The Center for Internet Security

The CIS Security Metrics

November 1st

2010

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 (150) 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-eight (28) metric definitions for seven (7) important business functions: Incident Management, Vulnerability Management, Patch Management, Application Security, Configuration Management, Change Management and Financial Metrics

CIS Security Metrics v1.1.0

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?	• Cost of Incidents • Mean Cost of Incidents • Mean Incident Recovery Cost • Mean-Time to Incident Discovery • Number of Incidents • Mean-Time Between Security Incidents • 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 • Percent of Systems with No Known Severe Vulnerabilities • Mean-Time to Mitigate Vulnerabilities • Number of Known Vulnerability Instances • Mean Cost to Mitigate Vulnerabilities
Patch Management	How well are we able to maintain the patch state of our systems?	• Patch Policy Compliance • Patch Management Coverage • Mean-Time to Patch • Mean Cost to Patch
Configuration Management	What is the configuration state of systems in the organization?	• Percentage of Configuration Compliance • Configuration Management Coverage • Current Anti-Malware Compliance
Change	How do changes to system configurations affect the	• Mean-Time to Complete Changes

Management	security of the organization?	• Percent of Changes with Security Reviews • Percent of Changes with Security Exceptions
Application Security	Can we rely on the security model of business applications to operate as intended?	• Number of Applications • Percent of Critical Applications • Risk Assessment Coverage • Security Testing Coverage
Financial Metrics	What is the level and purpose of spending on information security?	• IT Security Spending as % of IT Budget • IT Security Budget Allocation
Future Functions	Community recommendations for additional business functions include:	• Data / Information • Software Development Life-Cycle • Configuration Management • Third Party Risk Management • Additional Financial and Impact Metrics • Authentication and Authorization • Data and Network Security • Remediation Efforts • Anti-Malware Controls

Metric Categories

Metrics are organized into a hierarchy based on their purpose and audience. Management metrics are generally the most valuable to the organization but may require that foundational technical metrics be in place.

Table 2: Metric Categories

Metric Categories		
Management Metrics	Provide information on the performance of business functions, and the impact on the organization. Audience: Business	• Cost of Incidents • Mean Cost of Incidents • Percent of Systems with No Known Severe Vulnerabilities • Patch Policy Compliance • Percentage of Configuration Compliance

	Management	• Percent of Changes with Security Reviews • IT Security Spending as % of IT Budget
Operational Metrics	Used to understand and optimize the activities of business functions. Audience: Security Management	• Mean Incident Recovery Cost • Mean-Time to Incident Discovery • Mean-Time Between Security Incidents • Mean-Time to Incident Recovery • Mean-Time to Mitigate Vulnerabilities • Mean Cost to Mitigate Vulnerabilities • Mean Cost to Patch • Mean-Time to Patch • Mean-Time to Complete Changes • Percent of Changes with Security Exceptions • IT Security Budget Allocation
Technical Metrics	Provide technical details as well as a foundation for other metrics. Audience: Security Operations	• Number of Incidents • Vulnerability Scanning Coverage • Number of Known Vulnerability Instances • Patch Management Coverage • Configuration Management Coverage • Current Anti-Malware Compliance • Number of Applications • Percent of Critical Applications • Risk Assessment Coverage • Security Testing Coverage

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.

Updating Data over Time

It is possible that data gathered for metrics may change over time. For example, the number of affected records or hosts may change during the investigation of an incident. Metric values should be calculated using the current best known data values that can be provided at the time of metric calculation. A data element should only be included in a metric calculation if data is available. For example if it is only known that an incident has occurred but no analysis of the scope has occurred by the calculation date of monthly incident metrics, that incident should be included in incident counts but not included in calculations of mean records lost. When updated data is available it should be included in future metric calculations and updated values should be used when presenting metric results. Later, additional metrics could be added later to compare estimates to later (actual) values.

Data Attributes

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

Table 3: Security Incidents Table

The Security Incident Table contains information regarding each of the incidents discovered by the organization.

Security Incidents Table

Name	Type	De-Identified	Required	Description
Incident ID	Number	No	Yes	Unique identifier for the incident. Generally auto-generated.
Technology ID	Text / Number	Yes	No	Unique identifier for the technology. Generally auto-

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

				generated.
Event ID	Text / Number	No	No	Unique identifier for the event.
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	Unique identifier for the entity that first discovered the incident.
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.
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.
Report ID	Number	Yes	No	Unique identifier for reporting of incident.
Incident Analysis ID	Number	No	No	Unique identifier for incident analysis.
Attacker	Text	No	No	Type of attacker. Use values Hackers, Spies, Terrorists, Corporate Raiders, Professional Criminals, Vandals, or Voyeurs.

Table 4: Security Incident Classification Table

The Security Incident Classification Table contains information regarding the classification of incidents using taxonomies agreed upon by the organization.

Security Incident Classification Table				
Name	**Type**	**De-Identified**	**Required**	**Description**
Incident ID	Number	No	No	Unique identifier for the incident. Generally auto-generated.
Incident Name	Text	No	No	Name of the incident.
Incident Description	Text	No	No	Description of the incident.
Classification	Text	No	No	Classification of the incident using Howard-Longstaff taxonomy
Additional Classification	Text	No	No	Additional, optional classifications of the incident for internal or other reporting purposes. Incidents may include more than one tag.
Effect Rating	Text	Yes	No	Estimated effect of the incident on the organization, using the US-CERT effect table.
Criticality Rating	Text	Yes	No	Criticality of the systems involved in this incident, using the US-CERT criticality table.
Additional Priority	Text	No	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	No	No	The ISO codes of the country where the target company/server(s) reside.

Table 3: Security Events Table

The Security Events Table contains information regarding the relationship among security events and incidents.

Security Events Table				
Name	**Type**	**De-Identified**	**Required**	**Description**
Event ID	Number	No	Yes	Unique identifier for the event.
Event Name	Text	No	No	Name of the event.
Date of Occurrence	Date / Time	No	No	Date and time the event occurred.
Date of Discovery	Date / Time	No	No	Date and time the event was discovered.
Discovered By	Text	No	No	Unique identifier for the entity that first discovered the event.
Attacker	Text	No	No	Type of attacker. Use values Hackers, Spies, Terrorists, Corporate Raiders, Professional Criminals, Vandals, or Voyeurs.
Tool	Text	No	No	Type of tool used. Use values Physical Attack, Information Exchange, User Command, Script or Program, Autonomous Agent, Toolkit, Distributed Tool, or Data Tap.
Vulnerability	Text	No	No	Type of vulnerability exploited. Use values Design, Implementation, or Configuration.
Action	Text	No	No	Type of action performed. Use values Probe, Scan, Flood, Authenticate, Bypass, Spoof, Read, Copy, Steal, Modify, Delete, Target, Account, Process, Data, Component, Computer, Network, or Internetwork.
Objective	Text	No	No	Reason for attack. Use values Challenge, Status, Thrill, Political Gain, Financial Gain, or Damage.

Table 4: Security Incident Impact Analysis Table

The Security Incident Impact Analysis Table contains information resulting from the review and analysis of security incidents that occurred within the organization.

Security Incident Impact Analysis Table				
Name	**Type**	**De-Identified**	**Required**	**Description**

Incident Analysis ID	Number	No	Yes	Unique identifier for incident analysis.
Incident ID	Number	No	No	Unique identifier for the incident.
Technology ID	Text / Number	Yes	No	Unique identifier for the technology.
Vulnerability ID	Text / Number	No	No	Unique identifier for the vulnerability instance.
Detected by Internal Controls	Boolean	No	No	Whether the incident was detected by a control operated by the organization.
Response Protocol Followed	Boolean	No	No	Whether incident response protocol was followed.
Business Continuity Plan Executed	Boolean	No	No	Whether business continuity plan was executed following incident.
Reoccurring	Boolean	No	No	Whether incident has occurred before.
Root Cause	Text	No	No	Text description of the root cause of the incident.
Direct Loss Amount	Number	No	No	Quantifiable, direct financial loss verified by management due to money, IP or other assets lost or stolen.
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.
Cost of Business System Downtime	Number	No	No	Total losses (if any) attributed to the time businesss systems were unavailable or non-operational.
Cost of Containment	Number	No	No	Total cost to contain incident.
Cost of Recovery	Number	No	No	Total cost to recover from incident for effort and equipment and costs to repair or replace affected systems.
Customers Affected	Boolean	No	No	Whether or not customer data was affected by the incident.
Loss of Personally Identifiable Information	Boolean	No	No	Whether or not PII was lost during the incident.
Data Types Lost	Text	No	No	CCN (Credit Card Numbers) SSN (Social Security Numbers or Non-

				US Equivalent) NAA (Names and/or Addresses) EMA(Email Addresses) MISC (Miscellaneous) MED (Medical) ACC(Financial Account Information) DOB (Date of Birth) FIN (Financial Information)
Records Affected	Number	No	No	Total number of records affected in data breach incidents.
Cost of Restitution	Number	No	No	Total cost of notification, restitution and additional security services offered to affected customers in data breach incidents.
PCI Penalties	Number	No	No	Total cost of PCI penalties defined by PCI DSS.

Table 5: Security Incident Reporting Table

The Security Incident Reporting Table contains information regarding the incident reports the organization may have published. These reports may fulfill internal management requests or external governance and compliance requirements.

Security Incident Reporting Table				
Name	**Type**	**De-Identified**	**Required**	**Description**
Report ID	Number	Yes	No	Unique identifier for reporting of incident.
Report Date	Date/Time	No	No	Date incident was reported.
Internal	Boolean	No	No	Whether report is internal or external.
Industry Sector	Text	No	No	Sector the organization belongs to.
Organization Size	Number	No	No	Size of the organization.

Table 6: Technologies Table

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

Technologies Table

Name	Type	De-identified	Required	Description
Technology ID	Text / Number	No	Yes	Unique identifier for the technology. Generally auto-generated.
Name	Text	No	No	Name from CPE Dictionary which follows the following structure: cpe:/{PART}:{VENDOR}:{PRODUCT}:{VERSION}:{UPDATE}:{EDITION}:{LANGUAGE}.
Part	Text	No	No	Platform. Use value: H, O, or A. H, O, and A represent hardware, operating system, and application environment respectively.
Vendor	Text	No	No	Vendor from CPE Dictionary. This is the highest organization-specific label of the DNS name.
Product	Text	No	No	Product from CPE Dictionary. This is the most recognizable name of the product.
Version	Text	No	No	Version from CPE Dictionary. Same format as seen with the product.
Update	Text	No	No	Update or service pack information from CPE Dictionary.
Edition	Text	No	No	Edition from CPE Dictionary. May define specific target hardware and software architectures.
Language	Text	No	No	Language 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*.[2]
Business Unit	Text	No	No	Organizational business unit that the technology belongs to.
Owner	Text	No	No	Unique identifier for individual within the organization that is responsible for the technology.
Classification	Text	No	No	Classification of technology: Servers, Workstations, Laptops, Network Device, Storage Device, Applications, Operating systems

Table 7: Effect Rating Table

The Effect Rating Table contains the values for the Effect Rating dimension used in the Security Incident Classification Table.

Security Incident Effect Rating Table

Value	Rating	Definition
0.00	None	No effect on a single agency, multiple agencies, or critical infrastructure
0.10	Minimal	Negligible effect on a single agency
0.25	Low	Moderate effect on a single agency
0.50	Medium	Severe effect on a single agency or negligible effect on multiple agencies or critical infrastructure
0.75	High	Moderate effect on multiple agencies or critical infrastructure
1.00	Critical	Severe effect on multiple agencies or critical infrastructure

Table 8: Criticality Rating Table

The Criticality Rating Table contains the values for the Criticality Rating dimension used in the Security Incident Classification Table.

Security Incident Criticality Rating Table

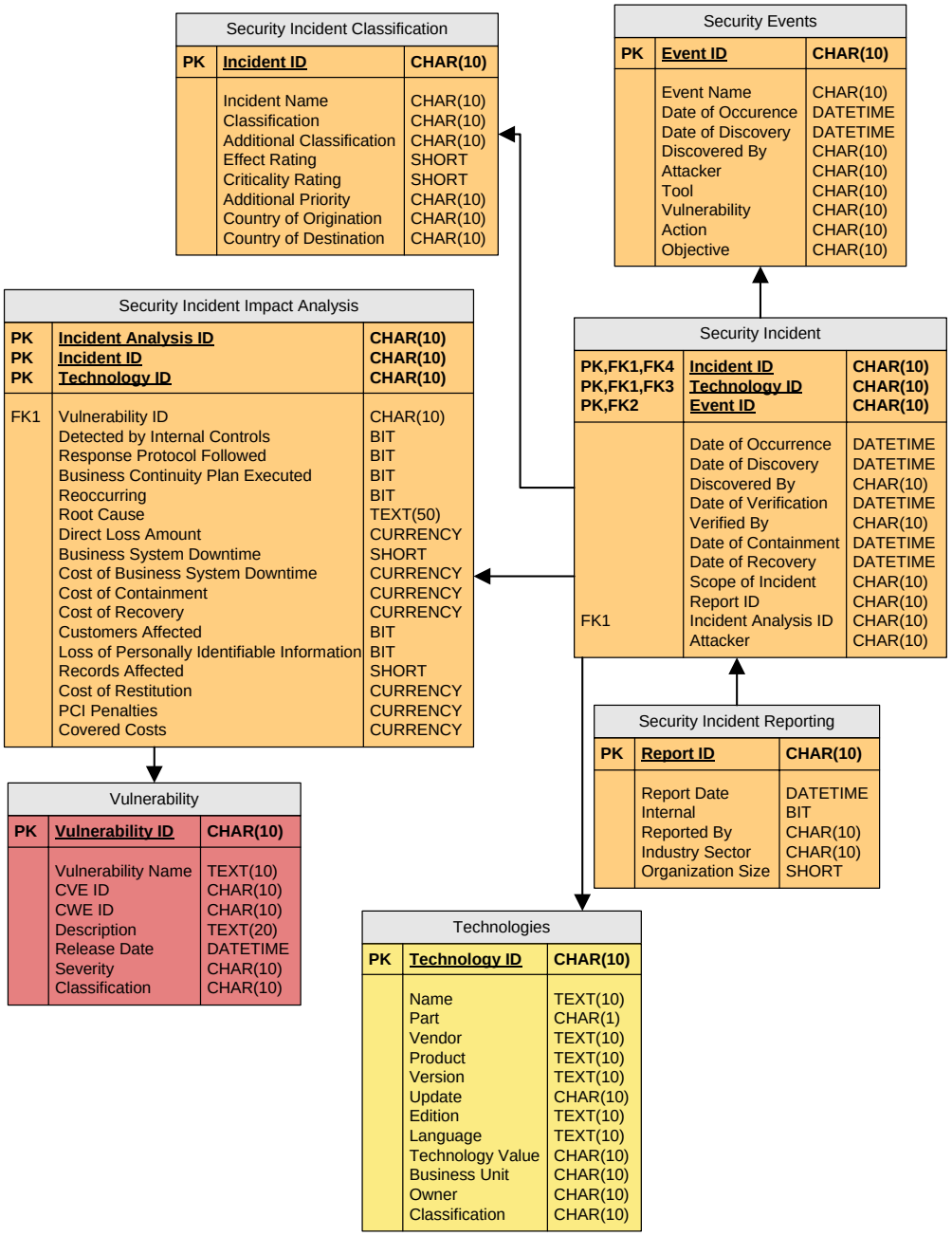
Value	Rating	Definition
0.10	Minimal	Non-critical system (e.g., employee workstations), systems, or infrastructure

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

0.25	Low	System or systems that support a single agency's mission (e.g., DNS servers, domain controllers) but are not mission critical
0.50	Medium	System or systems that are mission critical (e.g., payroll system) to a single agency
0.75	High	System or systems that support multiple agencies or sectors of the critical infrastructure (e.g., root DNS servers)
1.00	Critical	System or systems that are mission critical to multiple agencies or critical infrastructure

The diagram below shows the relationship of tables described in Incident Management Data Attributes:

Diagram 1: Relational Diagram for Incidents Data Attributes



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[3], 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

Howard and Longstaff[4] recommend the following taxonomy:

- Attackers – an individual who attempts one or more attacks in order to achieve an objective
  - Hackers – attackers who attack computers for challenge, status or the thrill of obtaining access
  - Spies – attackers who attack computers for information to be used for political gain
  - Terrorists – attackers who attack computers to cause fear for political gain
  - Corporate Raiders – employees who attack competitor's computers for financial gain
  - Professional Criminals – attackers who attack computers for personal financial gain
  - Vandals – attackers who attack computers to cause damage
  - Voyeurs – attackers who attack computers for the thrill of obtaining sensitive information
- Tool – a means that can be used to exploit a vulnerability in a computer or network
  - Physical Attack – a means of physically stealing or damaging a computer, network, its components, or its supporting systems
  - Information Exchange – a means of obtaining information either from other attackers, or from the people being attacked
  - User Command – a means of exploiting a vulnerability by entering commands to a process through direct user input at the process interface
  - Script or Program – a means of exploiting a vulnerability by entering commands to a process through the execution of a file of commands or a program at the process interface

[3] 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>

[4] Howard & Longstaff. A Common Language for Computer Security Inicdents. (October 1998).

  - Autonomous Agent – a means of exploiting a vulnerability by using a program, or program fragment, which operates independently from the user
  - Toolkit – a software package which contains scripts, programs, or autonomous agents that exploit vulnerabilities
  - Distributed Tool – a tool that can be distributed to multiple hosts
  - Data Tap – a means of monitoring the electromagnetic radiation emanating from a computer or network using an external device
- Vulnerability – a weakness in a system allowing unauthorized action
  - Design – a vulnerability inherent in the design or specification of hardware or software whereby even a perfect implementation will result in a vulnerability
  - Implementation – a vulnerability resulting from an error made in the software or hardware implementation of a satisfactory design
  - Configuration – a vulnerability resulting from an error in the configuration of a system
- Action – a step taken by a user or process in order to achieve a result
  - Probe – an action used to determine the characteristics of a specific target
  - Scan – an action where a user or process accesses a range of targets sequentially in order to determine which targets have a particular characteristic
  - Flood – access a target repeatedly in order to overload the target's capacity
  - Authenticate – an action taken by a user to assume an identity
  - Bypass – an action taken to avoid a process by using an alternative method to access a target
  - Spoof – an active security attack in which one machine on the network masquerades as a different machine
  - Read – an action to obtain the content of the data contained within a file or other data medium
  - Copy – reproduce a target leaving the original target unchanged
  - Steal – an action that results in the target coming into the possession of the attacker and becoming unavailable to the original owner or user
  - Modify – change the content of characteristics of a target
  - Delete – remove a target or render it irretrievable
- Target
  - Account – a domain of user access on a computer or network which is controlled according to a record of information which contains the user's account name, password, and user restrictions
  - Process – a program in execution, consisting of the executable program, the program's data and stack, its program counter, stack point and other registers, and all other information needed to execute the program

  - Data – representations of fact, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means
  - Component – one of the parts that make up a computer or network
  - Computer – a device that consists of one or more associated components
  - Network – an interconnected or interrelated group of host computers, switching elements, and interconnecting branches
  - Internetwork – a network of networks
- Unauthorized Result – an unauthorized consequence of an event
  - Increased Access – an unauthorized increase in the domain of access on a computer or network
  - Disclosure of Information – dissemination of information to anyone who is not authorized to access that information
  - Corruption of Information – unauthorized alteration of data on a computer or network
  - Denial of Service – intentional degradation or blocking of computer or network resources
  - Theft of Resources – unauthorized use of computer or network resources
- Objectives
  - Challenge, Status, Thrill
  - Political Gain
  - Financial Gain
  - Damage

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)"[5]. 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)
- [Schultz 90] — Priority 1 (human life, human safety) to Priority 5 (minimize disruption to computing processes)

[5] 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>

- [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 9: 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.
Type	Operational
Audience	Security Management
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:

	$$MTTID = \frac{\sum(Date\_of\_Discovery - Date\_of\_Occurrence)}{Count(Incidents)}$$
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.
Visualization	Column Chart x-axis: Time (Week, Month, Quarter, or Year) y-axis: MTTID (Hours per Incident)

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>

Mean Time between Security Incidents

Objective

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

Table 10: 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.
Type	Operational
Audience	Security Management
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: $$MTBSI = \frac{\sum(Date\_of\_Occurence[Incident_n] - Date\_of\_Occurence[Incident_{n-1}])}{Count(Incidents)}$$
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.
Visualization	Bar Chart X-axis: Time (Week, Month, Quarter, Year) Y-axis: MTBSI (Hours per Incident)

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 11: 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.
Type	Operational
Audience	Business Management, Security Management
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 (MTIR) 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 $MTIR = \frac{\sum(Date\_of\_Recovery - Date\_of\_Occurrence)}{Count(Incidents)}$
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.
Visualization	Bar Chart X-axis: Time (Week, Month, Quarter, Year) Y-axis: MTIR (Hours per Incident)

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

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

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>

Cost of Incidents

Objective

Organizations need to understand the impact of security incidents. Impact can take many forms from negative publicity to money directly stolen. Monetary costs provide a set of units that can be directly compared across impact of the incidents and across organizations.

In order to make effective risk management decisions, the impact of incidents needs to be measured and considered. Understanding of the costs experienced by the organization can be used to improve security process effectiveness and efficiency.

Table 12: Cost of Incidents

Metric Name	Cost of Incidents
Version	1.0.0
Status	Final Draft for Review

Description	Cost of Incidents (COI) measures the total cost to the organization from security incidents occurring during the metric time period. Total costs from security incidents consists of the following costs:
	• Direct Loss
	  ○ Value of IP, customer lists, trade secrets, or other assets that are destroyed
	• Cost of Business System Downtime
	  ○ Cost of refunds for failed transactions
	  ○ Cost of lost business directly attributable to the incident
	• Cost of Containment
	  ○ Efforts and cost
	  ○ Consulting services
	• Cost of Recovery
	  ○ Cost of incident investigation and analysis
	  ○ Effort required to repair and replace systems
	  ○ Replacement cost of systems
	  ○ Consulting services for repair or investigation
	  ○ Additional costs not covered by an insurance policy
	• Cost of Restitution
	  ○ Penalties and other funds paid out due to breaches of

	contacts or SLAs resulting from the incident ○ Cost of services provided to customers as a direct result of the incident (e.g. ID Theft Insurance) ○ Public relations costs ○ Cost of disclosures and notifications ○ Legal costs, fines, and settlements
Type	Management
Audience	Business Management, Security Management
Question	What is the total cost to the organization from security incidents during the given period?
Answer	A positive integer value that is greater than or equal to zero. A value of "0.0" indicates there were no measured costs to the organization.
Formula	Cost of Incidents (COI) is calculated by summing all costs associated with security incidents during the time period: COI = ∑(Direct Loss + Cost of Business System Downtime + Cost of Containment + Cost of Recovery + Cost of Restitution)
Units	$USD per incident
Frequency	Monthly
Targets	Ideally there would be no security incidents with material impacts on the organization, and the metric value would be zero. In practice a target can be set based on the expected loss budget determined by risk assessments processes.
Sources	Incident tracking systems will provide incident data. Cost data can come from both management estimates, ticket tracking systems and capital and services budgets.
Visualization	Bar Chart X-axis: Time (Month) Y-axis: COI ($)

Usage

Cost of Incidents (COI) represents the overall known outcome of security systems, processes, and policies. The lower the COI, the less the organization is impacted by security incidents.

Optimal conditions would reflect a low value of COI. Costs experienced by organizations may vary as a result of the threat environment, controls in place, and resiliency of the organization. Over time as processes and controls become more effectiveness, COI should be reduced.

Limitations

- Some incidents such as exposure of business strategy via social engineering may not have a direct incident costs. Significant harm, bad press, or competitive disadvantage may still be experienced for which it is not practical to assign a cost.
- Some new controls may have significant costs and/or address recovery from multiple incidents.
- This metric relies on the common definition of "security incident" as defined in Terms and Definitions.
- This metric relies on an organization being able to produce costs or cost estimates related to security incidents.

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 COI 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

References

"Computer Security Incident Handling Guide". NIST Special Publication 800-61. National Institute of Standards and Technology. January 2004.

Dorofee, Killcrece, et al. "Incident Management Capability Metrics Version 0.1", Software Engineering Institute. April, 2007.

"Incident Cost Analysis and Modeling Project (ICAMP) Final Report 2". Committee on Institutional Cooperation Security Working Group. 2000.

Killcrece, Kossakowski, Ruefle and Zajicek. "State of the Practice of Computer Security Incident Response Teams" Carnegie-Mellon Software Engineering Institute, October 2003.

West-Brown, Stikvoort, et al. "Handbook for Computer Security Incident Response Teams (CSIRTs)". Carnegie Mellon Software Engineering Institute. April 2003.

Mean Cost of Incidents

Objective

Organizations need to understand the impact of security incidents. Impact can take many forms from negative publicity to money directly stolen. Monetary costs provide a set of units that can be directly compared across impact of the incidents and across organizations.

In order to make effective risk management decisions, the impact of incidents needs to be measured and considered. Understanding of the mean costs the organization incurs from security incidents allows the organization to improve security process effectiveness and efficiency.

Table 13: Mean Cost of Incidents

Metric Name	Mean Cost of Incidents
Version	1.0.0
Status	Final Draft for Review

Description	Mean Cost of Incidents (MCOI) measures the mean cost to the organization from security incidents identified relative to the number of incidents that occurred during the metric time period. Total costs from security incidents consists of the following costs:
	• Direct Loss
	  ○ Value of IP, customer lists, trade secrets, or other assets that are destroyed
	• Cost of Business System Downtime
	  ○ Cost of refunds for failed transactions
	  ○ Cost of lost business directly attributable to the incident
	• Cost of Containment
	  ○ Efforts and cost
	  ○ Consulting services
	• Cost of Recovery
	  ○ Cost of incident investigation and analysis
	  ○ Effort required to repair and replace systems
	  ○ Replacement cost of systems
	  ○ Consulting services for repair or investigation
	  ○ Additional costs not covered by an insurance policy
	• Cost of Restitution

	○ Penalties and other funds paid out due to breaches of contacts or SLAs resulting from the incident ○ Cost of services provided to customers as a direct result of the incident (e.g. ID Theft Insurance) ○ Public relations costs ○ Cost of disclosures and notifications ○ Legal costs, fines, and settlements
Type	Management
Audience	Business Management, Security Management
Question	What is the average (mean) cost to the organization from security incidents during the given period?
Answer	A positive integer value that is greater than or equal to zero. A value of "0.0" indicates there were no measured costs to the organization.
Formula	Mean Cost of Incidents (MCOI) is calculated by summing all costs associated with security incidents by the number of security incidents that occurred during the time period: $$MCOI = \frac{\sum (Direct\_Loss + Cost\_Business\_Downtime + Cost\_Containment + Cost\_Recovery + Cost\_Restitution)}{Count(Incidents)}$$
Units	$USD per incident
Frequency	Monthly
Targets	Ideally there would be no security incidents with material impacts on the organization, and the metric value would be zero. In practice a target can be set based on the expected loss budget determined by risk assessments processes.
Sources	Incident tracking systems will provide incident data. Cost data can come from both management estimates, ticket tracking systems and capital and services budgets.
Visualization	Bar Chart X-axis: Time (Month) Y-axis: MCOI ($/Incident)

Usage

Mean Cost of Incidents (MCOI) represents the average impact of a security incident on the organization. This impact is the average known outcome resulting from the interaction of the threat environment with the security systems, processes, and policies or the organization. The lower the MCOI, the less the organization is impacted by security incidents on average. Optimal conditions would reflect a low value of MCOI. Costs experienced by organizations can vary as a result of the threat environment, systems and processes in place, and resiliency of the organization. Over time, the effectiveness of changes to an organization's security activities should result in a reduction in the Mean Cost of Incidents.

MCOI should provide a management indicator of the ability of the organization to alter the known impact expected from security incidents.

Limitations

- Some incidents such as exposure of business strategy via social engineering may not have a direct incident costs. Significant harm, bad press, or competitive disadvantage may still be experienced for which it is not practical to assign a cost.
- Some new controls may have significant costs and/or address recovery from multiple incidents.
- This metric relies on the common definition of "security incident" as defined in Terms and Definitions.
- This metric relies on an organization being able to produce costs or cost estimates related to security incidents.

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 COI 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

References

"Computer Security Incident Handling Guide". NIST Special Publication 800-61. National Institute of Standards and Technology. January 2004.

Dorofee, Killcrece, et al. "Incident Management Capability Metrics Version 0.1", Software Engineering Institute. April, 2007.

"Incident Cost Analysis and Modeling Project (ICAMP) Final Report 2". Committee on Institutional Cooperation Security Working Group. 2000.

Killcrece, Kossakowski, Ruefle and Zajicek. "State of the Practice of Computer Security Incident Response Teams" Carnegie-Mellon Software Engineering Institute, October 2003.

West-Brown, Stikvoort, et al. "Handbook for Computer Security Incident Response Teams (CSIRTs)". Carnegie Mellon Software Engineering Institute. April 2003.

Mean Incident Recovery Cost

Objective

Mean Incident Recovery Cost measures the total costs directly associated with the operational recovery from an incident. While the impact of similar incidents may vary across organizations, the technical recovery should be comparable on a per-system basis across firms.

Table 14: Mean Cost of Incidents

Metric Name	Mean Incident Recovery Cost
Version	1.0.0
Status	Final Draft for Review
Description	Mean Incident Recovery Cost (MIRC) measures the cost of returning business systems to their pre-incident condition. The following costs may be taken into consideration: • Cost to repair and/or replace systems • Opportunity cost of staff implementing incident handling plan • Cost to hire external technical consultants to help recover from the incident • Cost to installation new controls or procurement of new resources that directly addresses the re-occurrence of the incident (e.g. installation of AV software) • Legal and regulatory liabilities resulting from the incident
Type	Operational

Audience	Security Management
Question	What is the average (mean) cost of recovery from a security incidents during the given period?
Answer	A positive integer value that is greater than or equal to zero. A value of "0.0" indicates there were no measured costs to the organization.
Formula	Mean Incident Recovery Cost (MIRC) is calculated by summing all costs associated with recovering from security incidents by the number of security incidents that occurred during the time period: $MIRC = \frac{\sum(Cost\_Recovery)}{Count(Incidents)}$
Units	$USD per incident
Frequency	Monthly
Targets	Ideally, recovery from security incidents would have no material impacts on the organization, and the metric value would be zero. In practice a target can be set based on the expected loss budget determined by risk assessments processes, and planned incident recovery resources.
Sources	Incident tracking systems will provide incident data. Cost data can come from management estimates, ticket tracking systems and capital and services budgets.
Visualization	Bar Chart X-axis: Time (Month) Y-axis: MIRC ($/Incident)

Usage

Mean Incident Recovery Cost (MIRC) represents the average cost the organization incurs while recovering from a security incident. This cost is correlated to the capabilities and resiliency of the systems, processes, and policies. The lower the MIRC, the less the organization is impacted by security incidents on average, and the greater the general resiliency of the organization's systems. Optimal conditions would reflect a low value of MIRC. Costs experienced by organizations can vary as a result of the threat environment, systems, and processes in place. Over time, the effectiveness of changes to an organization's security activities should result in a reduction in the Mean Incidents Recovery Cost.

MIRC should provide a management indicator of the expected ability of the organization's resiliency and ability to recover from security incidents.

Limitations

- Some incidents, such as theft via social engineering, may not have a direct recovery costs as there may not be a clear end point or may be the result of several related incidents
- Establishment of new controls or procurement of new resources may have significant costs.
- This metric is dependent upon when during the incident management process cost information is collected. Depending if information is collected during the occurrence of the incident or following the incident may influence the metric outcome.

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 MIRC 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

References

"Computer Security Incident Handling Guide". NIST Special Publication 800-61. National Institute of Standards and Technology. January 2004.

Dorofee, Killcrece, et al. "Incident Management Capability Metrics Version 0.1", Software Engineering Institute. April, 2007.

"Incident Cost Analysis and Modeling Project (ICAMP) Final Report 1". Committee on Institutional Cooperation Security Working Group. 1998.

"Incident Cost Analysis and Modeling Project (ICAMP) Final Report 2". Committee on Institutional Cooperation Security Working Group. 2000.

Killcrece, Kossakowski, Ruefle and Zajicek. "State of the Practice of Computer Security Incident Response Teams" Carnegie-Mellon Software Engineering Institute, October 2003.

West-Brown, Stikvoort, et al. "Handbook for Computer Security Incident Response Teams (CSIRTs)". Carnegie Mellon Software Engineering Institute. April 2003.

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).

Table 15: Technologies Table

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

Technologies Table

Name	Type	De-identified	Required	Description
Technology ID	Text / Number	No	Yes	Unique identifier for the technology. Generally auto-generated.
Name	Text	No	No	Name from CPE Dictionary which follows the following structure: cpe:/{PART}:{VENDOR}:{PRODUCT}:{VERSION}:{UPDATE}:{EDITION}:{LANGUAGE}.
Part	Text	No	No	Platform. Use value: H, O, or A. H, O, and A represent hardware, operating system, and application environment respectively.
Vendor	Text	No	No	Vendor from CPE Dictionary. This is the highest organization-specific label of the DNS name.
Product	Text	No	No	Product from CPE Dictionary. This is the most recognizable name of the product.
Version	Text	No	No	Version from CPE Dictionary. Same format as seen with the product.
Update	Text	No	No	Update or service pack information from CPE Dictionary.
Edition	Text	No	No	Edition from CPE Dictionary. May define specific target hardware and software architectures.
Language	Text	No	No	Language 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]
Business Unit	Text	No	No	Organizational business unit that the technology belongs to.
Owner	Text	No	No	Unique identifier for individual within the organization that is responsible for the technology.
Classification	Text	No	No	Classification of technology: Servers, Workstations, Laptops, Network Device, Storage Device, Applications, Operating systems

Table 16: Vulnerability Information Table

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:

Vulnerability Information Table

Name	Type	De-identified	Required	Description
Vulnerability ID	Text / Number	No	Yes	Unique identifier for the vulnerability. Generally auto-generated. This can be an organization-specific identifier for the vulnerability.
Vulnerability Name	Text	No	No	Name of the vulnerability.
CVE ID	Number	No	No	Common Vulnerability Enumeration

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

				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 NVD or elsewhere)
Release Date	Date / Time	No	No	Date that the vulnerability was made publicly known.
Severity	Text	No	No	Severity rating for the vulnerability. May use Low, Medium, or High.
Classification	Text	No	No	Classification of the vulnerability.

Table 17: CVSS Score Table

The Common Vulnerability Scoring System (CVSS) score for each of the vulnerabilities computed.

CVSS Score Table

Name	Type	De-identified	Required	Description
Vulnerability ID	Text/Number	No	Yes	Unique identifier for the vulnerability.
Overall CVSS Score	Number	No	No	Overall CVSS Score
CVSS Base Score	Number	No	Recommended	CVSS Base Score
CVSS Temporal Score	Number	No	No	CVSS Temporal Score
CVSS Environmental	Number	No	No	CVSS Environmental Score

Score				
Access Vector	Text	No	No	CVSS classification of how the vulnerability is exploited. Uses values of Undefined, Local, or Remote.
Access Complexity	Text	No	No	CVSS rating of the complexity of the attack required to exploit the vulnerability. Uses values of Undefinted, High, or Low.
Authentication	Text	No	No	CVSS rating of the number of times an attacker must authenticate to exploit a vulnerability. Uses values of Undefined, Required, or Not Required.
Confidentiality Impact	Text	No	No	CVSS rating of the impact the vulnerability has on confidentiality of the technology. Use values of Undefined, None, Partial, or Complete.
Integrity Impact	Text	No	No	CVSS rating of the impact the vulnerability has on integrity of the technology. Use values of Undefined, None, Partial, or Complete.
Availability Impact	Text	No	No	CVSS rating of the impact the vulnerability has on integrity of the technology. Use values of Undefined, None, Partial, or Complete.
Impact Bias	Text	No	No	CVSS weight for impact. Use

				values of Normal, Weight Confidentiality, Weight Integrity, or Weight Availability.
Collateral Damage Potential	Text	No	No	Potential for loss through damage or theft of the asset. Uses values of Undefined, None, Low, Medium, or High.
Target Distribution	Text	No	No	Proportion of vulnerable systems. Uses value None, Low, Medium, High, or Not Defined.
Exploitability	Text	No	No	CVSS current state of exploit techniques. Uses value Undefined, Unproven that Exploit Exists, Proof of Concept Code, Functional Exploit Exists, or High.
Remediation Level	Text	No	No	CVSS stage of the remediation lifecycle. Uses value Official Fix, Temporary Fix, Workaround, Unavailable, or Not Defined.
Report Confidence	Text	No	No	CVSS degree of confidence in the existence of the vulnerability. Uses value Unconfirmed, Uncorroborated, Confirmed, or Undefined.
Generated On	Date/Time	No	No	Date and time the CVSS score was generated

Table 18: Identified Vulnerabilities Table

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:

Identified Vulnerabilities Table

Name	Type	De-identified	Required	Description
Vulnerability ID	Text / Number	No	Yes	Reference to the Vulnerability in the Vulnerability Information Table
Technology ID	Text / Number	Yes	Yes	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	Method that the vulnerability was detected. Use values of Vulnerability Scanner Name or Manual Detection.
Scan Name	Text	No	No	If using Vulnerability Scanner, name of the scan.
Scan Date	Date / Time	No	No	If using Vulnerability Scanner, date the scan took place.
Vulnerability Status	Text	No	Yes	Current status of the vulnerability instance. Uses values of Open, Not Valid, or Mitigated. Vulnerabilities should be flagged Open by default.

Table 19: Vulnerability Remediation Table

This table represents information regarding the remediation of 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:

Identified Vulnerabilities Table

Name	Type	De-identified	Required	Description
Vulnerability ID	Text / Number	No	Yes	Unique identifier for the vulnerability instance.
Technology ID	Text / Number	No	Yes	Unique identifier for the technology.
Open Date	Date / Time	No	No	Date and time when the vulnerability was submitted for remediation.
Status	Text	No	No	Current status of the remediation effort. Use values of Open or Closed.
Priority	Text	No	No	How quickly vulnerability should be remediated. Use values of High, Medium, Low.
Close Date	Date / Time	No	No	Date and time when the vulnerability was remediated, if applicable.
Closed By	Text	No	No	Unique identifier for entity that remediated the vulnerability.

Table 20: Exempt Technologies Table

Displays technologies exempt from vulnerability management:

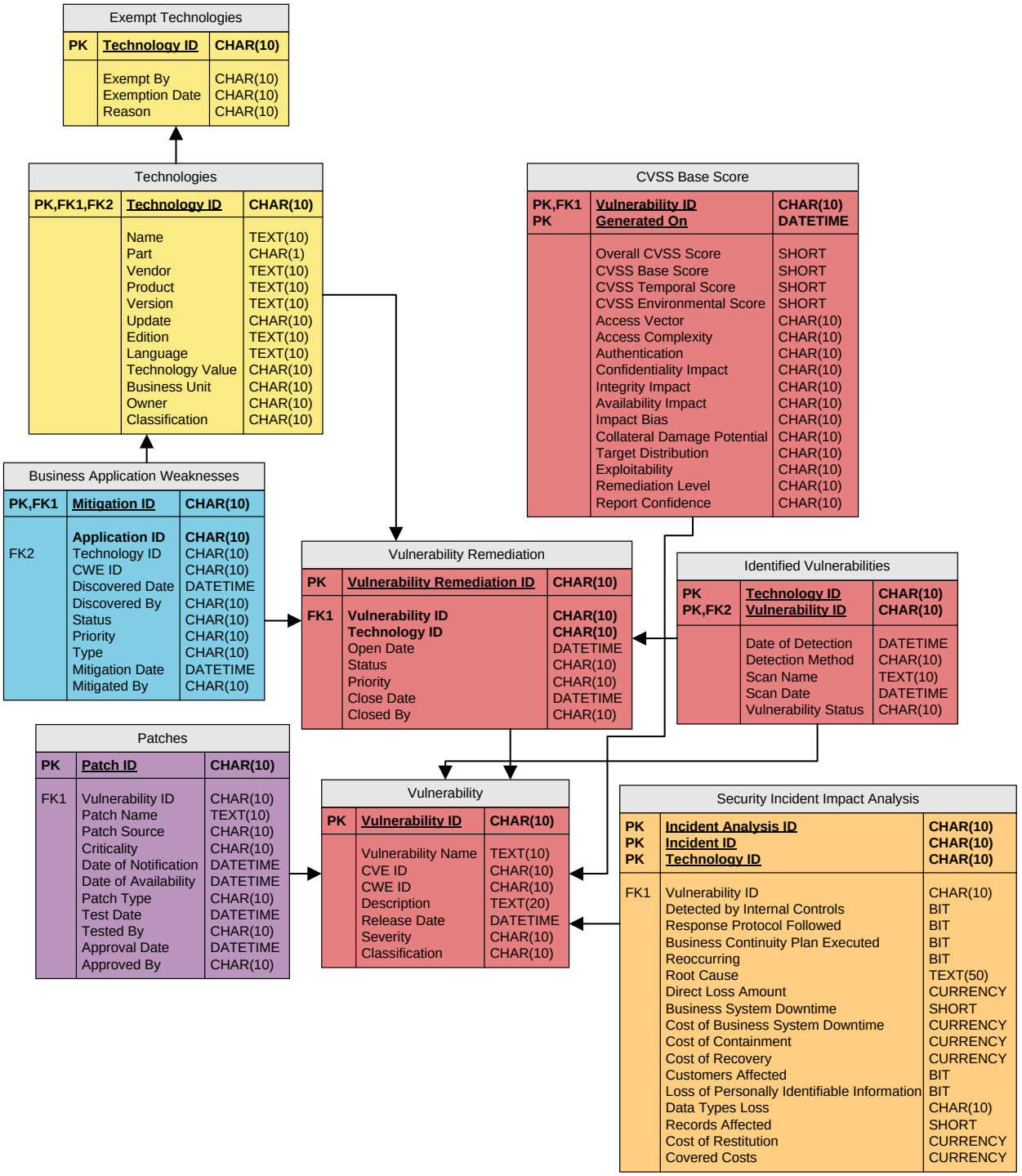
Technologies Table

Name	Type	De-identified	Required	Description
Technology ID	Text / Number	No	Yes	Unique identifier for the technology. Generally auto-

				generated.
Exempt By	Text	Yes	No	Unique identifier of the person who approved the exemption.
Exemption Date	Date / Time	No	No	Date and time the technology was exempt.
Reason	Text	No	No	Reason why technology was exempt

Diagram 2: Relational Diagram for Vulnerability Management Data Attributes

The diagram below shows the relationship of tables described in Vulnerability Management Data Attributes:



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.

Management and Operational Metrics

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 21: 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 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). 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
Type	Management
Audience	Business Management, Security Management

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. $$PSWKSV = \frac{Count(Systems\_Without\_Known\_Severe\_Vulnerabilities)}{Count(Scanned\_Systems)} * 100$$
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.
Sources	Vulnerability management systems will provide information on which systems were identified with severe vulnerabilities.
Visualization	Bar Chart X-axis: Time (Week, Month, Quarter, Year) Y-axis: PSWKSV (%)

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 22: 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. 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.
Type	Operational
Audience	Security Management
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: $MTTMV = \frac{\sum(Date\_of\_Mitigation - Date\_of\_Detection)}{Count(Mitigated\_Vulnerabilities)}$
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.
Sources	Vulnerability management systems will provide information on which systems were identified with severe vulnerabilities.
Visualization	Bar Chart X-axis: Time (Week, Month, Quarter, Year) Y-axis: PSWKSV (%)

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.

Mean Cost to Mitigate Vulnerabilities

Objective

This defines a metric for measuring the mean effort required to mitigate an identified vulnerability that can be remedied.

The metric is expressed in the context of a vulnerability management process, with the assumption that the organizations is scanning for known vulnerabilities, a formal system (i.e. change management and electronic ticketing system) is used to track activities to mitigate known vulnerabilities, and there is a known remedy for the vulnerability.

The metric is useful where a single vulnerability or remedy (no matter how many systems are affected) is expressed as a single change ticket or as one change ticket per affected network node.

Table 23: Mean Cost to Mitigate Vulnerabilities

Metric Name	Mean Cost to Mitigate Vulnerabilities
Version	1.0.0
Status	Final for Review
Description	The goal of this metric is to understand the effort required for vulnerability remediation activities. Risk management decisions can take into account the efficiency of vulnerability remediation and make more informed decisions around vulnerability policies, SLAs, and resource allocation in the IT environment.
Type	Operational
Audience	Security Management
Question	What is the average (mean) cost to the organization to mitigate an identified vulnerability during the given period?
Answer	A positive integer value that is greater than or equal to zero. A value of "0.0" indicates there were no measured costs to the organization.
Formula	This metric is calculated by summing the total cost to mitigate each vulnerability and dividing it by the total number of mitigated vulnerabilities. This count should also be done for each severity value (Low, Medium, and

	High): $$MCMV = \frac{\sum((Person\_Hours\_to\_Mitigate * Hourly\_Rate) + Other\_Mitigation\_Costs))}{Count(Mitigated\_Vulnerabilities)}$$
Units	$USD per Vulnerabilities
Frequency	Monthly
Targets	Ideally, all vulnerabilities would be remedied by a automated vulnerability remediation system, and the mean cost to remediate would be zero. In practice a target can be set based on the expected loss budget determined by risk assessments processes.
Sources	Vulnerability tracking systems will provide vulnerability data. Cost data can come from management estimates, ticket tracking systems, and capital and services budgets.
Visualization	Bar Chart X-axis: Time (Month) Y-axis: MCMV ($)

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 total cost and effort required to remediate and manage the known vulnerabilities in the organization.

Optimal conditions would reflect a low value in the metric. The lower the value the more efficient and cheaply the organization is able to mitigate identified vulnerabilities.

There may be a direct correlation between the number of un-mitigated vulnerabilities and the number of security incidents. Since vulnerabilities may not be addressed due to cost concerns,

an organization with a lower average remediation cost may be able to mitigate more vulnerabilities.

Limitations

Note that this assumes:

- Effort is tracked for vulnerability remediation
- Tickets are closed when the change is known to have mitigated the vulnerability
- Vulnerabilities can be tracked between scans on a vulnerability instance per-host basis
- We are not including in-progress tickets, vulnerabilities that have not been mitigated, or vulnerabilities that do not have a resolution.

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[8] 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.

8 http://oval.mitre.org/repository/index.html

Table 24: Technologies Table

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

Technologies Table

Name	Type	De-identified	Required	Description
Technology ID	Text / Number	No	Yes	Unique identifier for the technology. Generally auto-generated.
Name	Text	No	No	Name from CPE Dictionary which follows the following structure: cpe:/{PART}:{VENDOR}:{PRODUCT}:{VERSION}:{UPDATE}:{EDITION}:{LANGUAGE}.
Part	Text	No	No	Platform. Use value: H, O, or A. H, O, and A represent hardware, operating system, and application environment respectively.
Vendor	Text	No	No	Vendor from CPE Dictionary. This is the highest organization-specific label of the DNS name.
Product	Text	No	No	Product from CPE Dictionary. This is the most recognizable name of the product.
Version	Text	No	No	Version from CPE Dictionary. Same format as seen with the product.
Update	Text	No	No	Update or service pack information from CPE Dictionary.
Edition	Text	No	No	Edition from CPE Dictionary. May define specific target hardware and software architectures.
Language	Text	No	No	Language 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*.[9]
Business Unit	Text	No	No	Organizational business unit that the technology belongs to.
Owner	Text	No	No	Unique identifier for individual within the organization that is responsible for the technology.
Classification	Text	No	No	Classification of technology: Servers, Workstations, Laptops, Network Device, Storage Device, Applications, Operating systems

Table 25: Exempt Technologies Table

This table contains a list of technologies exempt from patch management.

Technologies Table

Name	Type	De-identified	Required	Description
Technology ID	Number	No	Yes	Unique identifier for the technology. Generally auto-generated.
Exemption Date	Date/Time	No	No	Date that the technology was exempt from patch management.
Exempt By	String	No	No	Unique identifier for entity granting exemption.
Reason	String	No	No	Reason for exemption.

Table 26: Patch Information Table

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

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

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
Patch Name	Text	No	No	The name of the patch.
Vulnerability ID	Number	No	Yes	One to many references to vulnerabilities in NVD addressed by this patch
Criticality Level	Text	No	No	Level of criticality as determined by the classification process, typically High, Medium, or Low.
Organization-Specific Criticality Level	Text	No	No	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	No	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 Type	String	No	No	Type of patch (service pack, application update, driver, etc.)

Table 27: Patch Activity Table

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.

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 ID	Number	No	Yes	Reference to the Patch in the Patch Information Table
Technology ID	Number	Yes	Yes	Number of instances of a specific technology. This is a count of all the technologies to which this patch applies.
Date of Installation	Date/Time	No	No	Date and time when the patch was installed (including any rebooting or reloading process).
Patch Status	Text	No	No	Current status of the patch. Use values Installed and Not Installed.
Priority	Text	No	No	Priority of patch installation. Use values of High, Medium, or Low.

Table 28: Patch Activity Review Table

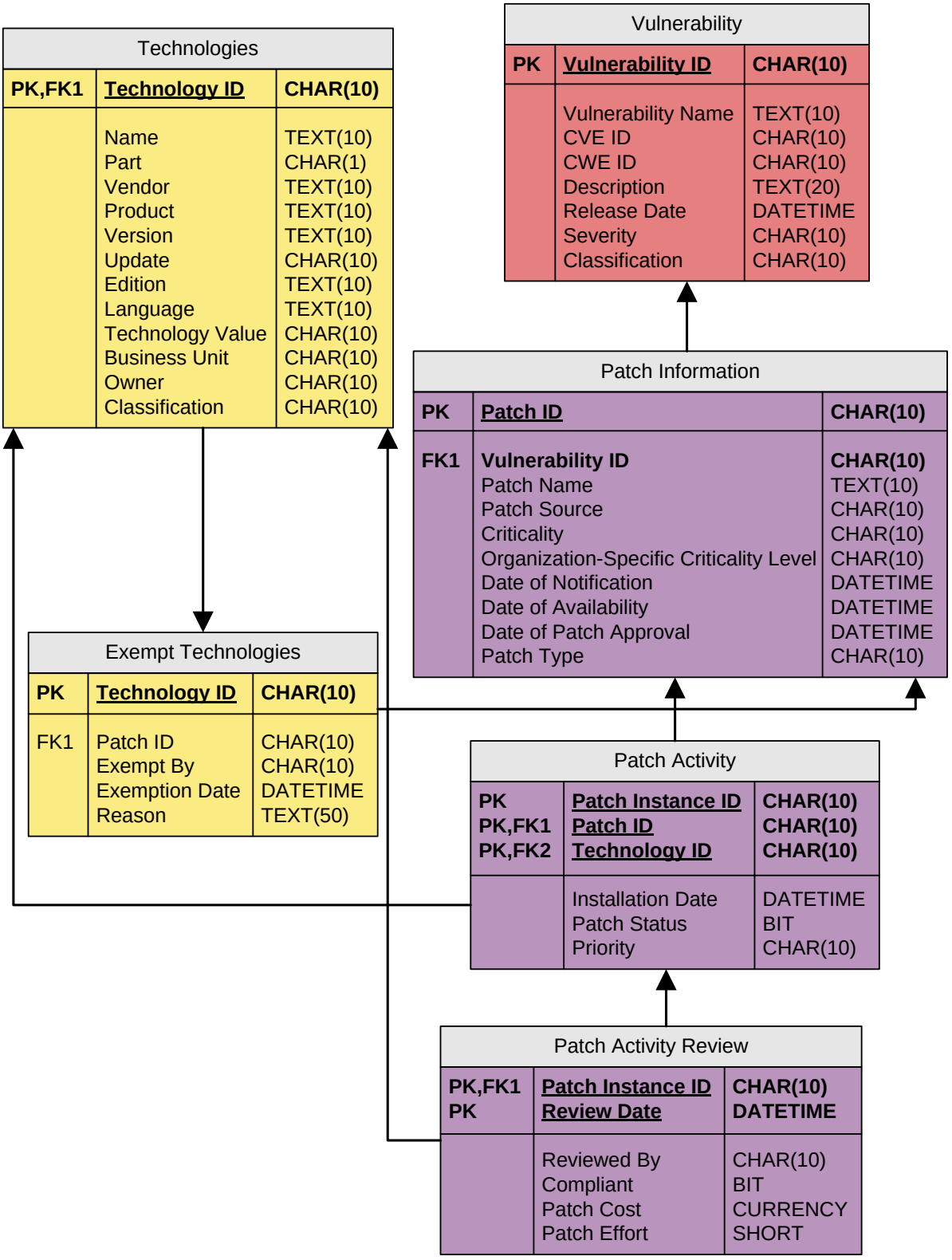
This table contains the verification that patches were installed properly and consistently throughout the organization.

Patch Activity Review Table

Name	Type	De-identified	Required	Description
Patch Instance ID	Number	No	Yes	Unique identifier for the patch instance. Generally auto-generated
Review Date	Date	No	No	Date review was conducted
Reviewed By	String	No	No	Entity that conducted the Review
Compliant	Boolean	No	No	Whether or not patch was installed in accordance to policy.
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.

Diagram 3: Relational Diagram for Patch Management Data Attributes

The diagram below shows the relationship of tables described in Patch Management Data Attributes:



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[11]:

- 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:

- 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).

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

- 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.

Management and Operational 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 29: 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. "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. "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.
Type	Management
Audience	Business Management, Security Management
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: $PPC = \frac{Count(Compliant\_Instances)}{Count(Technology\_Instances)} * 100$
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.
Sources	Patch management and IT support tracking systems will provide patch deployment data. Audit reports will provide compliance status.
Visualization	Bar Chart X-axis: Time (Week, Month, Quarter, Year) Y-axis: PPC (%)

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.

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 30: 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.
Type	Operational
Audience	Security Management
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:

	$$MTTP = \frac{\sum(Date\_of\_Installation - Date\_of\_Availability)}{Count(Completed\_Patches)}$$
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.
Sources	Patch management and IT support tracking systems will provide patch deployment data.
Visualization	Bar Chart X-axis: Time (Week, Month, Quarter, Year) Y-axis: MTTP (Hr/Patch)

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.

Mean Cost to Patch

Objective

This defines a metric for measuring the mean effort required to deploy a patch into an environment.

The metric is expressed in the context of a patch management process, with the assumption that the organizations has a formal system (i.e. patch management and electronic ticketing system) used to track activities to deploy patches.

The metric is useful where a single patch deployment (no matter how many systems are affected) is expressed as a single change ticket or as one change ticket per affected network node. This data can also be recorded as monthly totals where per-patch level granularity is not possible.

.

Table 31: Mean Cost to Patch

Metric Name	Mean Cost to Patch
Version	1.0.0
Status	Final
Description	The goal of this metric is to understand the effort required for patch management activities. Risk management decisions can take into account the efficiency of patch deployment to make more informed decisions around patch compliance policies, Service Level Agreements, and resource allocation in the IT environment.
Type	Operational
Audience	Security Management
Question	What is the average (mean) cost to the organization to deploy a patch during the given period?
Answer	A positive integer value that is greater than or equal to zero. A value of "0.0" indicates there were no measured costs to the organization.

Formula	Mean Cost to Patch is calculated by determining the total cost to deploy patches. These results are then averaged across the number of patches deployed in the current scope: $$MCP = \frac{\sum(Patch\_Cost + Other\_Patch\_Cost)}{Count(Deployed\_Patches)}$$ Patch Cost may be a determined aggregate cost or is calculated based upon the amount of effort put into the patching process as calculated by: Patch Effort * Hourly Rate. Other Patch Costs may include: • purchases of additional equipment • purchases of new software versions • cost associated with mitigation of a specific vulnerability • cost associated with vendor waiting to release patch until its next release cycle for identified vulnerabilities • cost associated with delaying updates until next update cycle • cost associated with identifying missing patches • cost associated with downtime during testing and installation of missing patches The cost of patch management systems should not be included in this cost. If a one-time cost is associated with multiple vulnerabilities the cost should be distributed evenly across the relevant vulnerabilities.
Units	$USD per Patch
Frequency	Monthly
Targets	Ideally, all patches would be deployed by an automated system, and the mean cost to patch would approach zero (given patch testing costs, etc.).
Sources	Patch management and IT support tracking systems will provide patch deployment data. Cost data can come from management estimates, ticket tracking systems, and services budgets.
Visualization	Bar Chart X-axis: Time (Month) Y-axis: MCP ($/Patch)

Usage

Keeping systems fully patched should reduce risk and result in lower incidents costs to the organization. Organizations generally have to balance the desire to patch systems with the cost and effort of preparing, testing, and deploying patches in their environment. Mean Cost to Patch allows the organization understand the cost of patch deployment, manage trends, and perform cost-benefit analysis patch updates, comparing the cost to the organization to the costs of any increases in security incidents.

Limitations

Note that this assumes:

- Effort is tracked for vulnerability remediation
- Tickets are closed when the change is known to have mitigated the vulnerability
- Vulnerabilities can be tracked between scans on a vulnerability instance per-host basis
- We are not including in-progress tickets, vulnerabilities that have not been mitigated, or vulnerabilities that do not have a resolution.

References

Cavusoglu, Cavusoglu, Zhang. "Economics of Security Patch Management." 2006.

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

Configuration Management Metrics

This section describes metrics for measuring security around configuration management in an organization's environment. Configuration management is important to organizations for both the deployment and ongoing management of systems.

The goal of Configuration Management is to provide control over the state of the IT infrastructure. Configuration management covers processes for the identification, recording, and reporting on the configuration state of the IT infrastructure. Some aspects of configuration management are also covered by other sets of security metrics, such as security patch management and vulnerability management.

Configuration management processes include: Identification of IT components, establishing control and authorized over changes, monitoring the status of configuration items, and verification and audit.

Key Questions in this business function are:

- What systems are in the organization?
- Are these systems configured as intended?
- What are the exceptions to intended configurations?

The initial metrics for this business function are:

- Configuration Compliance
- Configuration Assessment Coverage
- AV/AM Compliance

Data Attributes

Table 32: Technologies Table

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

Technologies Table				
Name	**Type**	**De-identified**	**Required**	**Description**

Technology ID	Text / Number	No	Yes	Unique identifier for the technology. Generally auto-generated.
Name	Text	No	No	Name from CPE Dictionary which follows the following structure: cpe:/{PART}:{VENDOR}:{PRODUCT}:{VERSION}:{UPDATE}:{EDITION}:{LANGUAGE}.
Part	Text	No	No	Platform. Use value: H, O, or A. H, O, and A represent hardware, operating system, and application environment respectively.
Vendor	Text	No	No	Vendor from CPE Dictionary. This is the highest organization-specific label of the DNS name.
Product	Text	No	No	Product from CPE Dictionary. This is the most recognizable name of the product.
Version	Text	No	No	Version from CPE Dictionary. Same format as seen with the product.
Update	Text	No	No	Update or service pack information from CPE Dictionary.
Edition	Text	No	No	Edition from CPE Dictionary. May define specific target hardware and software architectures.
Language	Text	No	No	Language 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*.[12]
Business Unit	Text	No	No	Organizational business unit that the technology belongs to.

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

Owner	Text	No	No	Unique identifier for individual within the organization that is responsible for the technology.
Classification	Text	No	No	Classification of technology: Servers, Workstations, Laptops, Network Device, Storage Device, Applications, Operating systems

Table 33: Configuration Status Accounting Table

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

Configuration Status Accounting Table

Name	Type	De-identified	Required	Description
Technology ID	Text / Number	No	Yes	Unique identifier for the technology. Generally auto-generated.
Baseline	Text	No	No	Description of baseline
Change ID	Text / Number	No	Yes	Unique identifier for the change, if applicable.

Table 34: Configuration Deviation Table

The following is a list of technologies with requests for deviation:

Configuration Deviation Table

Name	Type	De-identified	Required	Description
Technology ID	Text / Number	No	Yes	Unique identifier for the technology. Generally auto-generated.
Deviation ID	Text /	No	Yes	Unique identifier for deviation

	Number			request. Generally auto-generated.
Requested By	Text / Number	No	No	Unique identifier of entity submitting deviation request.
Request Date	Date / Time	No	No	Date and time deviation request was submitted.
Reason	Text	No	No	Reason for deviation.
Status	Text	No	No	Current status of deviation request. Use values Pending, Approved, or Not Approved.
Approval By	Text / Number	No	No	Unique identifier of entity approving/not approving deviation request.
Approval Date	Date / Time	No	No	Date and time deviation request was approved/not approved.

Table 35: Configuration Audit Table

The following is a list of technologies that have undergone configuration audit:

Configuration Deviation Table				
Name	**Type**	**De-identified**	**Required**	**Description**
Technology ID	Text / Number	No	Yes	Unique identifier for the technology. Generally auto-generated.
Audit ID	Text / Number	No	Yes	Unique identifier for configuration audit.
Audit Date	Date / Time	No	No	Date and time configuration audit occurred.
Audit By	Text /	No	No	Unique identifier for entity that

	Number			conducted the configuration audit.
Compliant	Boolean	No	No	Whether or not technology is compliant to configuration standards. Use values Compliant or Not Compliant.

Defined Metrics

Percentage of Configuration Compliance

Objective

The goal of this metric is to provide an indicator of the effectiveness of an organization's configuration management policy relative to information security, especially emerging exploits. If 100% of systems are configured to standard, then those systems are relatively more secure and manageable. If this metric is less than 100%, then those systems are relatively more exposed to exploits and to unknown threats.

Table 36: Percentage of Configuration Compliance

Metric Name	Percentage of Configuration Compliance
Version	1.0.0
Status	Final
Description	This document defines a metric for the effectiveness of configuration management in the context of information security. A percentage metric will allow benchmarking across organizations. This metric attempts to answer the question "Are system configuration compliance levels acceptable?" This question presumes the organization has defined an acceptable level of compliance, which may be less than 100% to account for the realities of ongoing change in the operational environments. The percentage of total computer systems in an organization that are configured in compliance with the organizations' approved standards. Compliance is a binary evaluation: a given system is either configured correctly according to the standard or it is not. Compliance can be evaluated

	by automated methods, manual inspection, audit, or some combination. The computer system population base is the total number of computer systems with approved configuration standards. This may be all systems or only a subset (i.e. only desktops, or only servers, etc.) The Configuration benchmark used is the CIS benchmarks if available (http://cisecurity.org). Additional metric results can be calculated for other or internal configuration benchmarks. Organizations that do not have approved standards for their computer systems should report "N/A" rather than a numeric value (0% or 100%) **In Scope** Examples of percentage of systems configured to approved standard could include: • Configuration of servers • Configuration of workstations/laptops • Configuration of hand-held devices • Configuration of other supported computer systems covered by the organizations patch policy **Out of Scope** Examples of computer system configurations that are not in scope include: • Temporary guest systems (contractors, vendors) • Lab/test systems performing to or in support of a specific non-production project • Networking systems (routers, switches, access points) • Storage systems (i.e. network accessible storage)
Type	Management
Audience	Business Management, Security Management
Question	What percentage of the organizations systems are in compliance with approved standards?
Answer	A positive integer value between zero and 100 inclusive, expressed as a

	percentage. A value of "100%" indicates that all technologies are in configuration management system scope.
Formula	Percentage of Configuration Compliance (PCC) is calculated by determining the total number of in-scope systems with approved configuration and then averaging this across the total number of in-scope systems: $PCC = \frac{\sum(In\_Scope\_Systems\_With\_Approved\_Configuration)}{Count(In\_Scope\_Systems)}$
Units	Percentage of Systems
Frequency	Monthly
Targets	The expected trend for this metric over time is to remain stable or increase towards 100%.
Sources	Configuration management and IT support tracking system audit reports will provide compliance status. Automated testing tools for CIS benchmarks are also available.
Visualization	Bar Chart X-axis: Time (Month) Y-axis: PCC (%)

Usage

The Percent of Configuration Compliance (PCC) represents the overall compliance to configuration policies. The higher the PCC the more consistent the organization's systems are and the easier is it to establish and maintain those systems.

Limitations

- This metric relies on the organization being able to identify all systems that are under configuration management. Some systems may be exempt from configuration management policies.
- This metric relies on the organization being able to verify that the system is in compliance to configuration policies

References

Center for Internet Security, Benchmarks (http://cisecurity.org)

IEEE Standard 828-1990, Software Configuration Management Plans.

ISO/IEC 12207:2008, Information technology — Software life cycle processes and ISO/IEC 15288: 2008, Information technology — System life cycle processes.

Ross, Katzke, Johnson, Swanson, Stoneburner and Rogers. Special Publication SP 800-53: Recommended Security Controls for Federal Information Systems (Rev 2). US National Institute of Standards and Technology, 2007

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

Change Management Metrics

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

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 change data record. These attributes were derived from the *ITIL v3 –Request for Change* data record.[13] Please note that some fields in the Request for Change record are documented here because they are not needed for configuration metrics calculations.

Table 37: Technologies Table

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

[13] 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>

Technologies Table

Name	Type	De-identified	Required	Description
Technology ID	Text / Number	No	Yes	Unique identifier for the technology. Generally auto-generated.
Name	Text	No	No	Name from CPE Dictionary which follows the following structure: cpe:/{PART}:{VENDOR}:{PRODUCT}:{VERSION}:{UPDATE}:{EDITION}:{LANGUAGE}.
Part	Text	No	No	Platform. Use value: H, O, or A. H, O, and A represent hardware, operating system, and application environment respectively.
Vendor	Text	No	No	Vendor from CPE Dictionary. This is the highest organization-specific label of the DNS name.
Product	Text	No	No	Product from CPE Dictionary. This is the most recognizable name of the product.
Version	Text	No	No	Version from CPE Dictionary. Same format as seen with the product.
Update	Text	No	No	Update or service pack information from CPE Dictionary.
Edition	Text	No	No	Edition from CPE Dictionary. May define specific target hardware and software architectures.
Language	Text	No	No	Language 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*.[14]
Business Unit	Text	No	No	Organizational business unit that the technology belongs to.
Owner	Text	No	No	Unique identifier for individual within the organization that is responsible for the technology.
Classification	Text	No	No	Classification of technology: Servers, Workstations, Laptops, Network Device, Storage Device, Applications, Operating systems

Table 38: Change Exemption Table

This table displays technologies that are exempt from changes.

Change Exemption Table

Name	Type	De-identified	Required	Description
Change Request ID	Number	No	Yes	Unique identifier for the change request. Generally auto-generated.
Technology ID	Text/Number	No	Yes	One-to-many reference to technologies that should undergo change.
Exempt By	Text	Yes	No	Unique identifier of the person who approved the exemption
Exemption Date	Date/Time	No	No	Date and time the technology was exempt
Reason	Text	No	No	Reason why technology was exempt

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

Table 39: Change Request Table

Change Request contains information regarding the approval of change requests.

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
Requested By	Text	Yes	No	Unique identifier of the person that submitted the change
Priority	Text	No	JNo	How soon the request should take place. Uses values *High*, *Medium*, and *Low*.
Change Type	Text	No	No	Type of change. Use values Architectural, Component, or Emergency.
Estimated Cost	Text	No	No	Estimated cost of the change in Level of Effort or actual dollar amounts
Status	Text	No	No	Current status of the request. Use values Approved, Not Approved, or Pending Approval.
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
Technology ID	Text/Number	No	No	One-to-many reference to technologies that should undergo configuration change.

Table 40: Change Item Table

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

Change Item Table

Name	Type	De-identified	Required	Description
Change ID	Number	No	Yes	Unique identifier for the 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 change.
Technology ID	Text/Number	No	No	One-to-many reference to the technologies that underwent configuration change.
Scheduled Start Date	Date/Time	No	No	Suggested date and time for the change
Start Date	Date/Time	No	No	Date and time change started. May use Approval Date.
Completion Date	Date/Time	No	No	Date and time the change was completed.

Table 41: Configuration Change Review Table

This table displays change requests that were reviewed following a change.

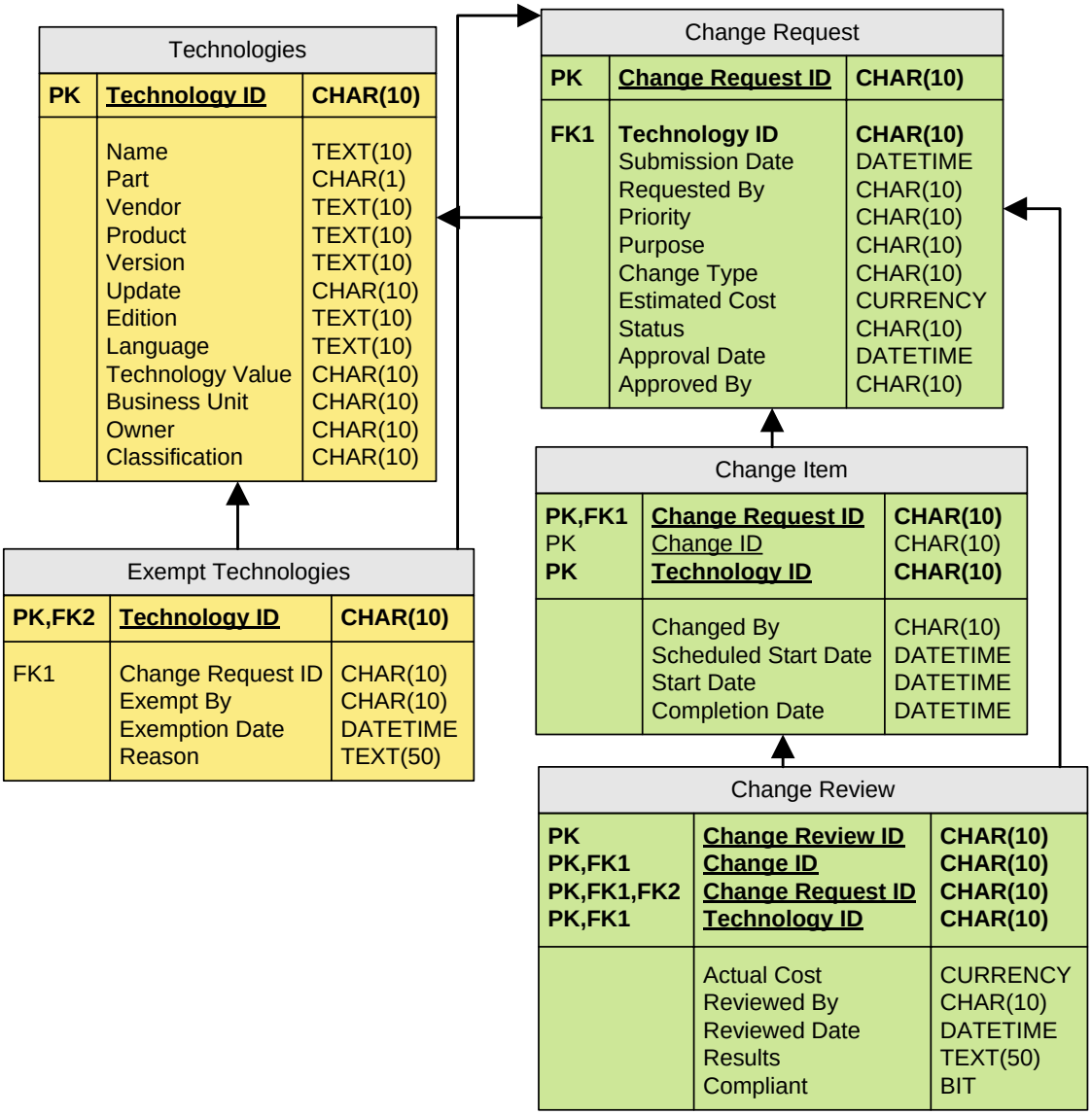
Change Review Table

Name	Type	De-identified	Required	Description
Change ID	Number	No	Yes	Unique identifier for the change. Generally auto-generated.
Change	Number	No	Yes	Unique identifier for the change

Request ID				request. Generally auto-generated.
Technology ID	Text/Number	No	No	One-to-many reference to technologies that should undergo configuration change.
Actual Cost	Text	No	No	Actual cost of the change in Level of Effort or actual dollar amounts
Change Review ID	Text/Number	No	Yes	Unique identifier for the change review.
Reviewed By	Text	Yes	No	Unique identifier of the person who reviewed the change.
Reviewed Date	Date/Time	No	No	Date and time the change was reviewed.
Results	Text	No	No	Results of the change review. Use values of In Compliance, Not in Compliance.
Compliant	Boolean	No	No	Whether or not change was completed in accordance to policy. Use values Compliant or Not Compliant.

Diagram 5: Relational Diagram for Configuration Management Data Attributes

The diagram below shows the relationship of tables described in Configuration Management Data Attributes:



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, and 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 42: 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.
Type	Operational
Audience	Security Management
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: $$MTCC = \frac{Sum(Completion\_Date - Submission\_Date)}{Count(Completed\_Changes)}$$
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.
Sources	Configuration management and IT support tracking systems will provide configuration change data.
Visualization	Bar Chart X-axis: Time (Week, Month, Quarter, Year) Y-axis: MTCC (Days/Request)

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 43: 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.
Type	Management
Audience	Business Management, Security Management
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. $$PCSR = \frac{Count(Completed\_Changes\_with\_Security\_Reviews)}{Count(Completed\_Changes)} * 100$$
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.
Sources	Configuration management and IT support tracking systems will provide configuration change data.
Visualization	Bar Chart X-axis: Time (Week, Month, Quarter, Year) Y-axis: PCSR (%)

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 44: 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.
Type	Operational
Audience	Security Management
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: $$PCSE = \frac{Count(Completed\_Changes\_with\_Security\_Exceptions)}{Count(Completed\_Changes)} * 100$$
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.
Sources	Configuration management and IT support tracking systems will provide configuration change data.
Visualization	Bar Chart X-axis: Time (Week, Month, Quarter, Year) Y-axis: PCSE (%)

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.

Table 45: Technologies Table

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

Technologies Table

Name	Type	De-identified	Required	Description
Technology ID	Text / Number	No	Yes	Unique identifier for the technology. Generally auto-generated.
Name	Text	No	No	Name from CPE Dictionary which follows the following structure: cpe:/{PART}:{VENDOR}:{PRODUCT}:{VERSION}:{UPDATE}:{EDITION}:{LANGUAGE}.
Part	Text	No	No	Platform. Use value: H, O, or A. H, O, and A represent hardware, operating system, and application environment respectively.
Vendor	Text	No	No	Vendor from CPE Dictionary. This is the highest organization-specific label of the DNS name.
Product	Text	No	No	Product from CPE Dictionary. This is the most recognizable name of the product.
Version	Text	No	No	Version from CPE Dictionary. Same format as seen with the product.
Update	Text	No	No	Update or service pack information from CPE Dictionary.
Edition	Text	No	No	Edition from CPE Dictionary. May define specific target

				hardware and software architectures.
Language	Text	No	No	Language 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*.[16]
Business Unit	Text	No	No	Organizational business unit that the technology belongs to.
Owner	Text	No	No	Unique identifier for individual within the organization that is responsible for the technology.
Classification	Text	No	No	Classification of technology: Servers, Workstations, Laptops, Network Device, Storage Device, Applications, Operating systems

Table 46: Business Applications Table

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

Business Applications Table

Name	Type	De-identified	Required	Description
Application ID	Number	No	Yes	Unique identifier for the application. Generally auto-generated.
Application Name	Text	Yes	No	The name of the business application from CPE Dictionary. This is the most recognizable name of the product.
Version	Text	No	No	Version from CPE Dictionary. Same format as seen with the product.

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

Vendor	Text	No	No	Vendor from CPE Dictionary. This is the highest organization-specific label of the DNS name.
Language	Text	No	No	Language from CPE Dictionary. Use value C/C++, JSP, ASP, .NET, J2EE, CGI, Perl, PHP, Web Services, or Other.
Web-Enabled	Boolean	No	No	Whether or not the application is web-enabled.
In-House Development	Number	No	No	Percentage of application developed in-house, if applicable.
Vendor Development	Number	No	No	Percentage of application developed by vendor, if applicable.
Custom	Number	No	No	Percentage of application that was customized, if applicable.
Database	Text	No	No	Primary database used. Use Oracle, MySQL, SQLServer, or Other.
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.
Owner	Text	Yes	No	Unique identifier for individual responsible for the application.
Hosting	Boolean	No	No	Whether application is managed internally or externally. Use values Internal or External.

Table 47: Business Application Status Table

Current status of all business applications within the organization:

Business Application Status Table				
Name	**Type**	**De-identified**	**Required**	**Description**
Application ID	Number	No	Yes	Unique identifier for the application. Generally auto-generated.
Technology ID	Text / Number	No	Yes	Unique identifier for the technology. Generally auto-generated.
Application Status	Text	No	No	Indicator of the application's current status. Uses values In Testing, In Development, or Production.
Status Changed Date	Date / Time	No	No	Date and time when application status was last changed.
Status Changed By	Text	No	No	Unique identifier for entity that updated the application status.

Table 48: Risk Assessments Table

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.

Risk Assessments Table				
Name	**Type**	**De-identified**	**Required**	**Description**
Assessment ID	Text / Number	No	Yes	Unique identifier for the assessment. Generally auto-generated.
Technology ID	Text / Number	No	Yes	Unique identifier for the technology the application resides on.

Date of Assessment	Date / Time	No	No	Date that risk assessment was completed.
Application ID	Text / Number	Yes	Yes	Unique identifier for the application.
Assessed By	Text	No	No	Unique identifier of the entity conducting the assessment.
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	Number	No	No	Total person-hours of the assessment
Assessment Cost	Number	No	No	Total cost of the assessment
Assessment Scope	Text	No	No	Scope of the risk assessment covering this application: *Organization*, *system*, or *application*
Compliant	Boolean	No	No	Whether or not the application is compliant with security control standards. Use values Compliant or Not Compliant.
Assessment Results	Text	No	No	Results of the assessment.

Table 49: Security Testing Table

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.

Security Testing Table				
Column Name	**Type**	**De-identified**	**Required**	**Column Description**

Testing ID	Text / Number	No	Yes	Unique identifier for the test. Generally auto-generated.
Technology ID	Text / Number	No	Yes	Unique identifier for the technology the application resides on.
Date of Testing	Date / Time	No	No	Date that security testing was performed.
Application ID	Text / Number	Yes	Yes	Reference identifier for the application.
Tested By	Text	No	No	Unique identifier of the entity conducting the testing.
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	Whether or not security test was automated. Use values 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).

Table 50: Business Application Weaknesses Table

Current mitigation status of weaknesses discovered on business applications.

Business Application Weaknesses Table				
Column Name	**Type**	**De-identified**	**Required**	**Column Description**
Mitigation ID	Text / Number	No	Yes	Unique identifier for the

				mitigation ticket. Generally auto-generated.
Application ID	Text / Number	No	Yes	Unique identifier for the application. Generally auto-generated.
Technology ID	Text / Number	No	Yes	Unique identifier for the technology the application resides on.
CWE ID	Text / Number	No	Yes	Unique identifier for the weakness.
Discovered Date	Date/Time	No	No	Date and time the weakness was discovered. May be the same date the risk assessment or security testing was performed.
Discovered By	Text	No	No	Unique identifier of the entity that discovered the weakness. May be Security Testing ID or Risk Assessment ID.
Status	Text	No	No	Current status of the mitigation effort. Use values of Mitigated or Not Mitigated.
Priority	Text	No	No	How quickly weakness should be mitigated. Use values of High, Medium, Low.
Type	Text	No	No	Type of weakness.
Mitigation Date	Date/Time	No	No	Date and time when the weakness was mitigated, if applicable.
Mitigated By	Text	No	No	Unique identifier for entity that mitigated the weakness, if applicable.

Table 51: Most Dangerous Programming Errors Table

CWE/SANS Top 25 Most Dangerous Programming Errors is a list of the most widespread and critical programming errors that can lead to serious software vulnerabilities.

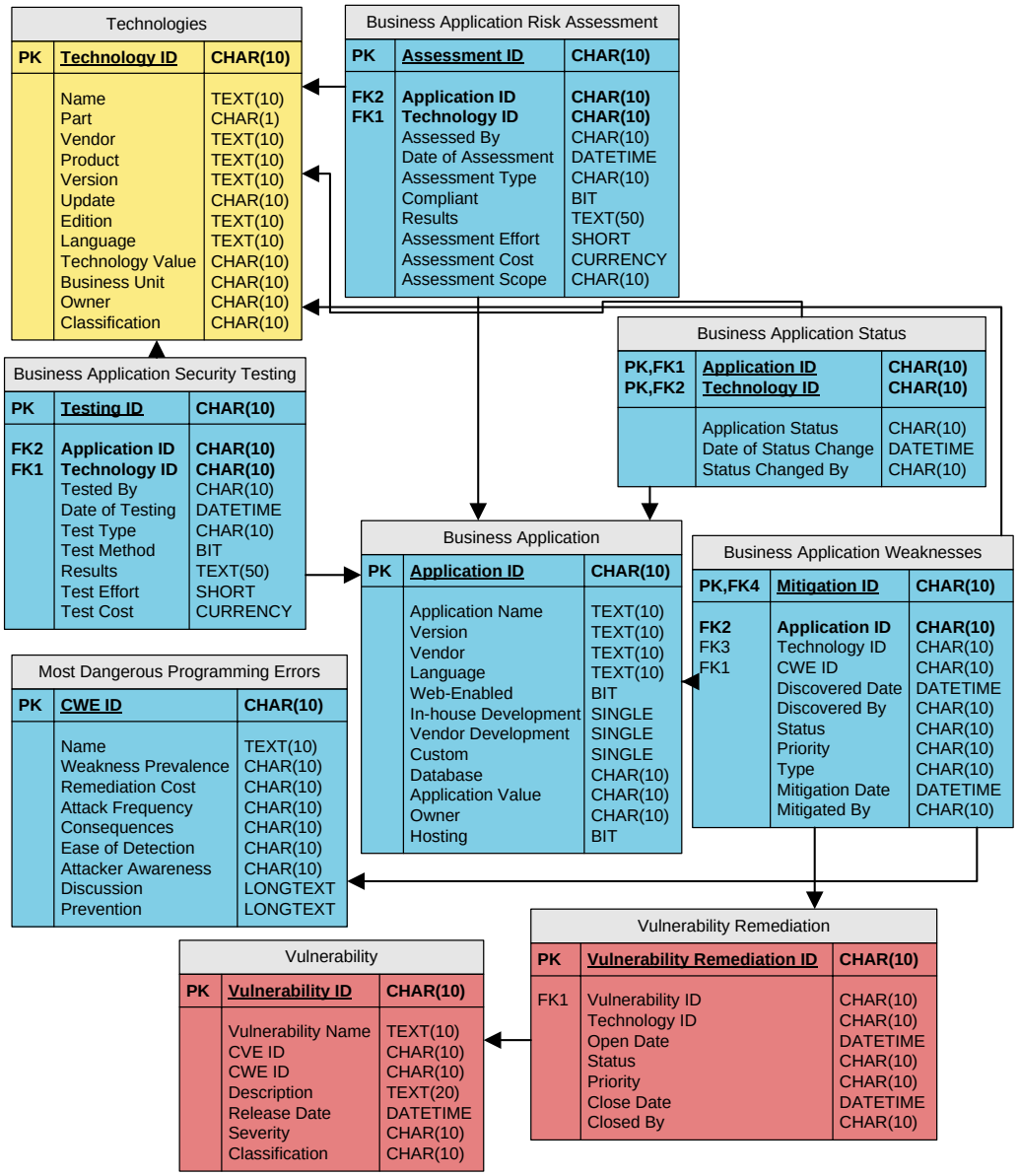
Most Dangerous Programming Errors Table

Column Name	Type	De-identified	Required	Column Description
CWE ID	Text / Number	No	Yes	Unique identifier for the weakness.
Name	Text	No	No	Name of the weakness.
Weakness Prevalence	Text	No	No	How often the weakness is encountered. Use values of Limited, Medium, Common, High, or Widespread.
Remediation Cost	Text	No	No	The amount of effort required to fix the weakness. Use values of Low, Medium, or High.
Attack Frequency	Text	No	No	How often the weakness occurs in vulnerabilities that are exploited by an attacker. Use values of Rarely, Sometimes, or Often.
Consequences	Text	No	No	Impact on the organization should the weakness be exploited. Use values of Code Execution, Security Bypass, Data Loss, Code Execution, or Denial of Service.
Ease of Detection	Text	No	No	How easy it is for an attacker to find this weakness. Use values of Easy, Moderate, or Difficult.
Attacker Awareness	Text	No	No	The likelihood that an attacker is going to be aware of this particular weakness, methods for detection, and

				methods for exploitation. Use values of Medium, High.
Discussion	Text	No	No	Discussion of the nature of the weakness and its consequences.
Prevention	Text	No	No	Steps to mitigate or eliminate the weakness.

Diagram 6: Relational Diagram for Application Management Data Attributes

The diagram below shows the relationship of tables described in Application Management Data Attributes:



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

Percentage of Critical Applications

Objective

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

Table 52: 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.
Type	Technical
Audience	Security 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: $$PCA = \frac{Count(Critical\_Applications)}{Count(Applications)} * 100$$
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 53: 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.
Type	Technical
Audience	Security 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: $$RAC = \frac{Count(Applications\_Undergone\_Risk\_Assessment)}{Count(Applications)} * 100$$
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 54: 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 performed on systems post-deployment (although they could be in pre-production testing). 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.
Type	Technical
Audience	Security 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: $$STC = \frac{Count(Applications\_Undergone\_Security\_Testing)}{Count(Deployed\_Applications)} * 100$$
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 55: 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, and 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 56: 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.
Type	Management
Audience	Business 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. $SBPITB = \frac{SecurityBudget}{ITBudget}$
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 divisions, 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 57: 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.
Type	Management
Audience	Business Management, Security 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

Technical Metrics

Incidents

Number of Incidents

Objective

Number of Incidents 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 58: Number of Incidents

Metric Name	Number of Incidents
Version	1.0.0
Status	Final
Description	Number of Incidents measures the number of security incidents for a given time period.
Type	Technical
Audience	Security 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 Number of Incidents (NI), the number of security incidents are counted across a scope of incidents, for example a given time period, category or business unit: NI = Count(Incidents)
Units	Incidents per period; for example, incidents per week or incidents per month
Frequency	Weekly, Monthly, Quarterly, Annually

Targets	NI 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.
Visualization	Column Chart X-axis: Time (Week, Month, Quarter, Year) Y-axis: NI (Incidents)

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 Number of Incidents 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 locations.

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>

Vulnerability Management

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 59: 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.
Type	Technical
Audience	Security 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: $$VSC = \frac{Count(Scanned\_Systems)}{Count(All\_Systems\_Within\_Organization)} * 100$$

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.
Sources	Vulnerability management and asset management systems will provide information on which systems are scanned for vulnerabilities.
Visualization	Bar Chart X-axis: Time (Week, Month, Quarter, Year) Y-axis: VSC (%)

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.

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 60: 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 that have been found on organization's systems during the vulnerability identification process.
Type	Technical
Audience	Security 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): *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.
Sources	Vulnerability management systems will provide information on which systems were identified with severe vulnerabilities.
Visualization	Bar Chart X-axis: Time (Week, Month, Quarter, Year) Y-axis: NKVI (Number of Vulnerabilities)

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

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 61: 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.
Type	Technical
Audience	Security 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. $PMC = \frac{Count(Technology\_Instances\_Under\_Patch\_Management)}{Count(Technology\_Instances)} * 100$
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.

Sources	Patch management and IT support tracking systems will provide patch deployment data.
Visualization	Bar Chart X-axis: Time (Week, Month, Quarter, Year) Y-axis: PMC (%)

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.

Configuration Management

Configuration Management Coverage

Objective

The goal of this metric is to provide an indicator of the scope of configuration management control systems and monitoring.

Accurate and timely detection of configuration changes, as well as the ability to assess the state of the current configuration through regular processes or automated means provides organizations with improved visibility into their security posture.

If 100% of systems are under configuration monitoring than the organization is relatively less exposed to exploits and to unknown threats resulting from un-approved, untested, or unknown configuration states.

Table 62: Configuration Management Coverage

Metric Name	Configuration Management Coverage
Version	1.0.0
Status	Final
Description	This metric attempts to answer the question "Are system under configuration management control?" This question presumes the organization has a configuration management system to test and monitor the configuration states of systems. The percentage of total computer systems in an organization that are under the scope of a configuration monitoring/management system. Scope of configuration monitoring is a binary evaluation: a given system is either part of a system that can assess and report it's configuration state or it is not. Configuration state can be evaluated by automated methods, manual inspection, or audit, or some combination. The computer system population base is the total number of computer systems with approved configuration standards. This may be all systems or only a subset (i.e. only desktops, or only servers, etc.) Organizations that do not have approved standards for their computer systems should report "N/A" rather than a numeric value (0% or 100%). **In Scope** Examples of percentage of systems under configuration management may include: • Configuration of servers

	• Configuration of workstations/laptops • Configuration of hand-held devices • Configuration of other supported computer systems covered by the organizations configuration policy **Out of Scope** Examples of computer system configurations that are not in scope include: • Temporary guest systems (contractors, vendors) • Lab/test systems performing to or in support of a specific non-production project • Networking systems (routers, switches, access points) • Storage systems (i.e. network accessible storage)
Type	Technical
Audience	Security Operations
Question	What percentage of the organizations systems are under configuration management?
Answer	A positive integer value between zero and 100 inclusive, expressed as a percentage. A value of "100%" indicates that all technologies are in configuration management system scope.
Formula	Configuration Management Coverage (CMC) is calculated by determining the number of in-scope systems within configuration management scope and then averaging this across the total number of in-scope systems: $$CMC = \frac{\sum(In\_Scope\_Systems\_Under\_Configuration\_Management)}{Count(In\_Scope\_Systems)}$$
Units	Percentage of Systems
Frequency	Monthly
Targets	The expected trend for this metric over time is to remain stable or increase towards 100%.
Sources	Configuration management and asset management systems will provide

	coverage.
Visualization	Bar Chart X-axis: Time (Month) Y-axis: CMC (%)

Usage

The Configuration Management Coverage metric provides information about well the organization ensures the integrity of their network. Organizations can use this metric to evaluate their risk position in terms of concentrations of inconsistent state of systems.

The results of the coverage metric indicate the:

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

Limitations

The organization's critical systems (e.g. production servers) maybe out of scope of the configuration management system by design, for performance or network architecture reasons.

References

Ross, Katzke, Johnson, Swanson, Stoneburner and Rogers. Special Publication SP 800-53: Recommended Security Controls for Federal Information Systems (Rev 2). US National Institute of Standards and Technology, 2007

IEEE Standard 828-1990, Software Configuration Management Plans.

ISO/IEC 12207:2008, Information technology — Software life cycle processes and ISO/IEC 15288: 2008, Information technology — System life cycle processes.

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

Current Anti-Malware Coverage

Objective

The goal of this metric is to provide an indicator of the effectiveness of an organization's anti-malware management. If 100% of systems have current anti-malware detection engines and signatures, then those systems are relatively more secure. If this metric is less than 100%, then those systems are relatively more exposed to viruses and other malware.

The expected trend for this metric over time is to remain stable or increase towards 100%.

Table 63: Current Anti-Malware Coverage

Metric Name	Current Anti-Malware Coverage
Version	1.0.0
Status	Final
Description	This metric attempts to answer the question "Do we have acceptable levels of anti-malware coverage?" This question presumes the organization has defined what is an acceptable level of compliance, which may be less than 100% to account for ongoing changes in the operational environments. Malware includes computer viruses, worms, trojan horses, most rootkits, spyware, dishonest adware, crimeware and other malicious and unwanted software [http://en.wikipedia.org/wiki/Malware]. The percentage of total computer systems in an organization that have current, up-to-date anti-virus (or anti-malware) software and definition files. "Current" is a binary evaluation: a given system is either configured with both up-to-date detection engines and signatures or it is not. Compliance can be evaluated by automated methods, manual inspection, audit, or some combination. Current coverage of a system is defined as a the most recent version of the engine, and a signature file that is no more than 14 days older than the most recent signature file released. <u>In Scope</u> Examples of systems under considerations for this metric include:

	• Servers • Workstations/laptops • Hand-held devices • Other supported computer systems <u>Out of Scope</u> Examples of systems that are not under consideration for this metric include: • Temporary guest systems (contractors, vendors) • Lab/test systems performing to or in support of a specific non-production project • Networking systems (routers, switches, access points) • Storage systems (i.e. network accessible storage)
Type	Technical
Audience	Security Operations
Question	What percentage of the organizations systems have current anti-malware protection?
Answer	A positive integer value between zero and 100 inclusive, expressed as a percentage. A value of "100%" indicates that all technologies have current anti-malware coverage.
Formula	Current Anti-Malware Coverage (CAMC) is calculated by determining the number of in-scope systems with current coverage and then averaging this across the total number of in-scope systems: $$CMC = \frac{\sum(In\_Scope\_Systems\_with\_current\_Anti-Malware)}{Count(In\_Scope\_Systems)}$$
Units	Percentage of Systems
Frequency	Monthly
Targets	The expected trend for this metric over time is to remain stable or increase towards 100%.
Sources	Configuration management and Anti-malware systems (locally or centrally managed).

Visualization	Bar Chart X-axis: Time (Month) Y-axis: CAMC (%)

Usage

Current Anti-Malware Coverage(CAMC) represents the overall compliance to anti-malware policies. The higher the CAMC the greater the number of systems in the organization are running anti-malware with recent signature files, the less likely it is that existing known malware will infect or spread across the organizations systems, or fail to be detected in a timely manner.

Limitations

- Systems critical to the organization (e.g. production servers) maybe out of scope of the anti-malware management system by design, for performance, or network architecture reasons.
- Variation in type of anti-malware such as inbound email scanning vs. resident process scanning may be material. The completeness of signature files and frequency of updates may also vary.
- The time window defined as current may not be adequate if malware has its impact on the organization before signature files are developed, or before the current window has expired.

References

Ross, Katzke, Johnson, Swanson, Stoneburner and Rogers. Special Publication SP 800-53: Recommended Security Controls for Federal Information Systems (Rev 2). US National Institute of Standards and Technology, 2007

Application Security

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 64: 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.
Type	Technical
Audience	Security 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: $NOA = Count(Applications)$
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/

Appendix A: Glossary

Anti-malware

Anti-malware is security software that detects, blocks, and neutralizes malware of various types (see *Malware*).

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.[18] *Systemic Bias* is identified as the inherent tendency of a process to favor a particular outcome.[19]

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.

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

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

Containment

Containment is identified as limiting the extent of an attack.[20] 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.

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[21], 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.

Malware

Malware includes computer viruses, worms, trojan horses, most rootkits, spyware, dishonest adware, crimeware and other malicious and unwanted software [http://en.wikipedia.org/wiki/Malware].

---

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

[21] Also known as a "timestamp."

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.[22] 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.

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

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.

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

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 need. 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 65: 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.

Type	
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: $$PIDIC = \frac{Count(Incident\_DetectedByInternalControls = TRUE)}{Count(Incidents)} * 100$$
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 66: 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: $$MTDC = \frac{\sum(Date\_of\_Containment - Date\_of\_Discovery)}{Count(Incidents)}$$
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 67: 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. 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: $$MTDCP = \frac{\sum(Date\_of\_Installation - Date\_of\_Notification)}{Count(Completed\_Critical\_Patches)}$$
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 blackmark 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