Threat Assessment and Remediation Analysis (TARA) Overview

October 2013

Approved for Public Release: 11-4987.

Distribution Unlimited.

MITRE

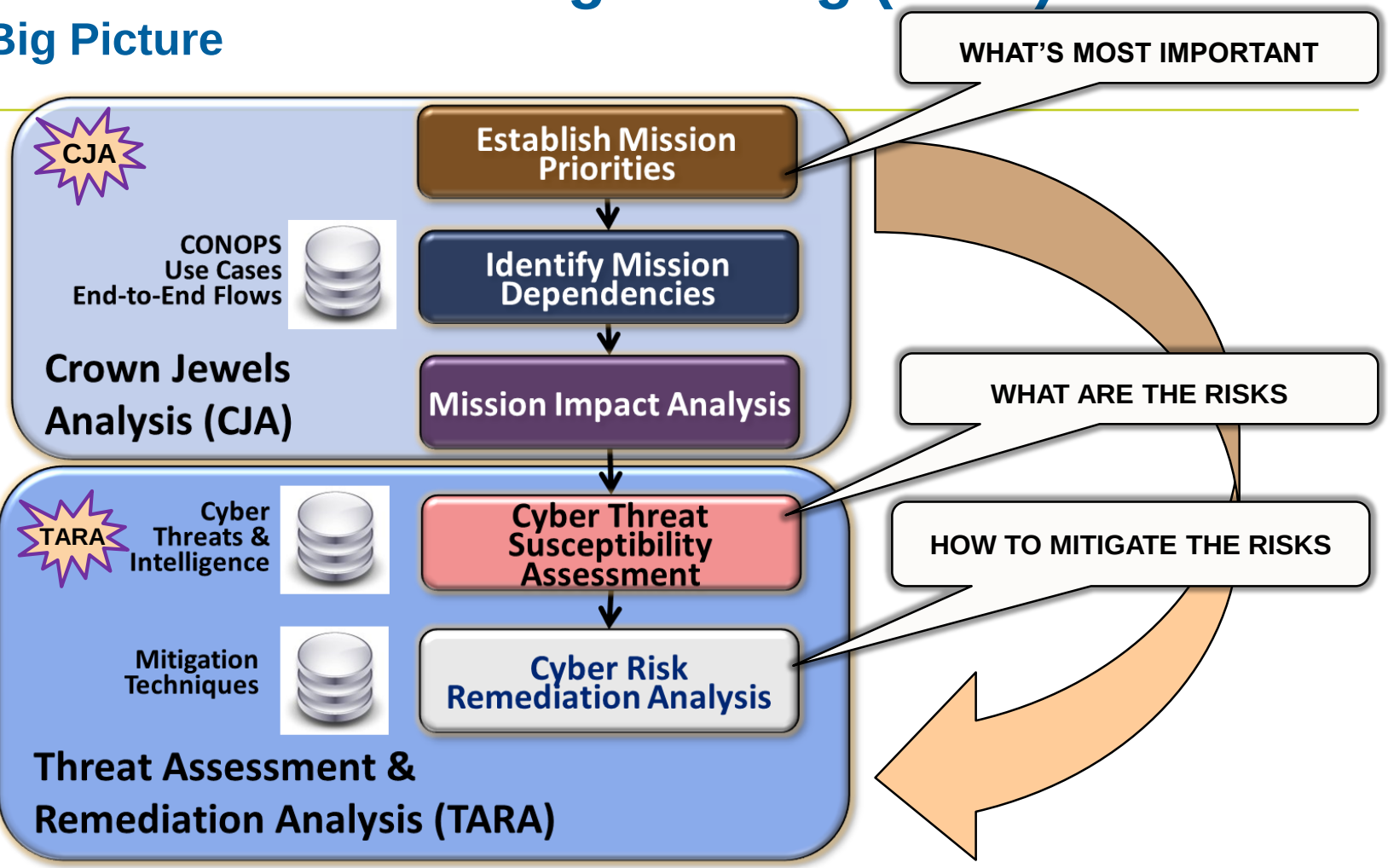
Threat Assessment & Remediation Analysis (TARA)

- **Methodology to identify and assess cyber threats and select countermeasures effective at mitigating those threats**
  - Leverages catalog of Attack Vectors (AVs), Countermeasures (CMs), and associated mappings
    - Use of catalog ensures that findings are consistent across assessments
  - Uses scoring models to quantitatively assess AVs and CMs
    - AVs ranked by risk, providing a basis for effective triage
    - CMs ranked by cost-effectiveness, providing a basis for identifying optimal solutions
  - Delivers recommendations
    - Allows programs to make informed choices on how best to improve a system's security posture and resilience
  - Can be performed standalone or as follow-on to criticality or mission impact (MI) analysis, such as Crown Jewels Analysis (CJA)
    - TARA performed in tandem with CJA supports Mission Assurance Engineering (MAE) objectives

MITRE

Mission Assurance Engineering (MAE)

The Big Picture

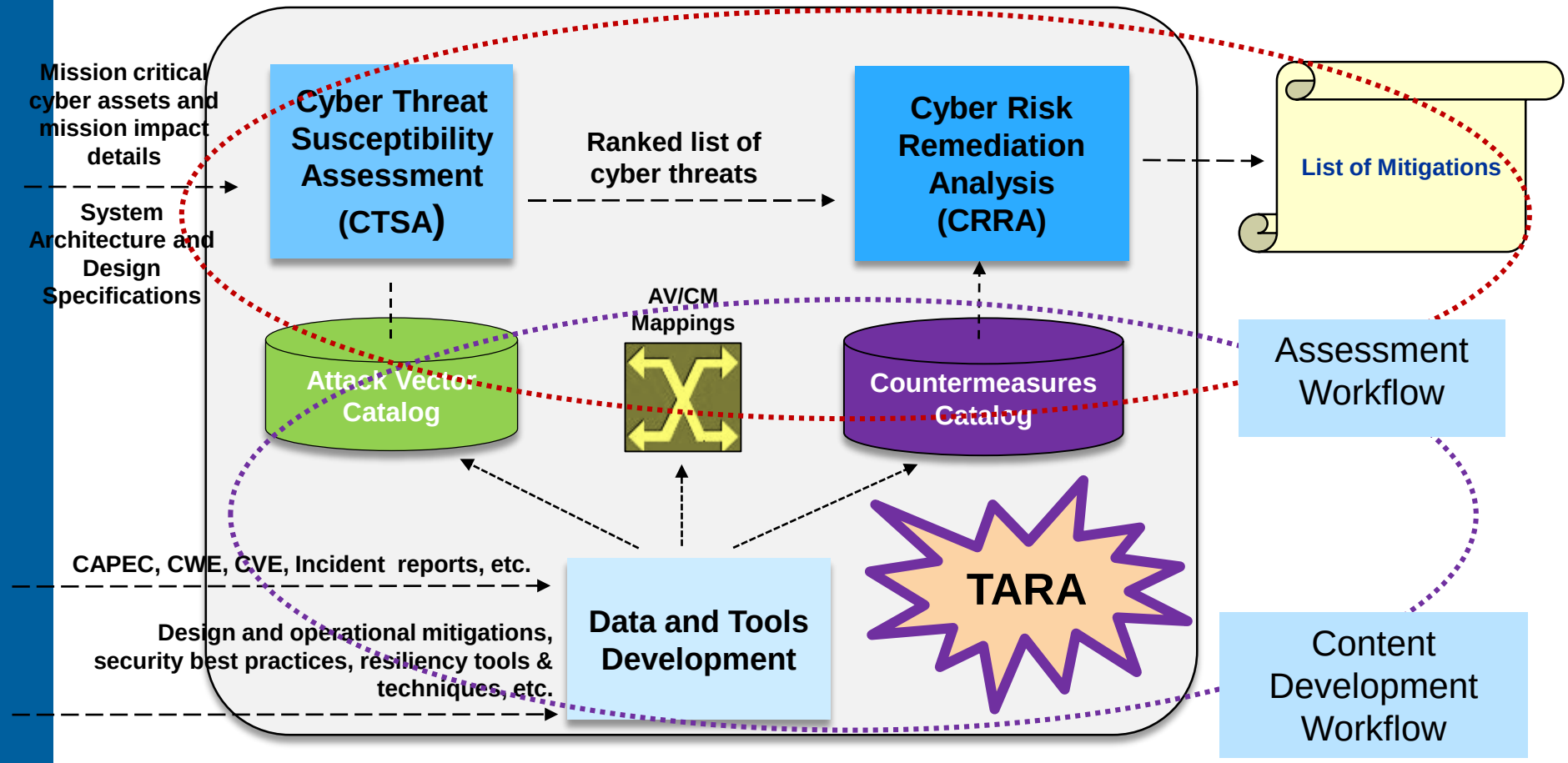


CJA and TARA together support the identification, assessment, and mitigation of cyber risk to mission essential assets

Approved for Public Release

MITRE

TARA Methodology Workflows



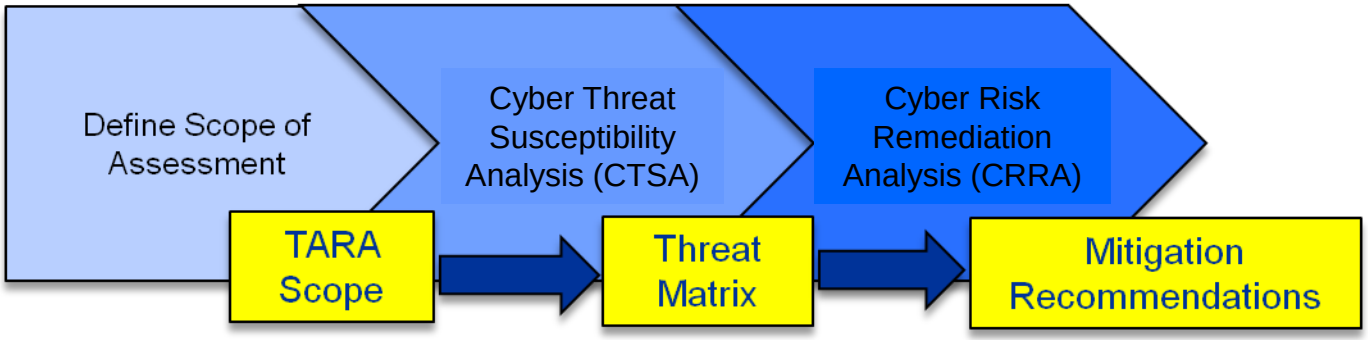
Workflow – Sequence of connected activities that produce useful work

Approved for Public Release

MITRE

TARA Assessment Workflow

Objective to identify and assess cyber threats and select countermeasures effective at mitigating those threats



The evaluation target(s)

The range of threats to be assessed

The adversary

The phase of the system acquisition lifecycle

Verify assessment scope with sponsor

Model the target

Perform catalog search to identify candidate AVs

Eliminate implausible AVs

Define a scoring model to rank plausible AVs

Construct the Threat Matrix

Select AVs to mitigate

Use AV/CM mappings to identify candidate countermeasures (CMs)

Eliminate implausible CMs

Define a scoring model to rank CMs

Select the best CM solution set

Develop well-formed recommendations

Approved for Public Release

MITRE

TARA Assessment Products

Threat Susceptibility Matrix

ID	Attack Vector Name	Risk Score	Evaluation Targets		
			Router	Web Server	Browser
T000105	Cross Site Scripting	2.1			x
T000008	Unsecured SNMP agent	1.9	x		
T000016	Simple Script Injection	1.8	x		x
T000049	Buffer Overflow	1.7	x	x	x
T000001	BIOS replaced with version that allows unsigned updates	1.6	x	x	x
T000021	Man in the Middle Attack	1.4		x	x

Provides a ranked list of cyber threats, mapped to components of the evaluation target

Answers the questions: Where and how is my system most susceptible?

Solution Effectiveness Table

System name: **System XYZ**			Assurance Level: **Medium**					
Countermeasure (CM)			**Mitigation Effectiveness (by Attack Vector ID)**					
ID	Countermeasure Name	Cost Index	T000105	T000008	T000016	T000049	T000001	T000021
			2.1	1.9	1.8	1.7	1.6	1.4
C000023	Change default SNMP community string values	1		P				
C000062	Disable client side scripting	3	P		P			
C000194	Disable hyperlinks in email	1	M		M			
C000015	Verify BIOS implemented security controls after BIOS image update	2					P	
C000018	Use checksums to verify the integrity of downloaded BIOS image updates	2					P	
C000024	Restrict SNMP community string value reuse	2		P				
C000081	Use strong mutual authentication	3						P
C000083	Use cryptography that is sufficient strong	3						P
C000136	Utilize processor-based protection capabilities	1				M		
C000238	Enforce sofware quality standards and guidelines that improve software quality	2				M		
C000090	Validate input fields use of NULL, escape, backslash, meta, and control characters	3	M		M			
C000002	Verify BIOS image write protection	2					M	
C000101	Verify buffer sizes	2				M		
C000247	Ensure trustworthiness of key personnel	3						M
	Totals	30	3	2	3	3	3	3

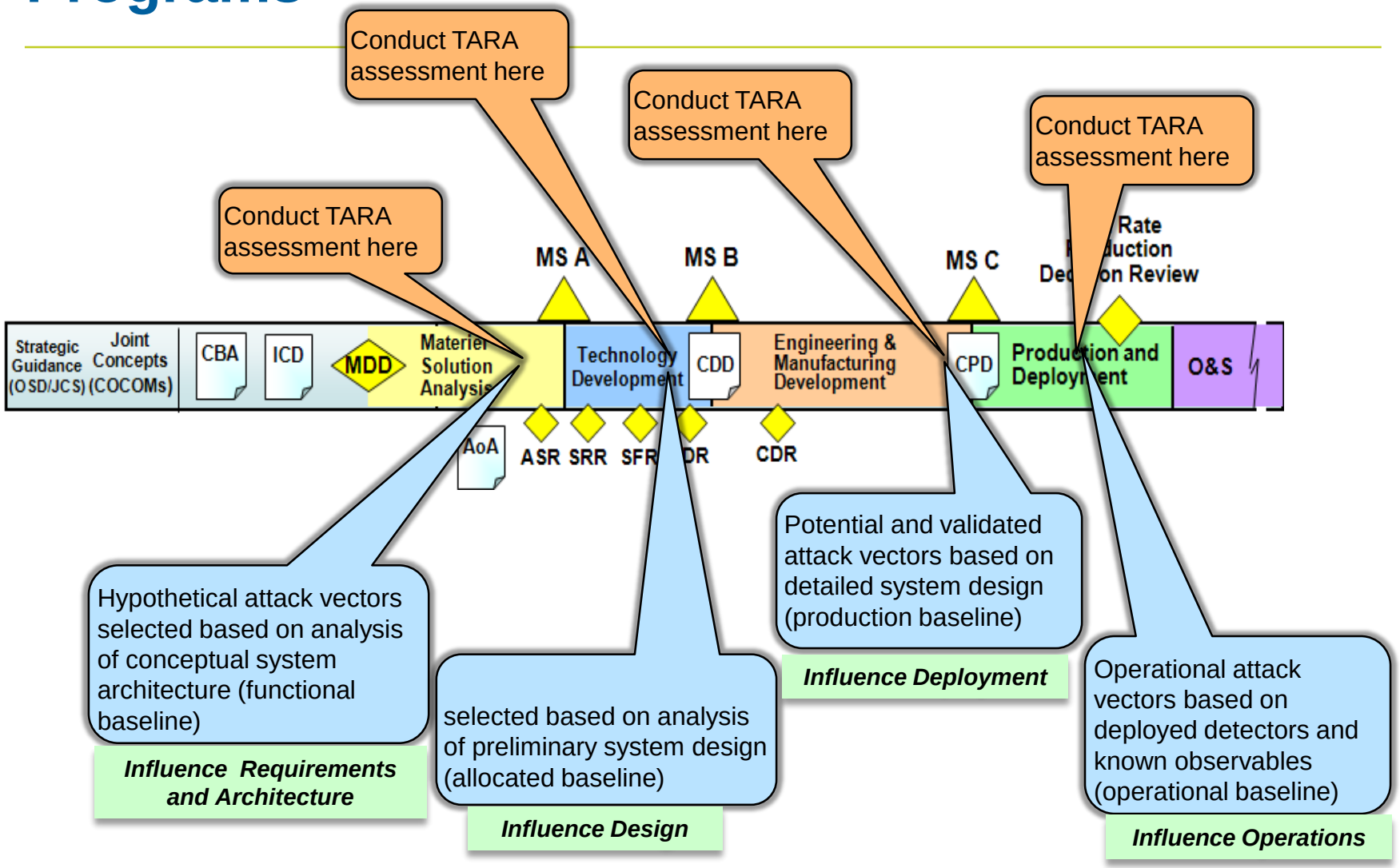
Provides a ranked list of countermeasures, mapped to cyber threats, and identifies the preventative or mitigating effect each countermeasure provides

Answers the questions: How are my threats mitigated and where are the gaps?

Approved for Public Release

MITRE

Threat-based Analysis Influence on Acquisition Programs



Approved for Public Release

MITRE

Applications of TARA

- **Threat-based Analysis of System Architecture**
- **Systems Security Engineering (SSE)**
- **Support to Acquisition Programs**
- **Program Protection Planning**
- **Vulnerability Assessment Planning**
- **Supply Chain Risk Management (SCRM) Analysis**

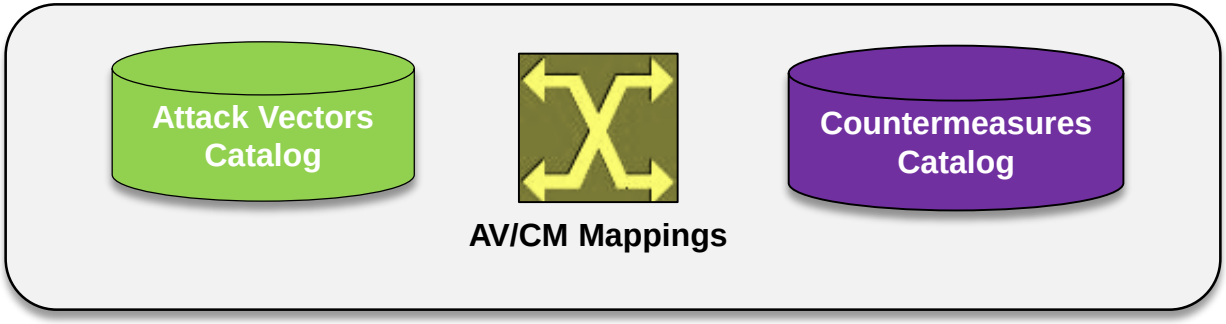
MITRE

TARA Catalog and Toolset

Approved for Public Release

MITRE

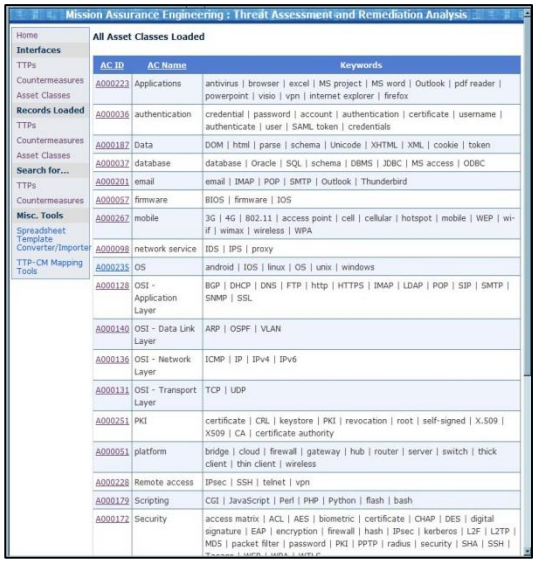
Objectives of the TARA Catalog



- **Provide a repository of Attack Vector (AV) and Countermeasure (CM) data used in TARA assessments**
- **Implement a standard data model to represent AVs and CMs**
- **Help establish consistency from one TARA assessment to the next**

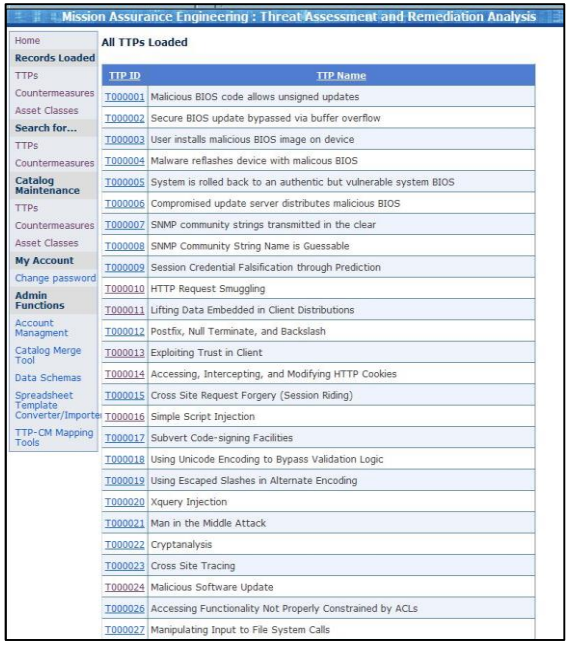
TARA Catalog Data

Vector Groups (VGs)



Named collection of attack vectors, e.g., architectural components, technologies, shopping carts, intrusion sets etc.

Attack Vectors (AVs)



Adversary approaches to compromise a cyber asset

Countermeasures (CMs)

Approaches for mitigating attack vectors

EXAMPLE

Vector Group
Password-based user authentication

Attack Vectors
Dictionary attack,
Rainbow tables,
Brute force, etc.

Countermeasures
Strong passwords,
Password aging,
Account lockouts, etc.

Approved for Public Release

MITRE

Attack Vectors (AVs)

"A sequence of steps performed by an adversary in the course of conducting a cyber attack"

- **Sources of Attack Vector data**
  - Open source data on attack patterns (CAPEC), software weaknesses (CWE), and vulnerabilities (CVE)
  - NIST publications
  - Details on security incidents that occur in the commercial sector
  - Classified security incident reporting
  - Security Threat Assessment Reports (STARs), Integrated Threat Assessments (ITAs), DIA Capstone, NASIC publications, etc.
  - Published security research
    - Weaponized exploits detailed at Blackhat, Defcon, Schmoocon, etc.

Countermeasures (CMs)

"Actions, devices, procedures, or techniques that meet or oppose (i.e., counters) a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken."
Source: CNSS 4009

- **Sources of CM data**
  - Open source data on attack patterns (CAPEC) and software weaknesses (CWE) often includes mitigation details
  - NIST publications
  - Industry recognized security best practices
  - Published security research
    - Journal articles detailing new approaches for detecting anomalous behavior, malware, etc.

Vector Groups

A named collection of Attack Vectors

Types of Vector Groups

- Architectural groupings
  - Client server, network, hardware, software, API, etc.
- Technology groupings
  - Database, web service, XML, email, Unix, Windows, etc.
- Shopping carts
  - Handpicked collection of attack vectors used in an assessment

Vector Group	Group Details		
	Attack Vectors	Counter-measures	Created
Network.routers	34	32	4/1/12
Network.firewalls	7	13	10/15/11
Malware	9	38	10/15/11
IdM.password	8	15	10/15/11
IdM.PKI	6	14	9/1/11
Webclient	21	54	9/1/11
Webservices.webserverplatform	14	33	9/1/11
Webservices.SOAP-UDDI-WSDL	34	73	9/1/11
Webservices.REST			
Webservices.HTTP-HTML-AJAX			
Virtualization	7	12	4/1/12
Crypto.SSL	6	13	9/1/11
Database	6	22	9/1/11
Messaging.JMS	7	29	6/1/12
XML	7	17	2/1/12
Supplychain.COTS	11	52	9/1/11
Software	7	12	9/1/11
Firmware.BIOS	6	15	9/1/11
IPNetwork.BGP	8	17	1/15/13
Networkmanagement.SNMP	6	11	9/1/11
GPS	29	8	9/1/11
Comms.Terrestrial			
Comms.LOS			
Comms.BLOS			
Comms.mobilewifi	6	7	9/1/11

Approved for Public Release

MITRE

Example Vector Group: Software

Attack Vectors

T000005	Exploitation of a zero-day vulnerability
T000006	Counterfeit web sites used to distribute malicious software updates
T000009	Malicious software implantation through 3rd party bundling
T000010	Adversary gains unauthorized access by exploiting a software vulnerability
T000016	Unauthorized / unrestricted copying
T000017	Clandestine changes to software or mission data
T000021	Software defects hidden/obscured by code complexity

Entries are a partial listing, in no particular order

Countermeasures

C000003	Strip debug info from production executables
C000006	Establish a software pedigree
C000014	Make it difficult for the APT
C000019	Apply static code analysis tools to identify software defects
C000020	Establish coding guidelines to improve software quality
C000021	Select programming languages that minimize potential for software defects
C000022	Enforce configuration management (CM) practices that protect source code
C000025	Develop an assurance case for software
C000026	Use dynamic analysis tools to assess software for runtime defects
C000032	Perform risk assessments for open source and unsupported products
C000038	Design to log securely
C000043	Ensure that developers are trained in how to develop secure software

Sources include: Software Assurance Workforce Education and Training Working Group, "Software Assurance: A Curriculum Guide to the Common Body of Knowledge to Produce, Acquire and Sustain Secure Software", DHS, October 2007.

Approved for Public Release

MITRE

TARA Toolset

Web-based tools supporting TARA assessments and catalog development

Catalog Search Tools

Catalog Update Tools

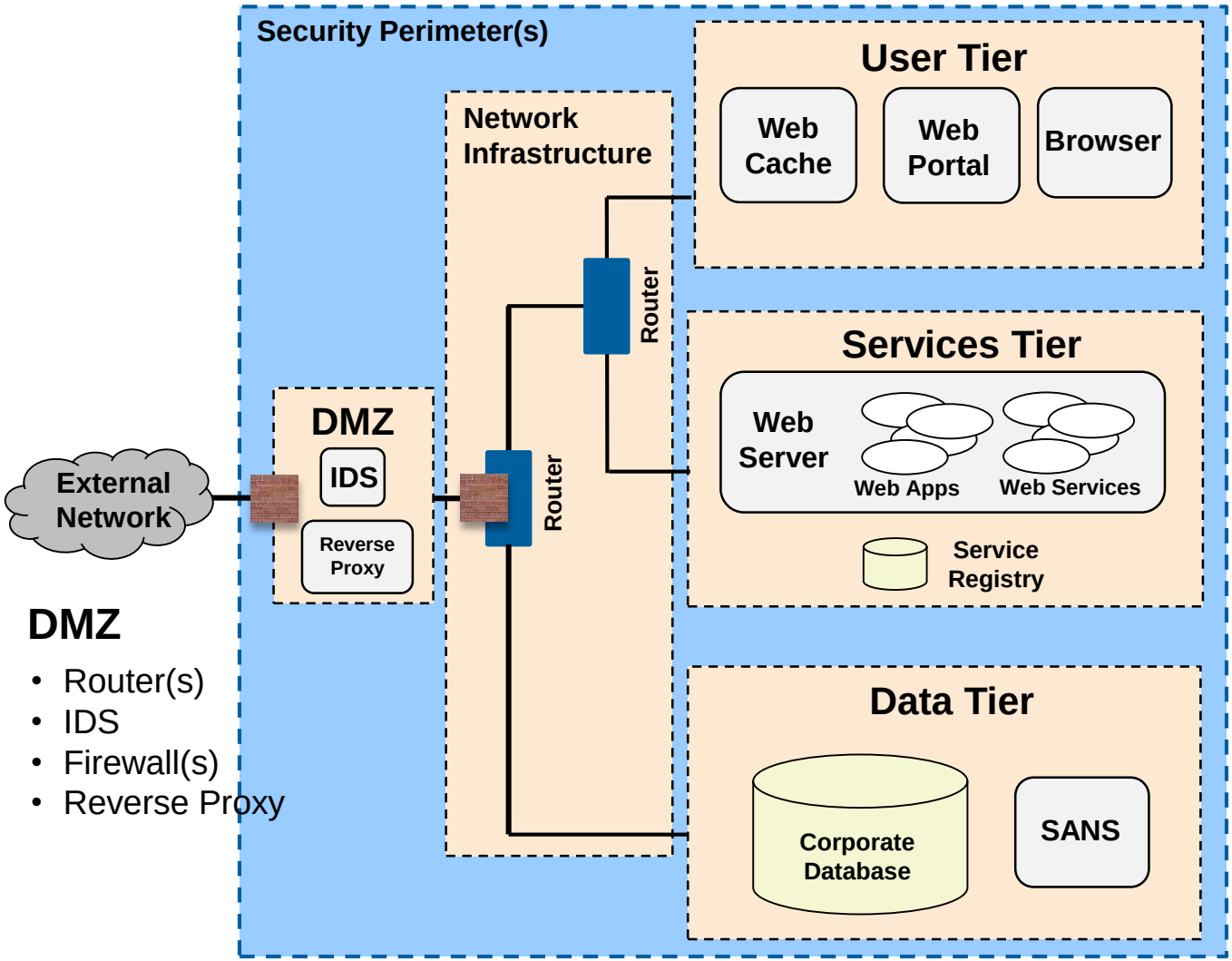
Approved for Public Release

MITRE

Worked Example

MITRE

Target(s) of Evaluation



DMZ

- Router(s)
- IDS
- Firewall(s)
- Reverse Proxy

Network Infrastructure

- Routers
- Switches
- Gateways

User Tier

- Web Cache
- Web Portal
- Browser

Services Tier

- Web Server
- Web Service
- Web Application
- Service Registry

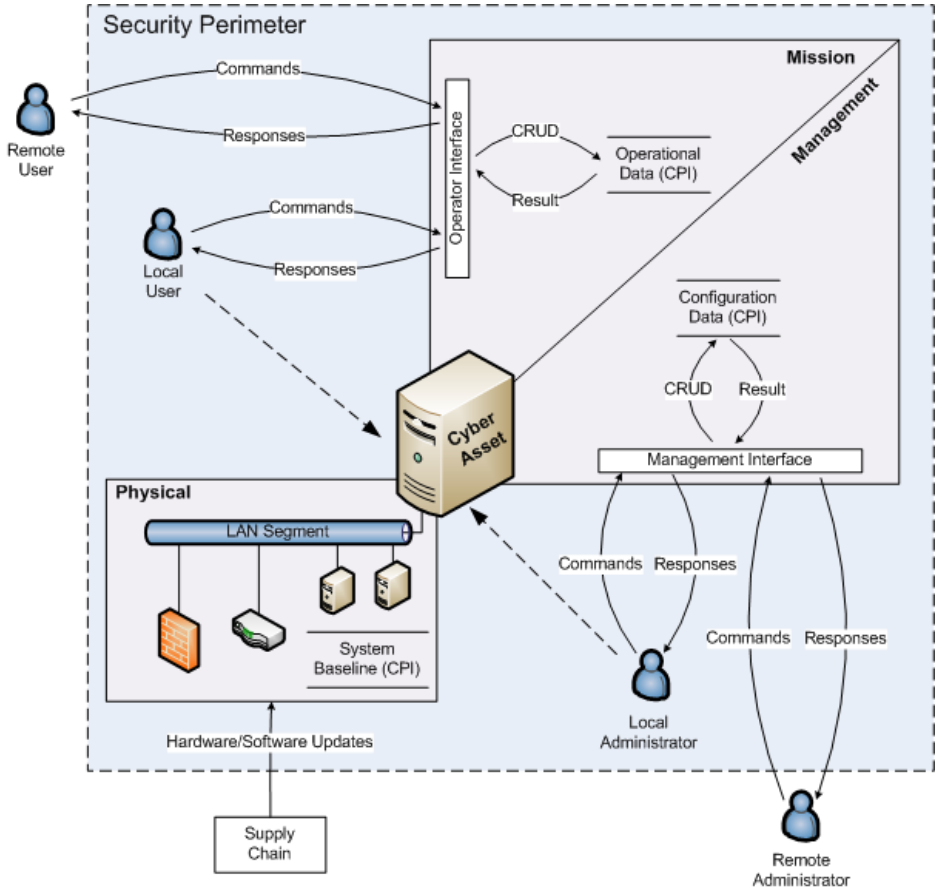
Data Tier

- SANS
- Database

Approved for Public Release

MITRE

Modeling the Attack Surface



ET modeled in 3 planes
- **Mission Plane**
- **Management Plane**
- **Physical Plane**

System Interfaces(s)
- **Standardized functions, CRUD: Create, Read, Update, Delete**
- **Special purpose algorithms**
- **May include Critical Program Information (CPI)**

Security Perimeter(s)
- **Users are local or remote relative to some security perimeter**
- **Nested perimeters subject to penetration by APT**

System data
- **Each plane (mission, management, physical) stores and processes data required within that plane**
- **May include CPI**

CRUD – Create, Read, Update, Delete

A model is a simplified representation of a system to facilitate analysis

Approved for Public Release

MITRE

Filling a Shopping Cart

(with Attack Vectors)

Vector Groups

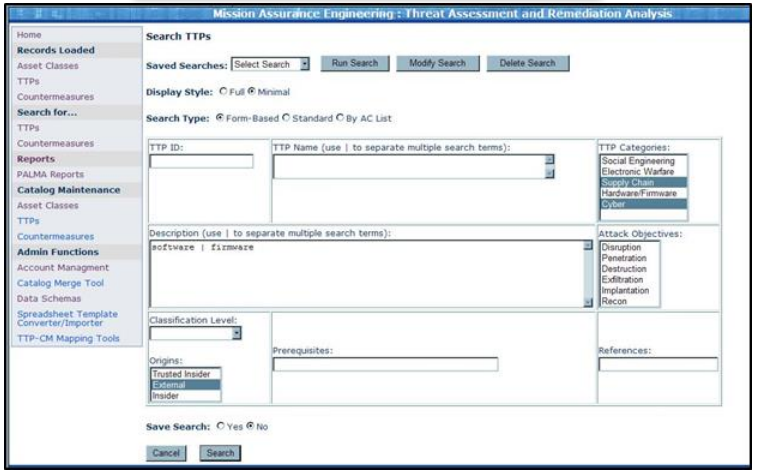
- Web Server
- Web Service
- Web Application
- Database
- XML
- Web 2.0

Search by Vector Group

Search by Category

Search by keyword

Catalog Search



Shopping Cart

Approved for Public Release

MITRE

Example Shopping Cart

ID	Attack Vector Name
T000001	BIOS replaced with version that allows unsigned updates
T000008	Unsecured SNMP agent
T000016	Simple Script Injection
T000021	Man in the Middle Attack
T000049	Buffer Overflow
T000105	Embedding Script (XSS) in HTTP Headers

A shopping cart is a collection of attack vectors being evaluated in a TARA assessment.

Attack vectors were picked at random for this example. In an actual TARA assessment, steps to develop a shopping cart include threat modeling, catalog content development, and external research on the system being evaluated and the technologies it incorporates.

Risk Scoring

Factors for assessing TTP Risk				Factor Weight	Attack Vectors					
Factor Range	Low = 1	Medium = 2	High = 3		T000001	T000008	T000016	T000021	T000049	T000105
Locality: How localized are the effects posed by this TTP?	isolated to single unit	external networks potentially impacted	all units globally and associated infrastructure	0.2	1	2	1	2	2	3
Impact: How serious an impact is loss of data confidentiality resulting from successful application of this TTP?	no impact from TTP	limited impact requiring some remediation	Data spills routinely exercised	0.2	2	1	1	1	2	3
Impact: How serious an impact is loss of system availability resulting from successful application of this TTP?	no impact from TTP	limited impact requiring some remediation	Simulated system outages routinely exercised	0.2	1	1	2	2	1	2
Prior Use: Is there evidence that this TTP has been successfully used before?	no evidence of TTP use	confirmed evidence of TTP use	widespread use of TTP reported	0.3	2	3	3	1	2	1
Stealth: How detectable is this TTP when it is applied?	TTP obvious without monitoring	detection likely with routine monitoring	undetectable	0.1	2	2	1	1	1	2
			Score	1.0	1.6	1.9	1.8	1.4	1.7	2.1

*Risk scoring is an **optional** step in a TARA assessment, which can be performed when a shopping cart includes more attack vectors than can be addressed given time and funding constraints. The spreadsheet above is used to evaluate each attack vector against a set of risk factors. This spreadsheet calculates a risk score for each attack vector as a weighted sum of risk factor values. This scoring approach is intended to rank attack vectors, not to assess absolute risk.*

In a TARA assessment, the risk factors, range of values, weightings, and the calculation can all be tailored to the needs of the program or sponsor. The only requirement is that each attack vector in the shopping cart be treated equally in how relative risk is assessed.

Threat Matrix

ID	Attack Vector Name	Risk Score	Evaluation Targets		
			Router	Web Server	Browser
T000105	Cross Site Scripting	2.1			x
T000008	Unsecured SNMP agent	1.9	x		
T000016	Simple Script Injection	1.8	x		x
T000049	Buffer Overflow	1.7	x	x	x
T000001	BIOS replaced with version that allows unsigned updates	1.6	x	x	x
T000021	Man in the Middle Attack	1.4		x	x

*The **Threat [Susceptibility] Matrix** combines shopping cart and risk scoring data across the range of evaluation targets being assessed. This artifact is a primary deliverable and represents the transition from threat susceptibility analysis to risk remediation analysis in the TARA methodology.*

MITRE

The AV/CM Mapping Table

Countermeasure (CM)			Mitigation Effectiveness (by Attack Vector ID)					
CM ID	Name	Cost Index	T000001	T000008	T000016	T000049	T000021	T000105
C000001	Verify secure BIOS update non-bypassability	Medium	M					
C000002	Verify BIOS image write protection	Low	M					
C000003	Verify recovery process to restore last-known-good BIOS image	Medium	M					
C000005	Institute secure BIOS update capabilities using RTU	High	P					
C000015	Verify BIOS implemented security controls after BIOS image update	Low	P					
C000018	Use checksums to verify the integrity of downloaded BIOS image updates	Low	P					
C000023	Change default SNMP community string values	Very Low		P				
C000024	Restrict SNMP community string value reuse	Low		P				
C000041	Use same character encoding	Medium			P			
C000062	Disable client side scripting	Medium			P			P
C000064	Do not deploy content proxies that mask where data originates from	High			P			
C000065	Sanitize outbound content	High			M			
C000079	Only accept PKI credentials from a trusted certificate authority	Medium					M	
C000081	Use strong mutual authentication	Medium					P	
C000083	Use cryptography that is sufficient strong	Medium					P	
C000090	Validate input fields use of NULL, escape, backslash, meta, and control characters	Medium			M			M
C000101	Verify buffer sizes	Low				M		
C000103	Match buffer size to data input size	Low				M		
C000112	Restrict source of format strings	Low			M			
C000115	Limit user functional roles	Medium						M
C000121	Verify input sources	Medium						P
C000132	Use sandboxing to isolate running software	Medium						M
C000134	Select programming languages that minimize potential software defects	Medium				M		
C000135	Avoid use of dangerous memory functions and operations	Low				M		
C000136	Utilize processor-based protection capabilities	Very Low				M		
C000142	Enforce mutual authentication between communication parties	Medium					P	
C000146	Enable SSL TLS to protect sensitive web pages	Medium					P	
C000194	Disable hyperlinks in email	Very Low			M			M
C000220	Utilize best practice malware detection approaches	Medium			M			M
C000238	Enforce sofware quality standards and guidelines that improve software quality	Low				M		
C000247	Ensure trustworthiness of key personnel	Medium					M	

*The AVCM mapping table depicts the association of countermeasures to attack vectors in the TARA catalog. Catalog tools provide the means to export mapping table data in spreadsheet form, as depicted. In this example, each mapping characterizes whether a countermeasure has a **[P]**reventative effect or a **[M]**itigating effect for each attack vector listed in the threat matrix. A cost index is associated with each countermeasure to reflect the relative cost of ownership for that countermeasure on a linear scale [very low...very high]. These default cost index values can be tailored to reflect truth about the program, e.g., the cost of ownership may be significantly lower if the CM is already implemented as a security measure or security practice in a system that is already fielded.*

Approved for Public Release

MITRE

Countermeasure Scoring

Countermeasure (CM)					Mitigation Effectiveness (by Attack Vector ID)					
CM ID	Name	Cost Index	Utility Score	U/C Ratio	T000105	T000008	T000016	T000049	T000001	T000021
					2.1	1.9	1.8	1.7	1.6	1.4
C000023	Change default SNMP community string values	1	6	6.00		P				
C000062	Disable client side scripting	3	12	4.00	P		P			
C000194	Disable hyperlinks in email	1	4	4.00	M		M			
C000015	Verify BIOS implemented security controls after BIOS image update	2	6	3.00					P	
C000018	Use checksums to verify the integrity of downloaded BIOS image updates	2	6	3.00					P	
C000024	Restrict SNMP community string value reuse	2	6	3.00		P				
C000041	Use same character encoding	3	6	2.00			P			
C000081	Use strong mutual authentication	3	6	2.00						P
C000083	Use cryptography that is sufficient strong	3	6	2.00						P
C000121	Verify input sources	3	6	2.00	P					
C000136	Utilize processor-based protection capabilities	1	2	2.00				M		
C000142	Enforce mutual authentication between communication parties	3	6	2.00						P
C000146	Enable SSL TLS to protect sensitive web pages	3	6	2.00						P
C000005	Institute secure BIOS update capabilities using RTU	4	6	1.50					P	
C000064	Do not deploy content proxies that mask where data originates from	4	6	1.50			P			
C000238	Enforce sofware quality standards and guidelines that improve software quality	2	3	1.50				M		
C000090	Validate input fields use of NULL, escape, backslash, meta, and control characters	3	4	1.33	M		M			
C000220	Utilize best practice malware detection approaches	3	4	1.33	M		M			
C000002	Verify BIOS image write protection	2	2	1.00					M	
C000101	Verify buffer sizes	2	2	1.00				M		
C000103	Match buffer size to data input size	2	2	1.00				M		
C000112	Restrict source of format strings	2	2	1.00			M			
C000135	Avoid use of dangerous memory functions and operations	2	2	1.00				M		
C000247	Ensure trustworthiness of key personnel	3	3	1.00						M
C000001	Verify secure BIOS update non-bypassability	3	2	0.67					M	
C000003	Verify recovery process to restore last-known-good BIOS image	3	2	0.67					M	
C000079	Only accept PKI credentials from a trusted certificate authority	3	2	0.67						M
C000115	Limit user functional roles	3	2	0.67	M					
C000132	Use sandboxing to isolate running software	3	2	0.67	M					
C000134	Select programming languages that minimize potential software defects	3	2	0.67				M		
C000065	Sanitize outbound content	4	2	0.50			M			

*TARA provides a default approach to score countermeasures based on cost benefit analysis. This is an **optional** step used to rank countermeasures prior to their selection. This approach calculates a **Utility/Cost (U/C) ratio** for each countermeasure, based on its cost index and a utility score, which is calculated as the cumulative mitigation value of that countermeasure over the range of attack vectors. In this each example, a score of 6 is assigned to each [P]reventative mapping and a score of 2 is assigned to each [M]itigating mapping. Additionally, the cost index ordinal scale [very low... very high] is remapped to a numeric scale [1...5] in order to compute U/C ratios.*

Note that the scores assigned to mappings and the numeric scale used for cost can be tailored to suit the needs of the program. Once a U/C ratio is calculated for each countermeasure, the list is sorted so that countermeasures with higher U/C ratios appear on top and attack vectors are reordered left to right by decreasing risk score.

Countermeasure Selection Strategy

Countermeasure (CM) Selection Strategies

Assurance level: **Low**

For each attack vector
- *At least 2 CMs total*
- *At least 1 Preventative CM*
- *At least 1 Mitigation CM*

Assurance level: **Medium**

For each attack vector
- *At least 3 CMs total*
- *At least 1 Preventative CM*
- *At least 1 Mitigation CM*

Assurance level: **High**

For each attack vector
- *At least 5 CMs total*
- *At least 2 Preventative CM*
- *At least 1 Mitigation CM*

The countermeasure selection strategy establishes constraints on the selection of countermeasures in terms of the minimum number of preventative, mitigating, and total countermeasures required for each attack vector.

In this example, 3 assurance levels are defined: low, medium, high, each requiring progressively more total countermeasures.

This strategy can be tuned to the needs of a particular program or sponsor in terms of the number of countermeasures, the ratio of preventative to mitigating countermeasures etc.

Countermeasure Selection

Countermeasure (CM)		Mitigation Effectiveness (by Attack Vector ID)					
CM ID	Name	T000105	T000008	T000016	T000049	T000001	T000021
		2.1	1.9	1.8	1.7	1.6	1.4
C000023	Change default SNMP community string values		**P**				
C000062	Disable client side scripting	**P**		**P**			
C000194	Disable hyperlinks in email	**M**		**M**			
C000015	Verify BIOS implemented security controls after BIOS image update					**P**	
C000018	Use checksums to verify the integrity of downloaded BIOS image updates					**P**	
C000024	Restrict SNMP community string value reuse		**P**				
C000041	Use same character encoding			P			
C000081	Use strong mutual authentication						**P**
C000083	Use cryptography that is sufficient strong						**P**
C000121	Verify input sources	P					
C000136	Utilize processor-based protection capabilities				**M**		
C000142	Enforce mutual authentication between communication parties						P
C000146	Enable SSL TLS to protect sensitive web pages						P
C000005	Institute secure BIOS update capabilities using RTU					P	
C000064	Do not deploy content proxies that mask where data originates from			P			
C000238	Enforce sofware quality standards and guidelines that improve software quality				**M**		
C000090	Validate input fields use of NULL, escape, backslash, meta, and control characters	**M**		**M**			
C000220	Utilize best practice malware detection approaches	M		M			
C000002	Verify BIOS image write protection					**M**	
C000101	Verify buffer sizes				**M**		
C000103	Match buffer size to data input size				M		
C000112	Restrict source of format strings			M			
C000135	Avoid use of dangerous memory functions and operations				M		
C000247	Ensure trustworthiness of key personnel						**M**
C000001	Verify secure BIOS update non-bypassability					M	
C000003	Verify recovery process to restore last-known-good BIOS image					M	
C000079	Only accept PKI credentials from a trusted certificate authority						M
C000115	Limit user functional roles	M					
C000132	Use sandboxing to isolate running software	M					
C000134	Select programming languages that minimize potential software defects				M		
C000065	Sanitize outbound content			M			

Countermeasures are selected by applying the countermeasure selection strategy to each attack vector from left to right in the countermeasure scoring table. Countermeasure selection starts at the top of the table where U/C ratios are the highest. Selected countermeasures are highlighted.

In this example, the selection strategy for medium assurance is used. However the strategy cannot be fully satisfied for each attack vector listed, resulting in gaps. Alternative solution sets that meet the strategy constraints can be developed and provide the basis for incorporating sensitivity analysis in the course of a TARA assessment.

Solution Effectiveness

System name: **System XYZ**			Assurance Level: **Medium**					
Countermeasure (CM)			**Mitigation Effectiveness (by Attack Vector ID)**					
ID	Countermeasure Name	Cost Index	T000105	T000008	T000016	T000049	T000001	T000021
			2.1	1.9	1.8	1.7	1.6	1.4
C000023	Change default SNMP community string values	1		P				
C000062	Disable client side scripting	3	P		P			
C000194	Disable hyperlinks in email	1	M		M			
C000015	Verify BIOS implemented security controls after BIOS image update	2					P	
C000018	Use checksums to verify the integrity of downloaded BIOS image updates	2					P	
C000024	Restrict SNMP community string value reuse	2		P				
C000081	Use strong mutual authentication	3						P
C000083	Use cryptography that is sufficient strong	3						P
C000136	Utilize processor-based protection capabilities	1				M		
C000238	Enforce sofware quality standards and guidelines that improve software quality	2				M		
C000090	Validate input fields use of NULL, escape, backslash, meta, and control characters	3	M		M			
C000002	Verify BIOS image write protection	2					M	
C000101	Verify buffer sizes	2				M		
C000247	Ensure trustworthiness of key personnel	3						M
	Totals	30	3	2	3	3	3	3

The solution effectiveness table lists countermeasures that were selected, with each countermeasure detailing the preventative and/or mitigating effects it has on each attack vector assessed. At the bottom of the table a summary is provided to indicate whether the selection strategy was successful for each attack vector, with green indicating success and yellow indicating where a gap exists.

In the example above, T000008 has a gap relative to both the total number of countermeasures applied and the lack of a mitigating countermeasure, while the gap identified for T000049 relates to the absence of a preventative countermeasure. The summary also includes a summary cost index to support comparison with alternative solution sets. This artifact is a primary deliverable and represents completion of the risk remediation phase of a TARA assessment.

Approved for Public Release

MITRE

Summary

- **TARA is an engineering approach that is rigorous and repeatable, provides traceability, identifies gaps, and develops defense-in-depth**
- **TARA's objective is to influence programs early in the acquisition lifecycle where *the cost of change is less***
- **TARA applies model based systems engineering and tradeoff analysis to system security engineering**
- **TARA maintains and utilizes catalogs of attack vector and countermeasure data that incorporates data from a variety of sources including CAPEC, CWE, and CVE**
- **The TARA approach is flexible and can be tailored to meet the needs of MITRE sponsors and programs**
- **TARA has been applied to several Army, Navy, Air Force, and DoD acquisition programs**

For More Information

Online information and resources

http://www.mitre.org/publications/technical-papers/threat-assessment--remediation-analysis-tara

http://www.mitre.org/publications/systems-engineering-guide/enterprise-engineering/systems-engineering-for-mission-assurance/cyber-risk-remediation-analysis

http://www.mitre.org/publications/systems-engineering-guide/enterprise-engineering/systems-engineering-for-mission-assurance/cyber-threat-susceptibility-assessment

To schedule a demo, consultation, or for general inquiries

Jackson Wynn
(781) 271-3419
jwynn@mitre.org

Approved for Public Release

MITRE