric cannot be accurately measured. As new technologies are added their criticality needs to be determined and, if appropriate, included in this metric.

Vendor Reliance. This metric is reliant upon the vendor’s ability to notify organization of updates and vulnerabilities that need patching. If the vendor does not provide a program for notifying their customers then the technology, if critical, will always be a black mark on this metric.

Criticality Ranking. This metric is highly dependent upon the ranking of critical technologies by the organization. If this ranking is abused then the metric will become unreliable.

Patches in Progress. This metric calculation does not account for patch installations that are incomplete or on-going during the time period measured. It is not clear how this will bias the results, although potentially an extended patch deployment will not appear in the results for some time.

### *References*

Mell, Bergeron and Henning. Special Publication 800-40: Creating a Patch and Vulnerability Management Program. US National Institute of Standards and Technology, 2005.

# Index of Tables