Cybersecurity Architectural Risk Analysis Process

May 2017

As a leading technology and solutions development organization, the Alliance for Telecommunications Industry Solutions (ATIS) brings together the top global Information and Communications Technology (ICT) companies to advance the industry's most pressing business priorities. ATIS' nearly 200 member companies are currently working to address the All-IP transition, network functions virtualization, big data analytics, cloud services, device solutions, emergency services, M2M, cyber security, network evolution, quality of service, billing support, operations, and much more. These priorities follow a fast-track development lifecycle – from design and innovation through standards, specifications, requirements, business use cases, software toolkits, open source solutions, and interoperability testing.

ATIS is accredited by the American National Standards Institute (ANSI). The organization is the North American Organizational Partner for the 3rd Generation Partnership Project (3GPP), a founding Partner of the oneM2M global initiative, a member of and major U.S. contributor to the International Telecommunication Union (ITU), as well as a member of the Inter-American Telecommunication Commission (CITEL). For more information, visit www.atis.org.

Published by
Alliance for Telecommunications Industry Solutions
1200 G Street, NW, Suite 500
Washington, DC 20005

Table of Contents

Table of Figures

Table of Tables

1 Executive Summary

ATIS members, who consist of leading service providers and vendors in the Information and Communications Technology (ICT) industry, have collaborated on a Cybersecurity Ad Hoc. Launched in July 2015, one of the group's objectives is to create tools and practices to help organizations manage cybersecurity risk in the ICT industry. One outcome of the Cybersecurity Ad Hoc's work has been to create a process for performing an Architectural Risk Analysis (ARA) on ICT solutions for the purpose of enabling the proactive development of cybersecurity risk management steps for these solutions. This process includes procedures to determine security goals, identify and assess potential risks, and develop proactive steps to mitigate identified risks. The ARA Process explained in this document relies upon industry cybersecurity best practices to support many of the details involved in executing the process. This document also includes an illustrative example of the use of the process for a hypothetical health monitoring device and associated services which are delivered in an ICT service provider-managed context. Finally, some potential areas for additional work are identified to broaden the scope of the ARA Process and to further simplify its application.

2 Introduction

2.1 Purpose

This document describes a recommended process for ARA for ICT solutions. Through this process, threat modeling techniques are applied to ensure proper security considerations are part of the solution by design. This process will help prevent security from being an afterthought in the development of ICT solutions.

2.2 Scope

The process will help the service provider and their associated 3rd party partners and suppliers to assess the architecture by identifying key points where security controls are needed to thwart potential threats and the associated risks.

The process also helps to assess the architecture connecting the service provider's network to the cloud provider by identifying where security controls are needed. It includes an approach to properly identify and address security risks related to 3rd party components and platforms in service provider solutions.

2.3 Target Audience

The process is intended for those involved in planning, designing, engineering, evaluating, and implementing ICT solutions. Backgrounds in general computing, networking, and security are helpful in order to effectively apply the process and gain the best results.

3 ARA Process Elements & Overview

The process involves defining the attack surface of the solution and assessing how well the associated threats are mitigated through security controls. The high-level steps are described in Figure 3.1, which divides the process into three broad activities: Architecture Discovery, Threat Identification, and Risk Analysis.

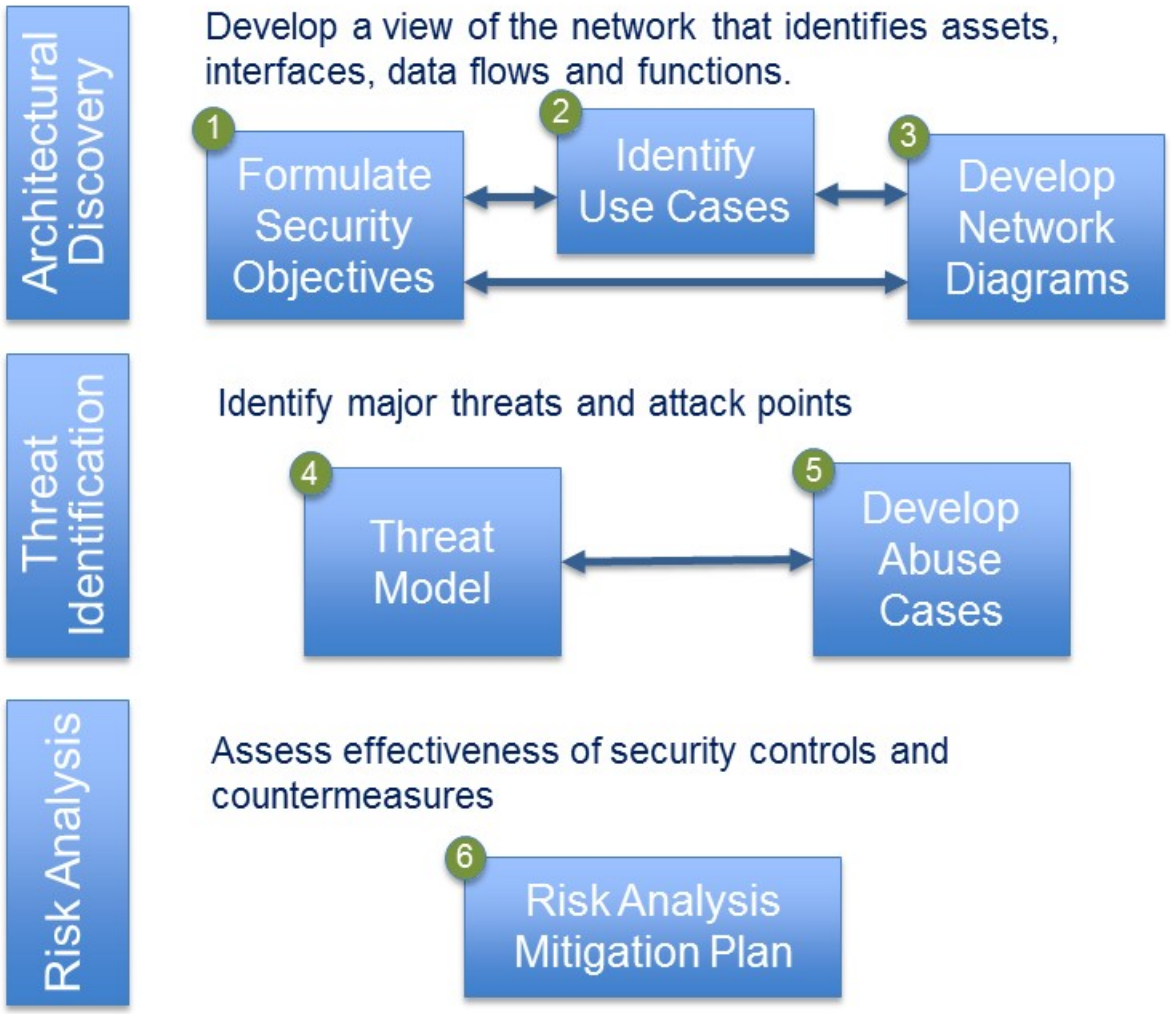


Figure 3.1 – Process Steps

The process may begin at any of the three steps within the Architectural Discovery activity as shown in the figure. The primary objective of Architectural Discovery is to develop a view of the network that identifies major assets, interfaces, dataflows, and functions. When applying the process to an existing network/system, prior information may be reused. It is often useful to iterate among these three steps as more information is gathered and developed.

For large complex systems, Architectural Discovery can be applied in a hierarchical manner. That is, the network can be divided into segments with each segment being analyzed separately. The segments can then be brought together for a complete analysis.

The deliverables for the Architectural Discovery activity include a list of use cases, network and/or data flow diagrams, tables of interfaces and protocols and descriptive text of security objectives and key assets.

The primary objective of the Threat Identification activity is to identify the major threats and attack points relevant to architecture and as such, rely on the deliverables from the Architectural Discovery work. Annotating diagrams with potential attack points is recommended.

Abuse Cases are a combination of the Use Cases and the identified attack vectors. It is important to note that each interface may be an attack point when doing this analysis. Iterating between abuse case and threat model efforts is recommended until thorough coverage is achieved.

The deliverables for the Threat Identification activity include a ranked list of threats correlated to assets and interfaces, a list of abuse cases and annotated diagrams indicating attack points.

The primary objective of the Risk Analysis activity is to assess the effectiveness of security controls and countermeasures that may be used. This activity utilizes the outputs from the previous activity to iterate through the entire list of threats individually until each has been fully evaluated.

The deliverables for this activity include a set of revised diagrams and reference material that reflect coverage of the security controls and any threats that may not be fully mitigated along with the set of actions that may be required to strengthen security.

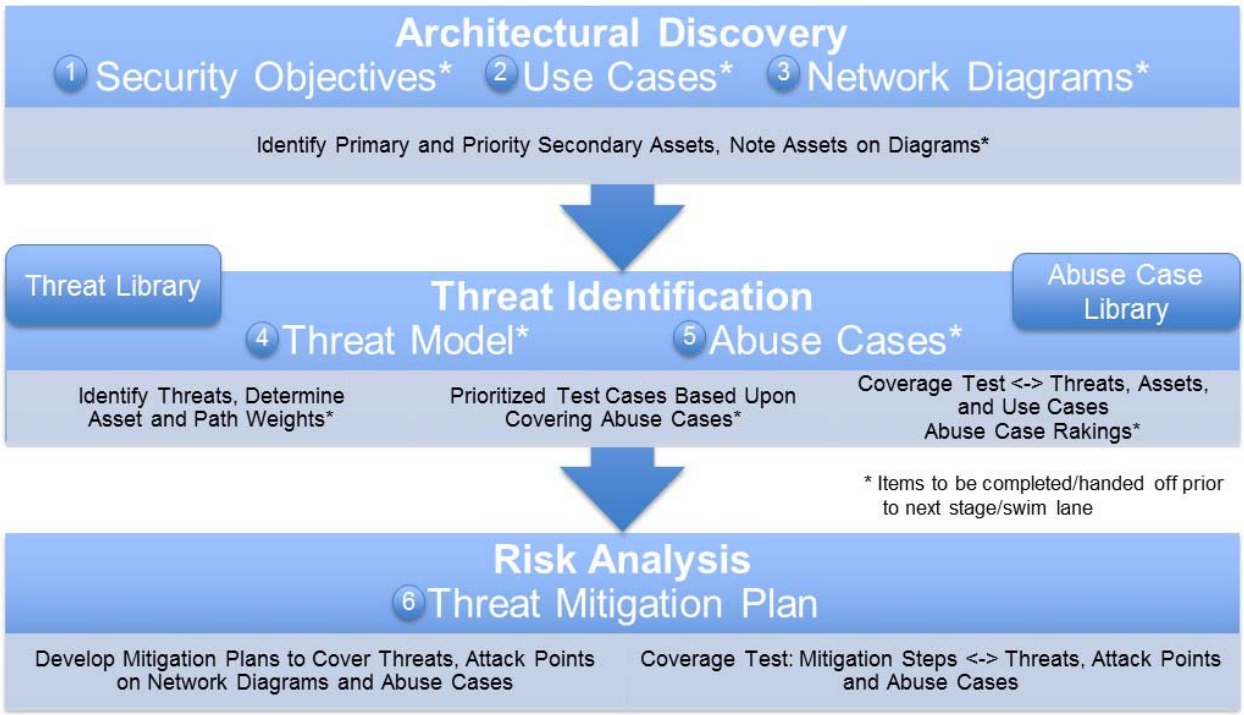


Figure 3.2 – ARA Process Lanes and Hand-offs

It is possible that there may be multiple groups involved in applying the ARA Process for a complex solution being built by a large organization. To this end, and to help ensure that a complex solution is analyzed in a complete fashion, the ARA Process includes a focus on hand-offs between the major process lanes. Figure 3.2 illustrates these hand-offs. It is recommended that the items identified by an asterisk (*) in Figure 3.2Figure 3.4 in each lane of the ARA Process be completed, reviewed, and documented prior to completing the steps in the next lane of the process. The documentation of these items is also important to support the traceability tools that are part of many of the ARA Process steps.

Although a specific process flow is implied by the diagrams, variations may be appropriate depending on whether one is analyzing an existing solution or is in the initial design phase of a solution. Information gathering and threat discovery are key elements of the process and some items may lend themselves to parallel activities, iterative cycles, or beginning with various steps, depending on information that may be readily available. Table 3.1 provides an expanded description of each step, identifies key contributors, and defines entry and exit criteria for each step.

Table 3.1 – Process Steps

Step	Short Description	Long Description	Contributors	Input	Output
1	Security Objectives	Document security objectives that correlate to the functionality and services provided by the network. Correlate to Confidentiality, Integrity, and Availability (CIA). Define 3rd Party Trust Models.	Lead designers or architect, including those representing 3rd Party components.	Project plan.	Documented security objectives.
2	Use Cases	Identify/document functional Use Cases with security impacts Work includes relevant 3rd Party Use Cases. Identity assets and their attributes that may be of interest to attackers.	Product Management, lead designers, architect, or engineers.	Project goals, requirements, or pre-existing use cases	Use cases and assets that may have security impact.
3	Network Diagrams	Develop (or locate) an architectural diagram that depicts the major components, interfaces, and trust boundaries of the solution. Ensure that data flows, trust boundaries, and protocols of critical data paths are identified. Work includes relevant 3rd Party components.	Lead designers or architect.	Existing documentation or high-level scenarios/use-cases of functional operation.	Architectural diagram depicting the solution.
4	Threat Model	Identify risks, threats, threat vectors, and actors. Document risks and threats that may target assets and associated interfaces and protocols.	Developer or engineer.	Threat library derived from industry research.	Annotated architectural diagrams and use case documentation with threats identified.
5	Abuse Cases	Document how use cases could be abused, identifying the overall solution impact and if possible, the methods of misuse.	Lead designers or architect.	Design and architecture documentation.	Ranked list of threats and abuse cases and list of key assets and interfaces
6	Threat Mitigation Plan	Does each threat have an associated security control to mitigate it? Determine additional controls and/or research needed to mitigate threats including those which are manifested via 3rd Party components	Developer or engineer.	Annotated architectural diagrams and use case documentation with threats identified.	Description of how each threat is/is-not mitigated. Annotate diagrams with security control and indicate mitigation. Actions to strengthen security.

Annex B includes an ARA process chart that summarizes the key activities, descriptions, and deliverables in a single figure.

3.1 Architectural Discovery

The objective of the architectural discovery activity is to develop a view of the solution under analysis that identifies major assets, interfaces, data flows, and functions. These items comprise the core components that contribute to the network's attack surface and are critical to the remaining steps in the process. Components provided by 3rd parties that can contribute to the attack surface of the solution should also be explicitly identified along with expectations for trusted and un-trusted assets, interfaces, data flows, and functions for such components.

3.1.1 Security Objectives

Clear security objectives are required to provide an operational context for developing a threat model and conducting the detailed analysis. Security objectives, or goals, should be as specific as possible so that analysis can clearly assess whether the goals have been met. Answering the following questions may help to formulate security objectives that are clear, concise, and specific:

a) What are the key assets that comprise the solution? Identify critical components and data that are stored, processed, or vital to driving operational transactions. Key assets that are provided by 3rd party components should also be clearly identified and the expected trust levels associated with those assets designated.
b) How important are those assets to the organization, given identified business goals? Classify key assets in order of importance or value to the organization, using a 4- or 5-level scale. Consider the impact to finances, reputation, and operational efficiency when classifying those assets. The classification will serve as a guide in determining the degree of protection from attack these assets require.
c) What are the key services that the solution provides? Rank them by importance.
d) What services and assets must be protected against attack and at what cost?

By considering the above, security goals can be developed to address key security objectives relating to the protection of assets and operational functionality from attack. Refer to Clause 5.1 for additional information.

Almost all real-world solutions will include key components from 3rd parties. Each 3rd party component should be clearly identified, and the following requirements for each component should be specified as part of the solution's security objectives:

- What assets does the component provide?
- What is the expected trust level associated with the functionality, data, and interfaces provided?
- How will each asset and interface be treated from a trust level perspective?
    - As fully trusted.
    - As completely un-trusted.
    - With a limited level of trust that should be clearly specified.

The 3rd party security objectives should be carefully reviewed with appropriate personnel in the 3rd party organization, and any changes made subsequent to such a review should be carefully examined for impact on the solution's ARA results. The combination of the 3rd party components ***Security Objectives***, ***Assets***, and ***Trust Requirements*** that are specified at this step constitute a ***Trust Model*** for a given 3rd party component in the overall solution.

3.1.2 Use Cases

Once security objectives have been documented, functional Use Cases and their associated assets and attributes that may be of interest to attackers can be identified.

3.1.2.1 Use Cases

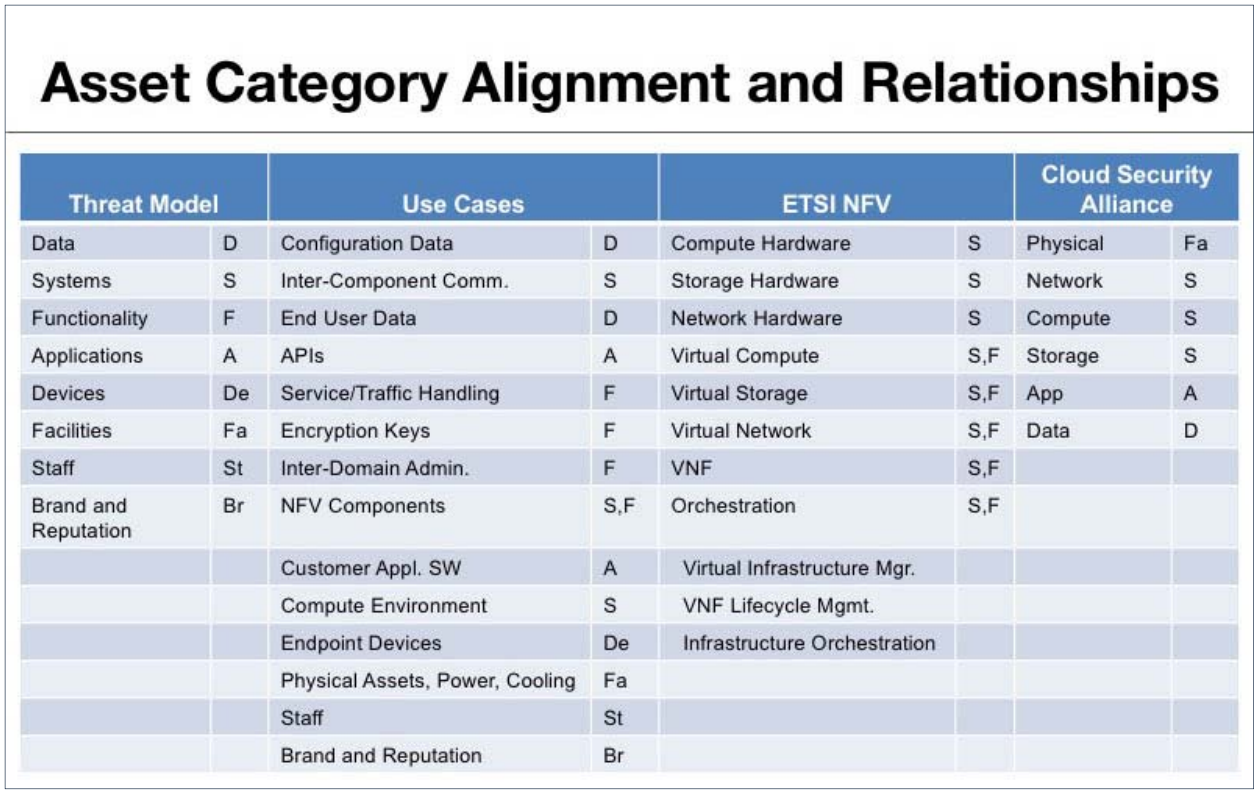
It is important to create a fully documented set of use cases that describe the functionality of a solution to which the ARA process is applied. Once these use cases are developed, the next step is to identify the assets associated with the solution's use cases as a starting point for the threat analysis (see clause 3.2.1) and architectural diagram (see clause 3.1.3) steps that follow.

3.1.2.2 Assets

In the context of the ARA process, an ***Asset*** is defined as *a resource of value, such as the data in a database* or *solution resource, which must be protected.* Two types of assets were identified and defined as follows:

- ***Primary Assets*** – Resources that represent the primary or fundamental value delivered and/or the associated service or solution.
- ***Priority Secondary Assets*** – Resources upon which the primary asset depends to the extent that they can be attacked with the intent of enabling *likely threat actors' goals* associated with the ***Primary Asset***.

There are several asset models being used across the industry practices that have been reviewed (see Annex Annex A: Industry Reference Material). The existing assets from the European Telecommunications Standards Institute (ETSI) NFV and the Cloud Security Alliance (CSA) address sub-sets of the assets defined here.



Asset Category Alignment and Relationships

Threat Model		Use Cases		ETSI NFV		Cloud Security Alliance	
Data	D	Configuration Data	D	Compute Hardware	S	Physical	Fa
Systems	S	Inter-Component Comm.	S	Storage Hardware	S	Network	S
Functionality	F	End User Data	D	Network Hardware	S	Compute	S
Applications	A	APIs	A	Virtual Compute	S,F	Storage	S
Devices	De	Service/Traffic Handling	F	Virtual Storage	S,F	App	A
Facilities	Fa	Encryption Keys	F	Virtual Network	S,F	Data	D
Staff	St	Inter-Domain Admin.	F	VNF	S,F		
Brand and Reputation	Br	NFV Components	S,F	Orchestration	S,F		
		Customer Appl. SW	A	Virtual Infrastructure Mgr.			
		Compute Environment	S	VNF Lifecycle Mgmt.			
		Endpoint Devices	De	Infrastructure Orchestration			
		Physical Assets, Power, Cooling	Fa				
		Staff	St				
		Brand and Reputation	Br				

Figure 3.3 – ARA Process Asset Cross-References – ETSI NFV and CSA

Figure 3.3 above illustrates how the assets identified as part of the ARA Process relate to those defined by ETSI NFV and CSA. This information is included to facilitate the use of the ETSI and CSA Best Practices in support of the ARA Process.

Example assets that may be used as a starting point for further analysis are shown below. It is likely that other assets may be identified, and that assets presented here might be primary assets in some contexts and priority secondary assets in others.

- **Inter-Component Communications** – These assets provide communications between components in the solution. They could include software interfaces such as those used between NFV components, physical links/protocols, or other inter-element communications mechanisms.
- **Configuration Data** – This is data that is used to determine the function of the solution. Examples would include element configuration settings, communications settings, routing information, numbering plan data, etc.
- **End-User Data** – This is data that is specific to individual end users. It could include names, physical or logical addresses, telephone numbers, credit card, or other financial information, etc.
- **APIs** – Applications Programming Interface assets includes software interfaces at key solution boundaries used for control, status, configuration, or other purposes. Examples might include Representational State Transfer (REST) or other web-based APIs used for external control, Network control/status Interfaces, or VM interfaces.
- **Service/Traffic Handling** – These assets cover the basic services that the solution provides. Also, included in this category are traffic capacity and handling capabilities provided by the solution.
- **Encryption Keys** – These assets protect critical data, communications, or other assets via encryption.

- **Inter-Domain Administrative Boundaries** – These assets represent the legitimate procedural interface points between users, administrators, and service providers. Examples of potential compromise might include the disruption of these interactions or compromising information exchanges.
- **NFV Components** – This class of assets includes network provider, customer, and 3rd party Network Functions Virtualization components which make up or otherwise participate in the solution or service. Possible considerations include corruption, substitution, or steps that otherwise compromise the function or integrity of NFV components.
- **Customer Application Software** – These assets are provided by the end customers that participate in or use capabilities of the solution or service. Examples might include applications on mobile devices and enterprise applications in data centers.
- **Compute Environment** – These assets represent the computing and data networking capabilities that execute the solution software. These assets might include virtual as well as "bare metal" assets.
- **Endpoint Device** – These assets are devices that are part of the solution or service being provided or devices that directly interface with the same. Examples include mobile smart devices, in-home consumer devices such as set top boxes, and enterprise end-point devices.
- **Physical Assets, Power, Cooling** – These assets are critical components that are required for the solution or service to function. They may present single points of failure or be protected via redundant or load sharing mechanisms. Examples might include cell towers/antennas, power grid or emergency power devices, and cooling systems,
- **Staff** – This category covers damage to assets via attacks manifested through staff that develop, operate, or administer the solution in question. Staff attacks may range from: compromises by staff, such as indirect attacks causing data to be exposed or procedures improperly performed without their direct knowledge; or attacks made with consent, manifested by placing attackers in legitimate positions as "insiders" (i.e., Social Engineering Attacks). There may also be attacks directly aimed at legitimate staff with intent to cause them direct harm in order to directly or indirectly achieve the attacker's goals.
- **Brand and Reputation** – These assets represent the quality, reliability, trustworthiness, company valuation, etc., of the solution provider's services, personnel and business practices. Attacks aimed at this asset class seek to damage or destroy an organization's value and trustworthiness from the perspective of customers and/or stakeholders.

3.1.2.3 Mapping of Use Cases to Assets

This section contains some example of solution use cases which are likely to be encountered in ICT applications. These are analyzed in terms of the assets shown above. The examples consist of:

1. Service provider is building a private cloud to host a network functions virtualization (NFV)-oriented architecture.
2. Service provider is building a cloud infrastructure to offer a public cloud offering:
    a. Software as a Service (SaaS)
    b. Infrastructure as a Service (IaaS)
    c. Platform as a Service (PaaS)
3. Service provider is evaluating a public cloud provider to acquire:
    a. IaaS Capabilities
    b. PaaS Capabilities

Any use case functionality associated with the 3rd party components (identified as part of the security objectives step) should be included in the specification of the use cases, and the associated 3rd party assets should be clearly identified and designated as primary or secondary assets as appropriate. During the

analysis, questions regarding data protection, trust boundaries, developer/application and platform segmentation, and risk management responsibilities should be considered.

Category	Use Case	Potential Solution Assets														
		Inter-Component Comm.	Configuration Data	End User Data	APIs	Service/Traffic Handling	Encryption Keys	Inter-Domain Admin Procedures	NFV Components	Customer Application SW	End-User	Compute Environment	Endpoint Device	Physical Assets, Power, Cooling	Staff	Brand and Reputation
NFV-based Service Provider Networks	Mobile Core and RAN	S	S	P	S	P	S	P	S	P	P	S	P	P	S	P
	IP Transport	S	P	S	S	P	S	P	S	S	P	S	P	S	S	P

P = Primary Asset S = Secondary Asset

Figure 3.4 – NFV-Based Service Provider Networks – Use Cases and Starting Point Assets

Figure 3.4 covers use cases for NFV-based service provider network solutions. Some questions and issues to consider when identifying assets for these solutions include:

- Multiplicity of vendors, maturity of components, in-house development, risks in Virtual Machine (VM), and cloud infrastructure.
- *Combinatorial issues – no two NFV solutions are alike*.

Category	Use Case	Potential Solution Assets														
		Inter-Component Comm.	Configuration Data	End User Data	APIs	Service/Traffic Handling	Encryption Keys	Inter-Domain Admin Procedures	NFV Components	Customer Application SW	End-User	Compute Environment	Endpoint Device	Physical Assets, Power, Cooling	Staff	Brand and Reputation
Service Provider Delivered SW as a Service (SaaS)	Service Provider delivered Enterprise or Comm. Application	S	P	S	S	P	S	P	S	S	P	S	P	S	S	P
	IoT/M2M Application	S	S	P	S	P	P	P	S	P	P	S	P	S	S	P
	SECaaS Application	P	P	S	S	S	P	P	S	P	P	S	S	S	S	P
	Bundled 3rd Party SaaS Application	P	P	S	S	S	P	P	P	P	P	S	S	S	S	P

P = Primary Asset S = Secondary Asset

Figure 3.5 – Service Provider-Delivered Software as a Service (SaaS) Solutions – Use Cases and Starting Point Assets

Figure 3.5 covers use cases for service provider-delivered SaaS solutions including those involving Security as a Service (SECaaS). In many cases, SaaS solutions will be bundled with service provider network

services and a set of service provider or 3rd party provider applications. Some issues to consider when identifying assets for these solutions include:

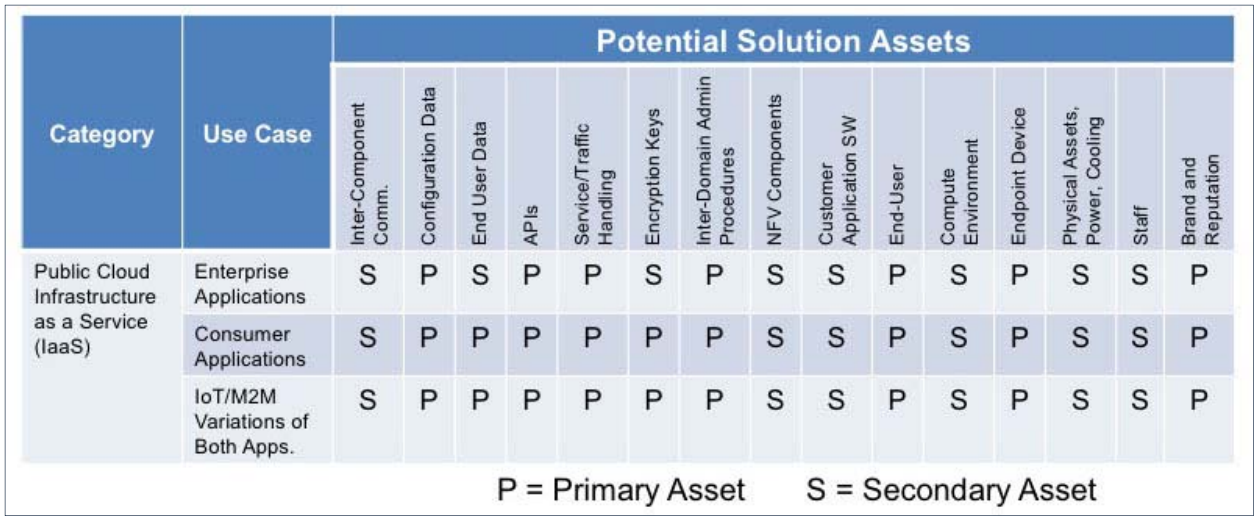
- These solutions are likely to have a larger attack surface as compared to those in other categories due to lower-level platform access; one needs to consider customer-customer boundaries and isolation more carefully.
- There may be a role for the service provider in protecting against threats generated by developers or 3rd parties.

Category	Use Case	Potential Solution Assets														
		Inter-Component Comm.	Configuration Data	End User Data	APIs	Service/Traffic Handling	Encryption Keys	Inter-Domain Admin Procedures	NFV Components	Customer Application SW	End-User	Compute Environment	Endpoint Device	Physical Assets, Power, Cooling	Staff	Brand and Reputation
Service Provider Public Cloud Platform as a Service (PaaS)	Enterprise Applications Bundled w/Network Svcs.	S	P	P	S	P	P	P	S	P	P	S	P	S	S	P
	Consumer Applications Bundled w/Network Svcs.	S	P	P	S	P	P	P	S	S	P	S	P	S	S	P
	Service Provider to Service Provider Applications (MVNO looks like a Service Provider)	P	P	P	P	P	S	P	S	P	P	S	P	P	S	P
	IoT/M2M variations of these	S	P	P	S	P	P	P	S	P	P	S	P	S	S	P
P = Primary Asset S = Secondary Asset																

Figure 3.6 – Service Provider Public Cloud Platform as a Service – Use Cases and Starting Point Assets

Figure 3.6 covers use cases for service provider-delivered Platform-as-a-Service (PaaS) via public cloud offerings. These use cases involve a service provider providing API access to network, transport, access, and/or higher-level communication services for use by enterprise or consumer-focused solution developers and providers. Some issues to consider when identifying assets for these solutions include:

- API-level access exposes additional potential for compromise:
  - New classes of large-scale attacks may be possible.
  - Greater risk of network overload or other compromise due to programmatic access to the network.



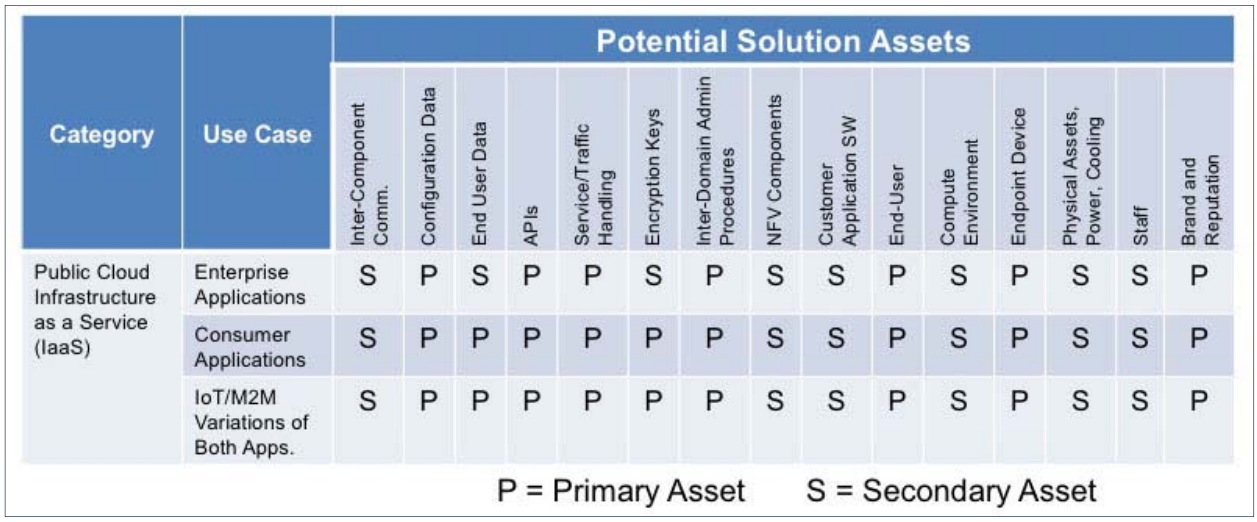
Category	Use Case	Potential Solution Assets														
		Inter-Component Comm.	Configuration Data	End User Data	APIs	Service/Traffic Handling	Encryption Keys	Inter-Domain Admin Procedures	NFV Components	Customer Application SW	End-User	Compute Environment	Endpoint Device	Physical Assets, Power, Cooling	Staff	Brand and Reputation
Public Cloud Infrastructure as a Service (IaaS)	Enterprise Applications	S	P	S	P	P	S	P	S	S	P	S	P	S	S	P
	Consumer Applications	S	P	P	P	P	P	P	S	S	P	S	P	S	S	P
	IoT/M2M Variations of Both Apps.	S	P	P	P	P	P	P	S	S	P	S	P	S	S	P

P = Primary Asset S = Secondary Asset

Figure 3.7 – Service Provider Public Cloud Infrastructure as a Service – Use Cases and Starting Point Assets

Figure 3.7 covers use cases for service provider-delivered Public Cloud Infrastructure as a Service (IaaS). These use cases involve a service provider that provides public cloud and related networking services for use by enterprise or consumer-focused solution developers and providers. IoT variations of both enterprise and consumer solutions are particularly important and are also included. Some questions and issues to consider when identifying assets for these solutions include:

- These use cases include service provider elements providing API access to network transport/access and higher-level communications services:
  - API-level access exposes additional potential for compromise.
  - New classes of large scale attacks may be possible.
- Greater risk of network overload or other compromise due to programmatic access to network.



Category	Use Case	Potential Solution Assets														
		Inter-Component Comm.	Configuration Data	End User Data	APIs	Service/Traffic Handling	Encryption Keys	Inter-Domain Admin Procedures	NFV Components	Customer Application SW	End-User	Compute Environment	Endpoint Device	Physical Assets, Power, Cooling	Staff	Brand and Reputation
Public Cloud Infrastructure as a Service (IaaS)	Enterprise Applications	S	P	S	P	P	S	P	S	S	P	S	P	S	S	P
	Consumer Applications	S	P	P	P	P	P	P	S	S	P	S	P	S	S	P
	IoT/M2M Variations of Both Apps.	S	P	P	P	P	P	P	S	S	P	S	P	S	S	P

P = Primary Asset S = Secondary Asset

Figure 3.8 – Service Provider Evaluating/Acquiring Public Cloud IaaS or PaaS Capabilities

Figure 3.8 covers use cases where service providers are evaluating or acquiring 3rd party public cloud capacity to provide capabilities to support solutions they deliver. These use cases involve the purchase of compute, storage, and/or networking capacity in the form of Platform-as-a-Service (PaaS) or Infrastructure-as-a-Service (IaaS) capabilities. Basic operating software and application libraries may also be included in what is provided by a 3rd party cloud provider. Some questions and issues to consider when identifying assets for these use cases and the associated solutions include:

- Resulting 3rd party control of data center, associated networking, and supporting operating software capability issues such as:
    - Indirect attacks via the 3rd party infrastructure, software, or personnel.
    - Potential security vulnerabilities in 3rd party provided operating or other software components.
    - Greater risk of computing or network overload or other load-related compromise due to a shared compute and networking environment.

It should be noted that use cases of this type rely heavily on a complete definition of the components, assets, and threats associated with the 3rd party Public cloud capabilities that are being acquired.

It is important to note the assets, as well as their classifications in terms of ***Primary*** vs. ***Priority Secondary Assets***, is intended to be a starting point to aid in the identification and final classification of the assets for the solution to which the ARA Process is being applied.

3.1.3 Network Diagrams

3.1.3.1 Diagram Utilization

The architectural diagrams provide a framework to visually depict the physical and virtual network elements that make up the delivery of service provider-delivered services, from hosted applications to end devices. The interrelationships and linkages of these network elements can highlight potential vulnerabilities to security threats. These threats are not always clearly evident with the introduction of new operational models that are implemented through the deployment of virtualized architectures. Therefore, the architectural diagrams also show the location, both physical and logical, of the assets. These assets have a topological and logical relationship with the use cases highlighted within the document.

The architectural diagram (Figure 3.9) highlights a service provider virtual architecture for applications provided to end devices. The architecture interfaces to and interconnects with both service provider-hosted and non-service provider-hosted facilities. The virtual service provider network view shows the core and access networks as virtual implementations of an LTE/4G architecture. The virtual core network would include the functions of an Evolved Packet Core (EPC). The virtual access network would include the functions of a Radio Access Network (RAN). Virtual Network Functions (VNFs) are interconnected via intra-VNF interfaces and also interconnected to the service provider virtual network. The management and control of these virtual functions are done by the management and orchestration functions.

The hosted facilities have several key attributes for applications development, management, and service enablement. These are:

- Network access for interconnection.
- Network services that provide QoS, firewall, intrusion detections, and other features.
- A software environment for an application hosting environment.
- Application software services for the administration and management of application delivery.

The end device is represented with the following components:

- “Network Access” which provides physical connections.

- "Network Services" which provide for authorization, bandwidth management, and network security functions.
- "Application Software" which provides application features and functions that ensure integrity of the application running on the end devices.

It is also essential that all of the 3rd party components and associated assets that are part of the solution (as identified in the security objectives and use case steps) be clearly identified on the architectural diagrams as they are developed.

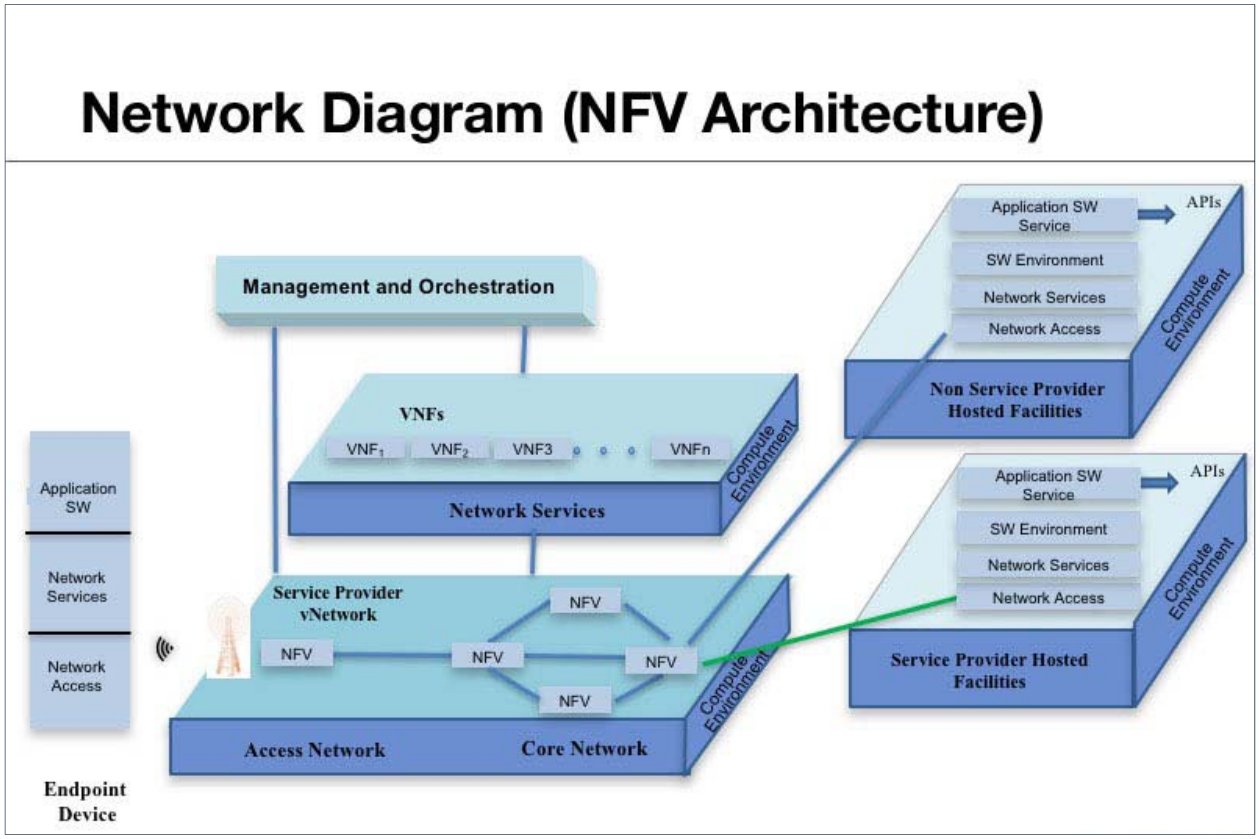


Figure 3.9 – NFV Architecture Network Diagram

3.1.3.2 Network Diagram for Hosted Use Cases

The use cases for the IaaS, PaaS, and SaaS are reflected in the network diagram shown in Figure 3.10.

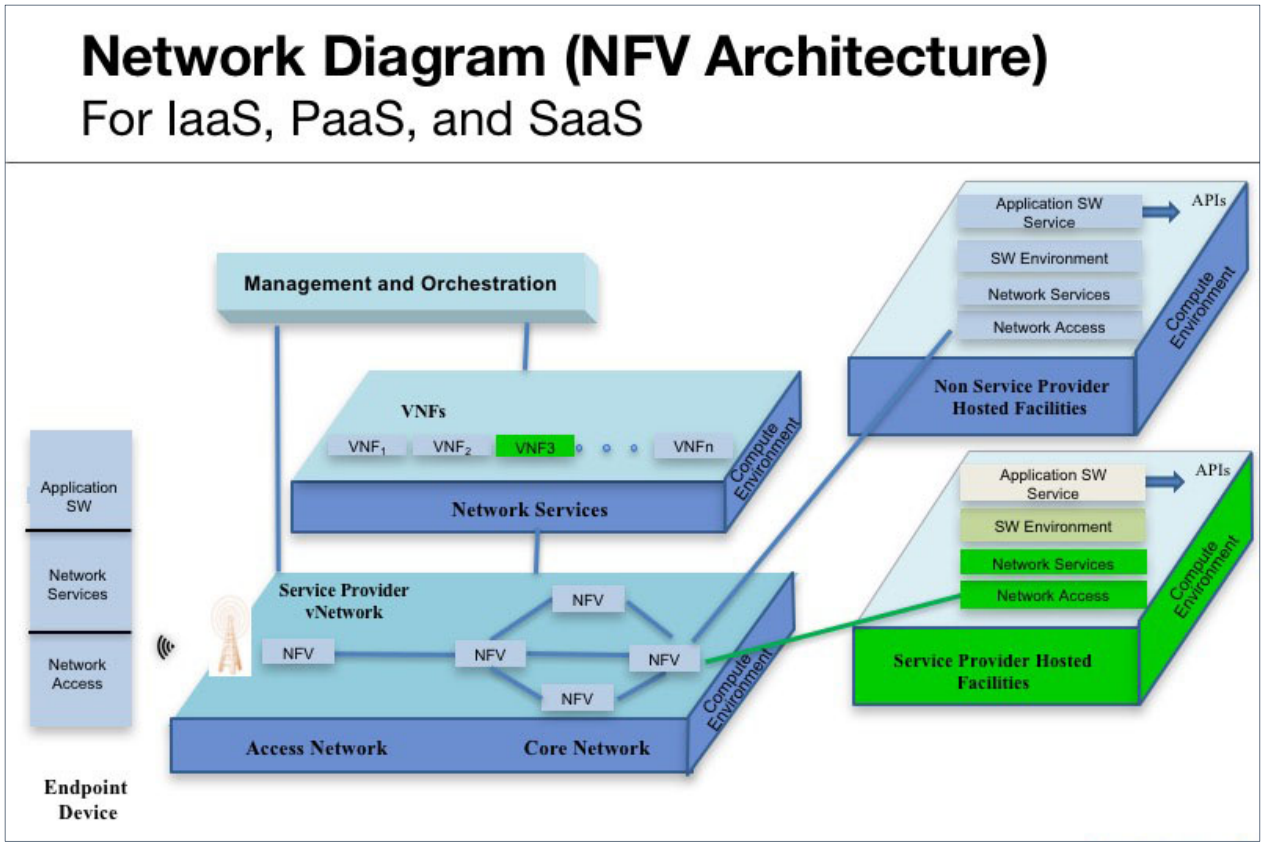


Figure 3.10 – Network Diagram for IaaS, PaaS, and SaaS

The diagram shows:

- IaaS as a service provider-hosted facility including the compute environment (servers, storage operating system), network access, and network services. These are shown in bright green.
- PaaS, as a service provider-hosted facility, including the compute environment, network access, network service, and a hosted software environment. This software environment allows for management of the application software to be done by the end user. The software environment is shown in pale green.
- SaaS allows for hosted application delivery and includes the network and software environment features needed to enable this service. The management of the application(s) is done by the management of the hosted facility. The application software service is shown in white.

3.1.3.3 Asset Mapping

The assets identified within the use cases modeled in this document can be associated with different parts of the service provider's network. This includes both physical and virtual components of the network as well as those associated with any 3rd party components previously identified. The use of network diagrams allows for an understanding of where these assets may reside. The assets depicted in the diagram (Figure 3.11) are a representative sample but this is not an exhaustive examination. Varying network topologies, architectures, and operational deployments will impact where these assets reside in the network.

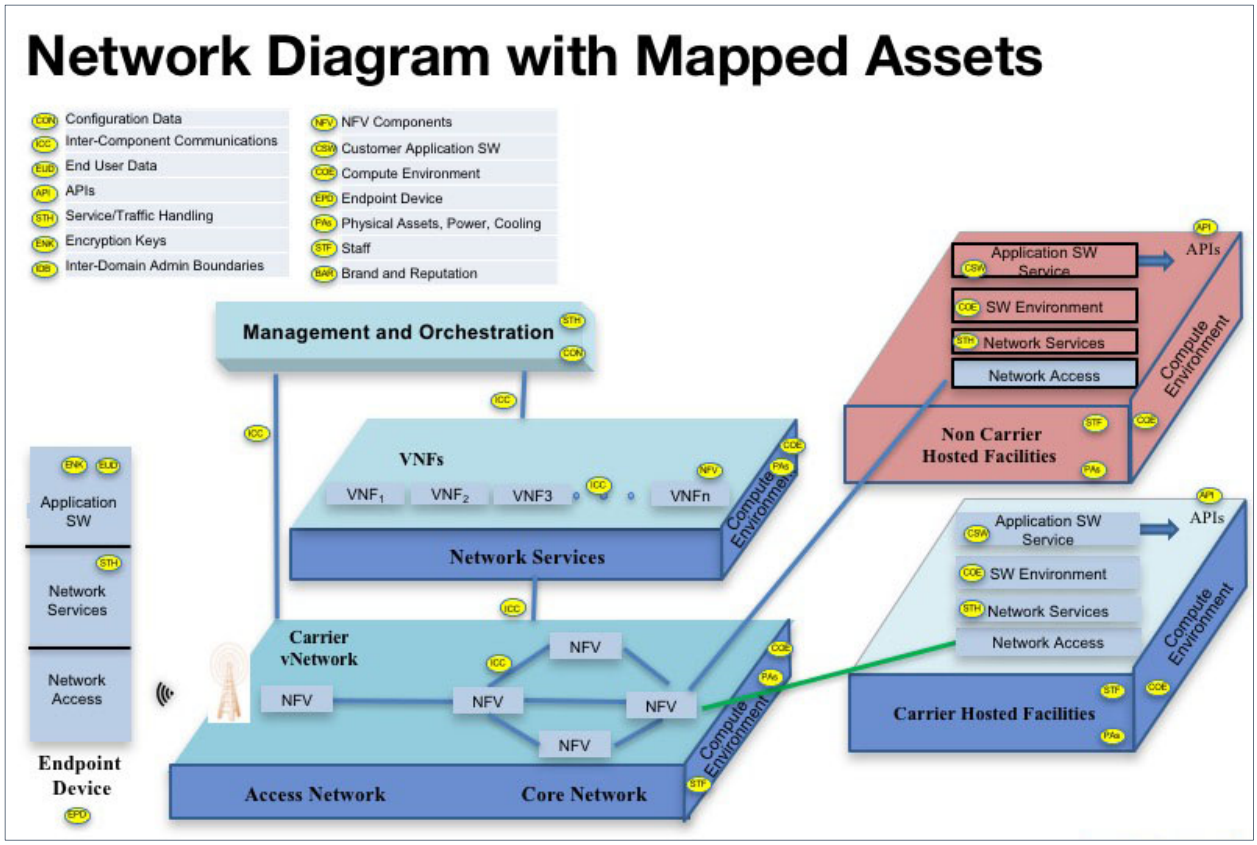


Figure 3.11 – Network Architecture Diagram with Asset Identification

Though each network deployment may vary, the assets shown may reside in areas within or outside of the portions of the solution directly controlled by the solution developer. The network architecture diagram is also utilized for the determination of physical or logical boundaries (i.e., *Trust Boundaries*) between solution owner provider and 3rd party components. The network architecture diagram is then utilized for threat modeling to reflect the relationship of the network components, including 3rd party components.

In Figure 3.11, the 3rd party components are designated in a highlighted color as part of the carrier hosted facilities. This notation may also be used to highlight 3rd party components within carrier-hosted facilities. Since the 3rd party components may not be fully trusted per the *Trust Models* defined in clause 3.1.1, assets identified within the 3rd party environment would be associated with appropriate risk factors as part of the threat analysis step. Identifying particular assets as *3rd party provided* allows for appropriately targeted threat assessment and mitigation.

3.2 Threat Identification

Threat identification and the associated risk assessment steps are critical parts of the ARA Process. This clause describes how the threat modeling and associated risk assessment steps are performed via a structured approach that systematically defines the range of potential threats, prunes the potential threats based upon likely threat actors, and then provides mechanisms for weighting the resulting threats.

An important aspect in this phase of the ARA Process is the development of abuse cases that realize the threats identified during the threat modeling process. Abuse cases must properly cover the threat landscape and must be mapped onto potential attack points identified in the architectural diagramming step in the ARA Process.

Both the threat modeling and abuse case steps must thoroughly consider threats that may manifest themselves through 3rd party components within the context of the previously defined *Trust Models* for these components.

3.2.1 Threat Model

3.2.1.1 Threat Modeling Methodology

Proper risk assessment and management requires the identification and understanding of threats as its starting point. There are many approaches or "threat models" in existence today, proposing various methods of exhibiting threats via a set of step-by-step procedures, catalogue-based techniques, or other means. No matter what approach is taken, a concise threat model supports the risk manager in understanding the nature of threats.

The threat model described here adopts an *asset-centric* approach coupled with an *adversary-centric* approach that characterizes the threat agents and their defining attributes, and thus their danger to their target, whatever that target might be.

The adversary-centric approach takes into account the adversaries, their motivations, their willingness to incur risk, and their likely targets. This model is based on the Intel Threat Agent Library (TAL) (Casey, 2007), and is explained in Clause 3.2.1.3.

It is important to consider that today's adversary-based threat represents a spectrum of activities that resemble a military operation in some respects. The figure below illustrates this spectrum of tactics, techniques, and procedures the adversary can take.

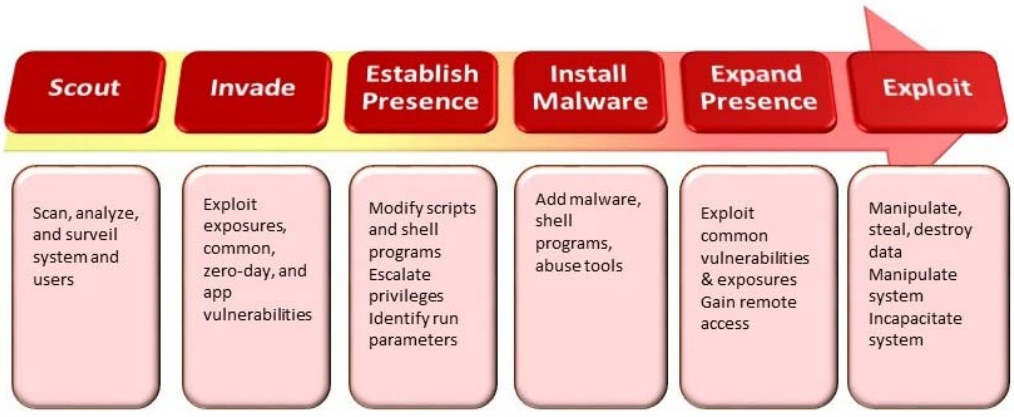


Figure 3.12 – Threat Activity Spectrum

The figure shows the pre-attack reconnaissance, initial interactions with the target, the processes of establishing a presence on or within the target, and the exploitation phase. Depending on the furtiveness of the attack, skill of attackers, and nature of the exploitations, an attack can remain active for weeks or months before it is discovered or the attacker "pulls the plug" on the exercise. Some attacks are never discovered.

The asset-centric approach focuses the threat assessment on those assets—data, systems, personnel, and so on—deemed critical to a system's or program's smooth and correct operation. By viewing each

critical asset as a potential target of attack, and then assessing the broad categories of attack (*attack classes*) against that asset, a preliminary estimate can be built of[1] the security defenses needed to protect each asset. In assessing the full spectrum of primary and supporting assets (and the avenues along which they can be reached for exploitation), and combining this analysis with the threat agent model, security risks can be addressed at an architectural level by building security defenses up front into applications, processes, and execution environments. This amounts to examining threats and identifying possible mitigations at the design and engineering stages of the solution, with the goal of building the defensive mechanisms into the architecture, processes, products, and software code as it is written or procured rather than patching it later. This method of threat modeling takes on a *defender's perspective*, rather than a reactive approach to cybersecurity.

The goal of generating both asset-centric and adversary-centric approaches is to produce a threat model that posits the most likely threats to the target assets based on the attackers' likely motives and capabilities, and on the assets' most threatened attributes. The threat model is then positioned to provide the following values:

1. It is an indispensable input into risk analysis. The individual threat vectors can be paired with corresponding loss valuations associated with each asset at risk to render an overall risk rating.
2. It serves to help decide what security measures are needed at given points within the solution architecture, and suggests a priority order in which those measures must be planned and incorporated into the overall security solution.

Non-technical solutions to threats, such as attacks on personnel and campus facilities, can also be mounted on the basis of the threat model, which highlights the likely threats in those (and other) areas.

The threat model is based on the following assumptions taken as axioms:

- Any solution or organization has assets of value worth protecting;
- These assets have vulnerabilities;
- Internal or external threats can exploit these vulnerabilities to harm the assets, expose them, or steal them; and
- Appropriate security countermeasures exist and can be introduced to mitigate the threat's associated risks.

Of these four statements, the last is the most tenuous – and the most important since it declares a path forward to neutralize the perceived threats and reduce the number and severity of the assets known vulnerabilities. This assumption is the driving force behind the threat model.

The remainder of this clause establishes the asset and adversary perspectives.

3.2.1.2 Asset-centric Methodology

An *asset* can be any object having value to someone: data, systems, devices, support infrastructure (e.g., power, water, HVAC[2], networks, telecom) people, structures, and facilities (e.g., operations centers; remote, unstaffed relay stations), corporate brand and reputation, and many other entities. An asset of any value is bound to be coveted by someone, or targeted for mischief or malice. In order for the asset to be

[1] To establish a precise set of security defenses requires factoring in the vulnerabilities, the probabilities of an attack being carried out, and the consequences of the attack. The threat model plus a risk assessment (not discussed here) are both required elements of an analysis that covers all of those factors.

[2] Heating, Ventilation, and Air Conditioning.

successfully attackable, it must have one or more characteristic that can be exploited or compromised in a way that will benefit the attacker, harm the asset user, or both.

Money, by way of example, is a valuable asset because it is portable, available to the user, acts as a proxy for transfers of property having dissimilar value between people, and has the backing of a trusted institution, which provides a consistent valuation of the money and assures both parties that a transaction involving money will not go awry. The characteristics that make it vulnerable are its inherent value (to support its function as a medium of exchange), its consistency (which permits it to be saved, retrieved, and exchanged at a later time with predictable usefulness when retrieved), and its relative stability (which allows its users to make sense of its value and performance as a proxy for goods in business transactions). It is therefore susceptible to theft, devaluation (through counterfeiting, for example), and attacks intended to destabilize it.

Common assets include:

1. Data
2. Systems
3. Functionality
4. Applications
5. Devices
6. Staff
7. Facilities
8. Brand and Reputation

The first five are specific to Information Technology (IT) while the last three have relevance beyond IT. They are included to indicate that the threat model methodology is extensible beyond the traditional IT realm. The list above is thus representative rather than comprehensive.

Each asset has characteristics, referred to hereafter as *attributes*, which make them vulnerable to attack. The assets mentioned above have the attributes given in Table 3.2 below.

Table 3.2 – Assets and their Attributes

Asset	Attributes
Data	Availability, Authenticity, Confidentiality, Integrity, Non-repudiation
Systems	Availability, Authenticity, Correct operation, System integrity, Data integrity
Functionality	Authenticity, Availability, Integrity
Applications	Availability, Authenticity, Integrity, Consistency of operation
Devices	Availability, Authenticity, Privacy, System integrity, Data integrity
Facilities	Protection, Availability, Integrity, Resources
Staff	Knowledge, Skills, Integrity, Availability
Brand and Reputation	Integrity, Marketability, Influence

When defining an asset, a key step is the correct determination of the asset's attributes, upon which this threat model methodology depends. Some attributes, such as those associated with data, are fairly well

known while others must be assembled. Deriving the attributes must be a thoughtful and well-informed effort if the modeling is to be effective.

3.2.1.2.1 Description

The steps of the asset threat model methodology are as follows:

1. **Determine the asset or assets to be modeled.** The method of determining them can vary, but it is a good idea to select them on the basis of how critical they are to the program or asset user. Many IT programs might say, for instance, that the protection and retention of data are their reason for being, and would therefore point to the data as the most important asset. Others might point to a capability to mine data in ways that give them an advantage over their competitors, and might therefore identify the proprietary software that mines that data as their most important asset. Still others might point to some other asset or a number of assets as having primary value to them or their customers, with other assets also having a high, though perhaps secondary, value. The chain of reasoning can extend to tertiary values, quaternary values, and so on; though overcomplicating the analysis by introducing too many assets can render the process ineffective.

2. **Establish the *attributes* of the asset.** By focusing on specific attributes that can be attacked to undermine the asset in some way, insight is gained into what constitutes a viable risk to the asset. The threat model methodology focuses on establishing the risks to each attribute and the threat agents capable of carrying out attacks that would degrade or remove one or more of those attributes.

3. **Determine the risks associated with each attribute.** Concentrating on the risks to each attribute gives deeper insight into the risks to the asset itself. “Risk” in this context is meant to suggest susceptibility to one or more threats, in the sense of a combustible asset being “at risk” to the threat of fire. In this sense of the word, “risk” does not imply a determination of a value such as would be derived from a quantitative risk assessment that requires calculations of the two components of risk: the magnitude of the potential loss and the probability that the loss will occur. The reader will need to keep this definition in mind during discussions of the threat model, to avoid misinterpretations. As an example of this, consider the confidentiality attribute of data. The obvious risks to confidentiality are loss through disclosure and theft[3]. Assessing the risks to the attributes instead of the asset itself gives a more focused perspective, enabling a more accurate assessment.

4. **Map each risk to one or more attack classes through which the risk is realized as a threat to the asset.** Rather than try to map each risk to an exhaustive set of actual attacks, the threat model maps the risks into *attack classes* – categories of attacks having common characteristics. Attack classes associated with theft of data, for example, include elevation of privilege (a common outsider tactic), abuse of access rights (a frequent insider attack), and logic bombs inserted into the code somewhere in the solution’s supply chain. By mapping to attack classes rather than to specific attacks, a set of defenses can be established to address broad categories of attack and in doing so adopt a more strategic approach to security. Attack classes are general rather than specific (e.g., “worm” rather than “attacker injects a worm through unsecured USB port”). Each of the attack classes can be further expanded into countless subcategories, types, and varieties of attack. By focusing on the broader concept of attack classes, it is recognized that while it is impractical to delineate the entire universe of risks, broadly categorizing them demonstrates the need to mount security countermeasures to address the broad risk via a defense-in-depth strategy. It further

[3] Although disclosure is obviously accomplished by stealing data, the difference between theft and disclosure lies in the motive behind the attack: disclosure implies that the stolen data is revealed to a larger audience than its owners intended (e.g., disclosure of a list of spies to the nation being spied on), while theft goes more towards surreptitious or threatened use of the data (e.g., theft of the list of spies as a prelude to spying on the spies themselves; or removing them from operation).

suggests the areas at which a pinpoint treatment might be called for. To help make that determination, attributes of the asset need to be ranked from most important to least. Doing so permits concentration to be focused on the highest risks.

5. **Map attack classes to specific attacks, as needed.** Listing specific attacks to the attack classes can help build a *threat library* for the asset. Threat libraries can subsequently be applied to, or used as input to, threat models for similar assets, or for classes of related assets.

The generic threat model is illustrated in Figure 3.13 below. This *target tree* shows the attributes of the asset, the risks to each attribute, and the primary attack classes through which each risk morphs into an actual threat to the asset.

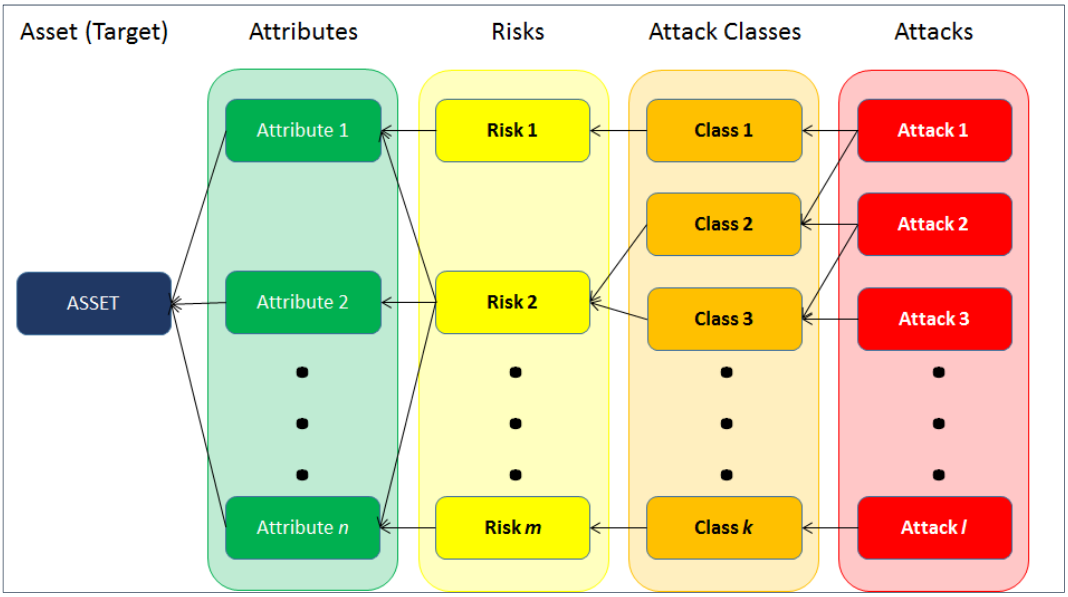


Figure 3.13 – Schematic of Asset Threat Model

The items in green indicate the attributes associated with an asset. All security risks (shown in yellow) to the asset affect one or more of these attributes. The orange items represent attack classes while the red items list actual attacks. In actual use, an attack tree of this sort must display at least one risk for each attribute, and at least one attack class for each risk. The inclusion of attacks for each attack class is not necessary as long as it is understood that no attack class is a null set; i.e., there is at least one attack for each attack class.[4]

In some cases, a risk can be manifested through several more narrowly defined risks. Denial of service, for example, is a general category of risk that can be further subdivided into deletion, ransom, and physical destruction. In an entirely analogous fashion, a broad attack class can be subdivided into smaller, more specific attack classes. As an example, physical attacks on an asset might be further divided into direct attacks (e.g., explosives) and environmental attacks (e.g., electrical sabotage, HVAC sabotage). Finally, attacks themselves can sometimes set the stage for other attacks or create conditions that magnify the effect of an attack. An attack on a facility's sprinkler system, for instance, might be an attack in and of itself

[4] Clearly, an attack class for which no actual attacks exist presents no danger to the asset.

(activation of the system to cause flooding) or it might precede an arson attack (deactivation of the system, to remove its fire suppression objective). This concept is illustrated in Figure 3.14 below.

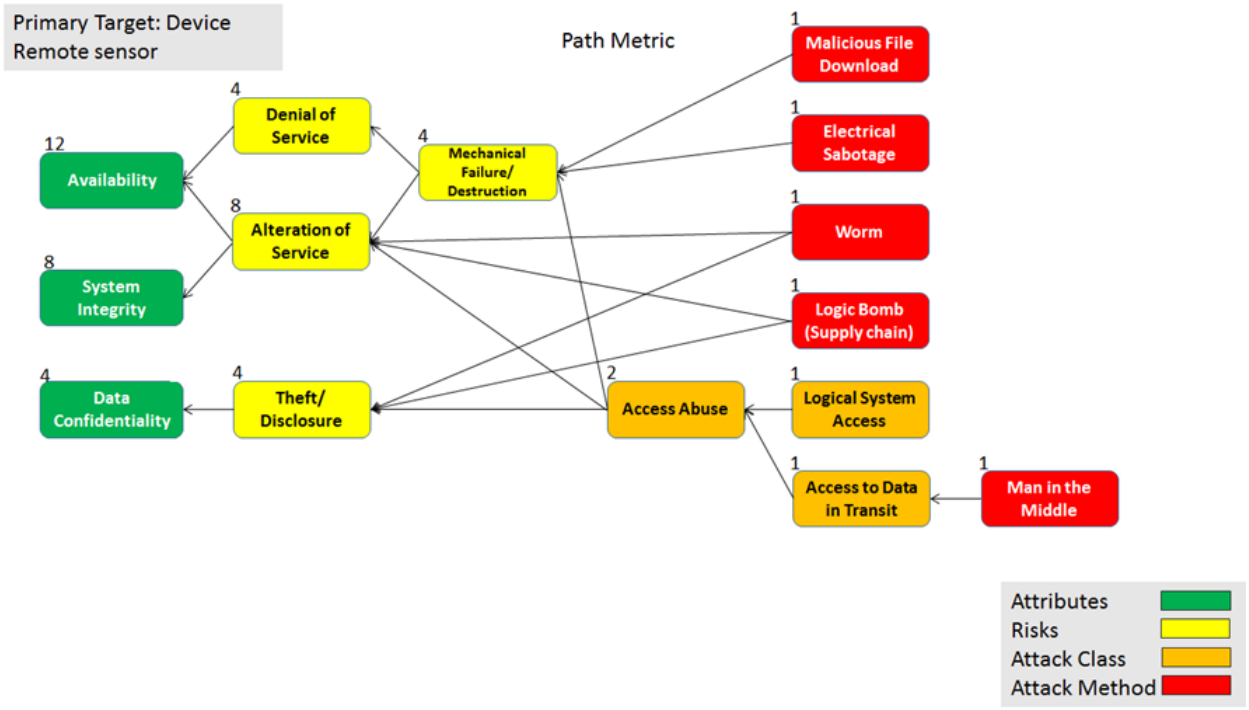


Figure 3.14 – Detailed Schematic

The asset threat model is usable for a broad class of assets, though discussion is focused on its applicability to information technology. The explanations are thus directed toward IT-related assets such as data, information systems, applications, and other related entities. Next, an example is presented of the model in play, with data as the asset under scrutiny.

3.2.1.2.2 Example: Data as an Asset

With respect to the asset-centric component, a threat model that posits data to be the most critical asset is among the easiest to build, at least at a high level. Given that the security attributes commonly associated with data are its availability, authenticity, confidentiality, integrity, and non-repudiation, the threat model focuses on establishing the risks to each and the threat agents capable of carrying out attacks that would degrade or remove one or more of those attributes. To do this, a set of risks is mapped to each of the five attributes of data listed above. Availability of the data to authorized users, for example, can be lost through tampering and denial of service attacks; while confidentiality can be lost through theft or disclosure of the data. Tampering and denial of service are thus the risks (in the sense described above) to data availability, while theft and disclosure are the risks run when trying to maintain data confidentiality.

The threats to each of the five data attributes are given in the table below.

Table 3.3 – Data Attributes and the Risks to Each

Attribute	Risks
Authenticity	Falsified data source
Availability	Tampering, denial of service
Confidentiality	Theft, disclosure
Integrity	Tampering
Nonrepudiation	Invalidation, breach of policy

Next, each risk is mapped to one or more attack classes through which the risk is realized as a threat to the data. Attacks associated with theft of data, for example, are elevation of privilege (a common outsider tactic), abuse of access rights (a frequent insider attack), and logic bombs inserted into the code somewhere in the solution's supply chain.

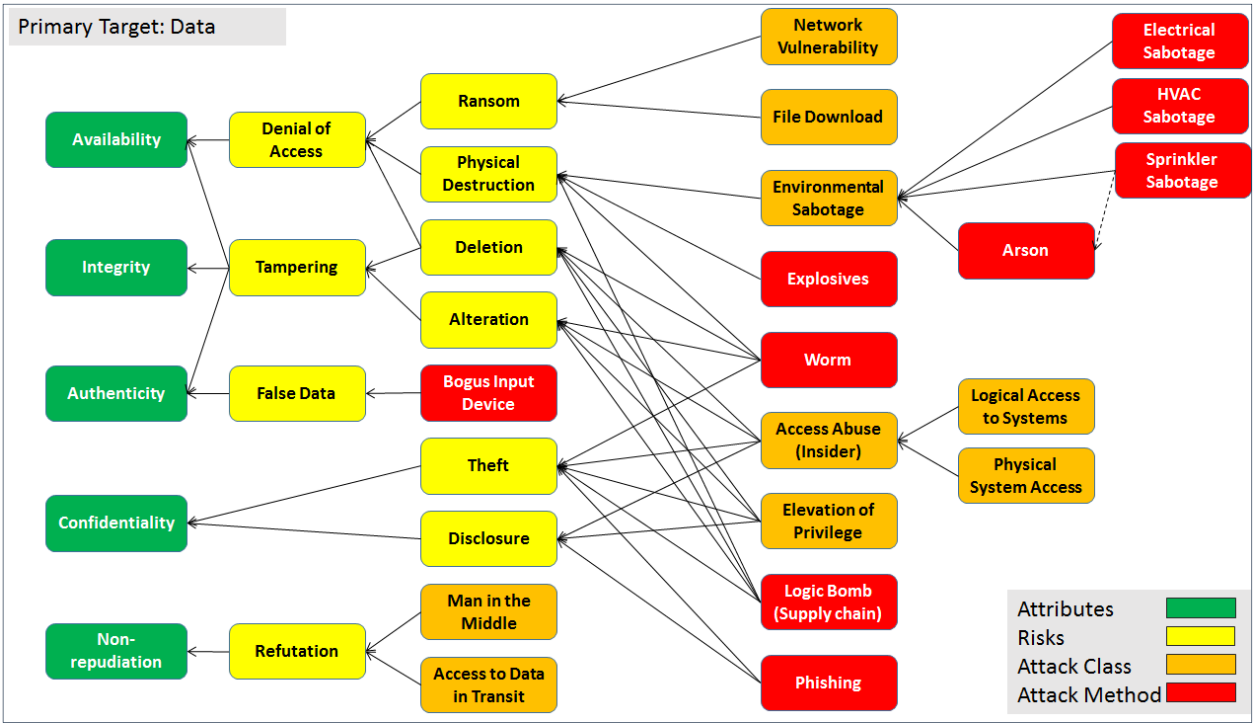


Figure 3.15 – Data Target Tree

The threat model for data is illustrated in Figure 3.15 above. This *target tree* illustrates the attributes of the data, the risks to each attribute, the primary attack classes through which each risk morphs into an actual threat to the data, and specific attacks associated with some attack classes. The dotted line pointing *Sprinkler Sabotage* to *Arson* shows an example of the indirect attack mentioned earlier.

In order for this exercise to have more than abstract academic value, it needs to be tied to a business model or some other real-world concept, so that the results of the threat modeling effort have real value to the involved parties. This is started by attaching some *ranking values* to the asset's attributes.

3.2.1.2.3 Ranking of Attributes

A value of delineating an asset's attributes lies in being able to determine the importance of each attribute to the larger business model or to the asset's *raison d'être* and thus, how to prioritize it in the allocation of resources for risk management. In ranking the attributes, it is important to consider both the value and the purpose of the asset. So, for an asset having *n* attributes, each attribute is given a number between 1 and *n*, inclusive; and it is required that no two attributes have the same rank. The attributes are ranked from least important (a rank of 1) to most important (a rank of *n*). Table 3.4 shows an example, again using data as the asset and illustrating a possible ranking of its five attributes ($n = 5$).

In this example, the asset is a dataset that details a corporation's 5-year market share strategy, which is highly sensitive and will be the cornerstone for that strategy. The table shows a configuration in which confidentiality is considered the most important attribute, and is given a rank of 5. In this ranking, integrity is ranked next because of the value of the data to the corporate strategy, then availability follows. Authenticity and non-repudiation are ranked lowest because in this case the sources of the data are considered unimpeachable.

Table 3.4 – Ranking of Data Attributes

Attribute	Rank
Availability	3
Authenticity	2
Confidentiality	5
Integrity	4
Nonrepudiation	1

Ranking the attributes identifies which branches in the attack tree (Figure 3.15) to consider most carefully when fleshing out the threat model. In order to attach meaning to that, a set of metrics must be defined that will give insight into the relative seriousness of each threat (with respect to business need) and the degree to which each attribute is threatened.

3.2.1.2.4 Metrics

The objective in defining metrics is to take some of the guesswork and "gut feel" out of the threat assessment process. The metrics defined here do not need to be (and are not) complicated; they do, however, need to supply a consistent recipe for quantifying the modeling process to aid prioritization. Two metric quantities are defined and applied to the process. The first measures (in some sense) the relative "weight" or severity to the asset of a given attack or attack class. The second counts the number of paths from each attack (or attack class) to each attribute. For convenience of discussion, they are labeled the *weight metric* and the *path metric*, respectively.

The **weight metric** is based on the ranking of the asset's attributes. Its purpose is to indicate the most dangerous threats to the asset, to a first approximation. For each node, the weight is calculated by adding the weights of the input nodes (i.e., the nodes whose arrows point to the node of interest), moving from the root to the leaf nodes (i.e., from left to right in the target trees depicted in this paper) through the target tree.

The **path metric** is based on the number of nodes and paths traversed in tracing each threat (or threat class) to the attributes it endangers. For each node, it is calculated by adding the weights of the input nodes, moving from the leaf nodes to the root (from right to left in representative figures) through the target tree. The leaf nodes representing the threats or threat classes are arbitrarily given a starting value of 1.

The easiest way to illustrate these processes is by example.

3.2.1.2.5 Example: Remote Sensor

Imagine that an asset in the form of a wearable remote sensor collects location and health data on a human or animal. It is battery-powered, remotely programmable, and has a two-way radio over which it sends its data to a home station and accepts reprogramming instructions (when needed) from the home station. The health and location data need to be confidential. The sensor must be reliable and operating at all times. The sensor needs to be able to send and receive data at all times (i.e., it needs it to be logically, continuously available). Based on this information, and other information available, it is determined that the asset to be modeled is the sensor itself, and the attributes of the sensor are *Availability*, *System Integrity*, and *Data Confidentiality*. The risks associated with each attribute are determined and tabulated below.

Table 3.5 – Data Attributes and the Risks to Each

Attribute	Risks
Availability	Tampering, denial of service
Data Confidentiality	Theft, disclosure
System Integrity	Tampering, alteration

Mapping each risk to attack classes and attacks comprises the attack tree shown in Figure 3.16 below.

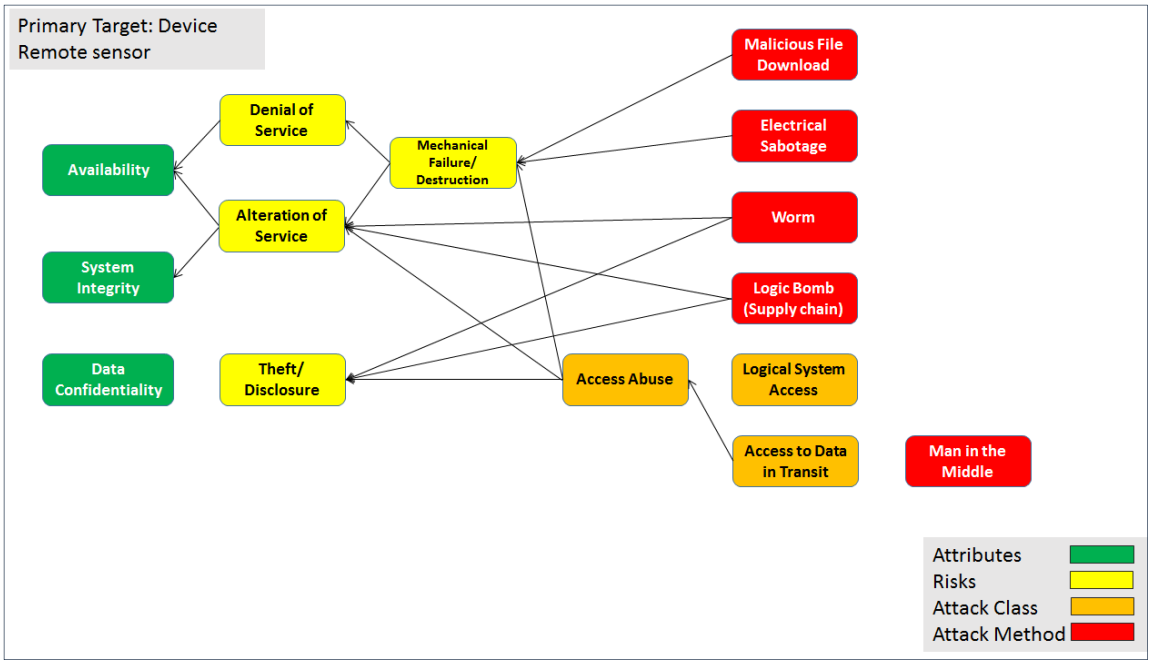


Figure 3.16 – Attack Tree for Remote Sensor

Data Confidentiality is ranked highest (3), *System Integrity* the next highest (2), and *Availability* last (1), reasoning that protection of the host animal is the most important factor and that *System Integ*rity of the device outranks *Availability*. This ranking is based on the potential for the device to begin operating erratically, thus it would no longer matter how available it or its data might be. The rankings are shown in Table 3.6.

Table 3.6 – Attribute Ranking of Remote Sensor

Attribute	Rank
Availability	1
System Integrity	2
Data Confidentiality	3

Next, the weight metric is evaluated. To do so, each node of the graph is tagged with a number to clarify the counting process. The attributes are labeled with their respective rankings from Table 3.6. These are shown as superscripts at the top-right corner of the three attribute boxes in Figure 3.17 below. Next, label the risk boxes in the same manner by adding the rankings of all attributes that point to them. In a similar manner, labeling proceeds from left to right until all nodes have been labeled with a number, as shown in the figure.

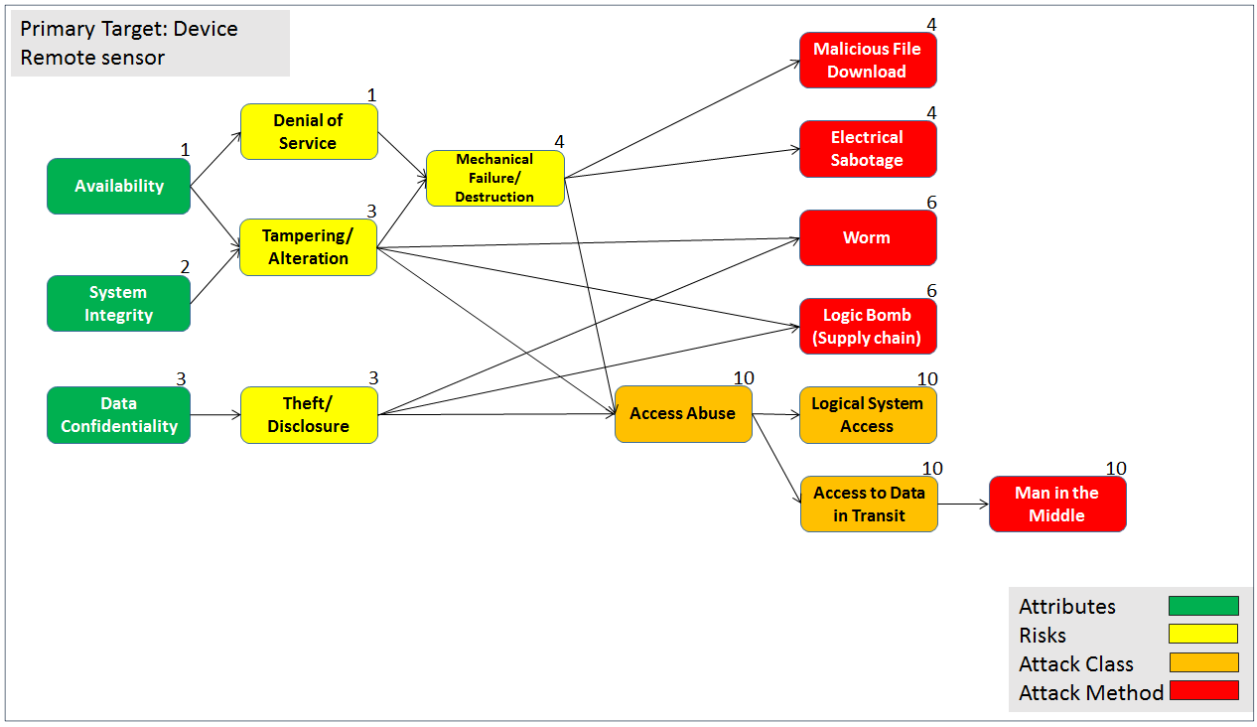


Figure 3.17 – Weight Metric

This metric shows the relative weight[5] of each attack class (and associated attacks) based on the ranking of the asset's attributes. The higher numbers indicate a higher level of danger, suggesting where security countermeasures are more direly needed.

Because the weight metric is a set of numbers, one for each leaf node, it can be written out in tabular form, an ordered *n*-tuple, or some other vector form, depending on need. In tabular form, it looks like this:

Table 3.7 – Example Weight Metrics

Attack (Class)	Weight
Malicious File Download	4
Electrical Sabotage	4
Worm	6
Logic Bomb	6
Logical System Access	10
Man in the Middle	10

[5] Clearly, a different initial ranking of the attributes would result in a different distribution of the weights, which suggests the possibilities of more general analyses based on variable attribute rankings.

The path metric – the number of paths from each attack (class) to each attribute – is as easily computed. To generate it, each attack or attack class (if it is a leaf node) is arbitrarily given a value of 1, as shown in Figure 3.18 below. These are shown as superscripts at the top-left corner of the six leaf nodes on the right-hand side of the figure. Next, label the attack class boxes in the same manner as before by adding the rankings of all attacks that point to them. Proceed from right to left until all nodes have been labeled with a number. The resulting superscripts on each attribute node indicate the number of attack paths that end at this attribute; the higher numbers suggesting the most endangered attributes.

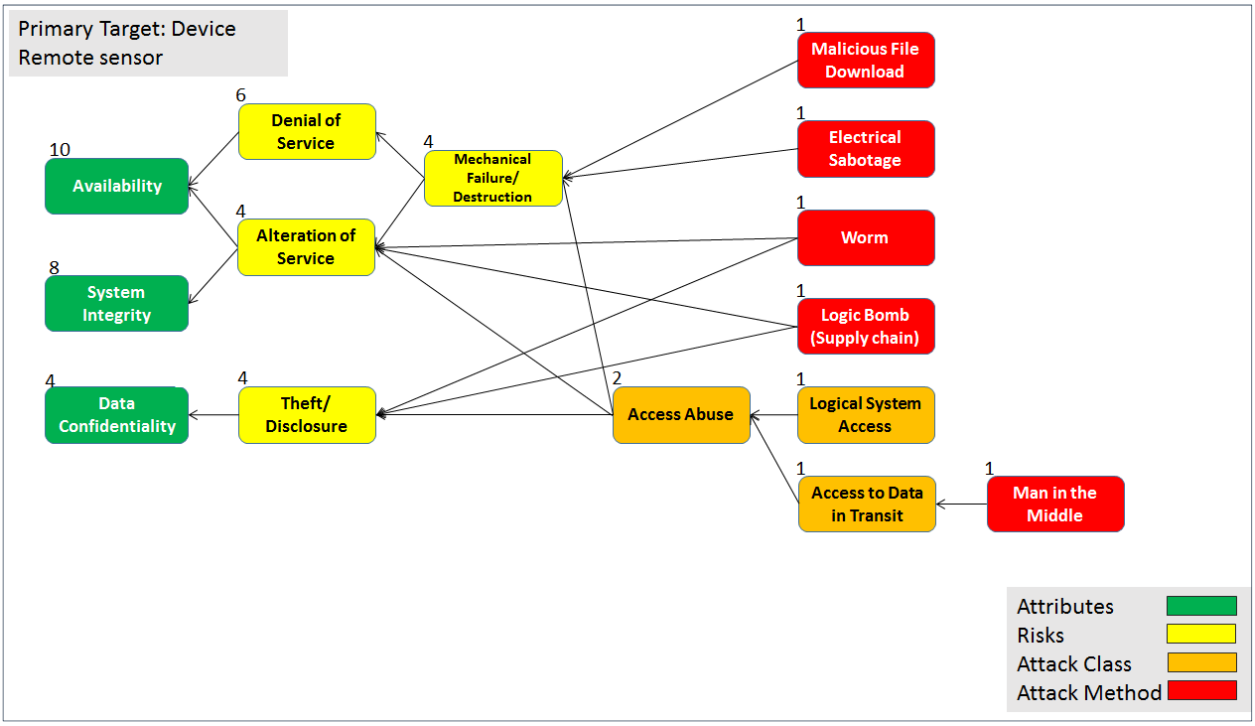


Figure 3.18 – Path Metric

Like the weight metric, the path metric can be written out in tabular or other forms. It can also be combined with the attribute ranking table as is done here:

Table 3.8 – Example Attribute Ranking

Attribute	Rank	Path
Availability	1	10
System Integrity	2	8
Data Confidentiality	3	4

The numbers suggest that *Availability* is the most threatened attribute, which might suggest re-considering earlier analysis, given that it was ranked as the least important attribute.

Some options at this point are to:

1. Recognize that *Data Confidentiality*, the preferred front-running attribute, is the least threatened and no further action is required.
2. Reconsider the initial ranking and recalculate the weighting metric to see how much it changes the concentration and distribution of countermeasures.
3. Re-examine the threat model to see if the numbers suggest a need for refinements or corrections. (The attack tree for the remote sensor in this example underwent a number of simplifying refinements as a result of analyzing the "whys" of the metrics.)

In re-examining the threat model, it is important to resist the impulse to simplify the attack tree so much that it ceases to represent the asset fairly.

It is also important to bear in mind that the asset-centric approach is only part of the overall threat modeling methodology. The adversary-centric component, which is instrumental in helping "prune" the attack tree by providing context on the adversary space, is addressed later in this document.

3.2.1.3 Adversary-centric Methodology

In addition to assets, this threat model takes into account the adversaries, their motivations, their willingness to incur risk, and their likely targets. This approach is based in part on the Intel Threat Agent Library (Casey, 2007), which has been adopted by the U.S. Department of Homeland Security.

The adversary aspect of the threat model is based on the fact that an adversary is constrained by the resources at their disposal (e.g., financial backing, equipment), their access, expertise, and ability to tolerate risks to themselves or their objectives. An adversary will follow four steps to achieve a successful attack:

1. Identify the objectives of an attack.
2. Identify the target to be attacked, and the type of attack that will bring about the desired objectives.
3. Gain access to the target at a level appropriate for the attack.
4. Carry out the attack. This can involve altering the target in ways that cover the evidence of the attack.

Interruption of or failure to complete any of these steps will cause the attack to fail. Therefore, the goal of a security architecture is to prevent the adversary from completing at least one of them. A secondary goal is to do this in the most efficient and cost-effective manner.

3.2.1.3.1 Description

The factors listed above – expertise, access to the target, backing (e.g., money), and tolerance to risk – are representative of eight general *defining attributes* a threat agent displays to one degree or another. Table 3.9 lists those attributes, along with categories that characterize each. The defining attributes help eliminate irrelevant threat agents by discounting those whose motives (e.g., access to the target, skill needed to affect an attack) do not qualify them as credible or capable attackers.

Table 3.9 – Threat Agent Attributes

	Defining Attribute	Category
1	Intent	Hostile or non-hostile

	Defining Attribute	Category
2	Access	Internal or external
3	Desired outcome of attack	Acquisition or theft, business advantage, damage, embarrassment (to attacked party), technical advantage.
4	Limits (legal and ethical limits that may constrain the agent)	Code of conduct, legal, minor extra-legal, major extra-legal.
5	Resources	Individual, club, contest, team, organization, government.
6	Skill level	None, minimal, operational, adept.
7	Objectives	Copy, deny, destroy, damage, take, all/don't care.
8	Visibility	Overt, covert, clandestine, multiple/don't care.

Every threat agent is characterized by a specific category in each of these defining attributes, as is shown in the Threat Agent Library (TAL) matrix, Table 3.10. The Intel TAL considers twenty-one categories of threat agent. The relevance of each to the asset under scrutiny is a matter of analysis and is determined, in part, by the nature of the threats and the attacks they might manifest. If the target solution is a small, unknown, and unimportant application, attacks might be more likely to originate with less lethal threat agents. If, however, the target solution's status was that of a national infrastructure, it might find itself a target of more serious contenders such as organized crime and nation-states.

The TAL matrix places each threat agent at the head of a column, and each threat agent attribute at the start of a row. (The exception to this convention being the intent attribute, which has been placed above the column definitions to more conveniently show which threat agents have hostile or non-hostile intent.) Each cell in the table represents an intersection of a threat agent with an attribute. Cells colored blue indicate that the threat agent is associated with the threat attribute, while cells colored white indicate no association. Because the TAL is defined and constructed so that each threat agent possesses a unique set of attribute categories, careful scrutiny of attribute categories will help establish the focus of the threat model by mapping the set of asset-centric threats with the threat agents likely to cause them, and will at the same time provide a first iteration of narrowing down the threat agent list (i.e., eliminating those threat agents that are unlikely to attack the asset).

Given a threat agent library, and a prioritized list of critical assets, it is possible to develop specific threat models for given threat agents by paring down the target trees to only those branches relevant to the threat agent. This process is illustrated with an assessment of the insider threat.

Table 3.10 – Threat Agent Library

Intent		NON-HOSTILE			HOSTILE																	
		Employee, Reckless	Employee Untrained	Info Partner	Anarchist	Civil Activist	Competitor	Corrupt Government Official	Data Miner	Employee Disgruntled	Government Cyber-warrior	Government Spy	Internal Spy	Irrational Individual	Legal Adversary	Mobster	Radical Activist	Sensationalist	Terrorist	Thief	Vandal	Vendor
Access (1)	Internal	■	■	■						■										■		■
	External				■	■	■	■	■		■			■	■	■	■	■	■		■	
Outcome (1-2)	Acquisition/Theft												■			■				■		
	Business Advantage						■	■	■			■			■							■
	Damage	■	■	■	■					■	■			■			■	■	■		■	
	Embarrassment	■	■	■		■				■	■			■	■		■	■				
	Tech Advantage						■	■	■			■	■									■
Limits (max)	Code of Conduct		■	■																		
	Legal	■													■							■
	Extra-legal, Minor					■	■	■	■				■				■	■		■	■	
	Extra-legal, Major				■					■	■	■		■		■			■			
Resources (max)	Individual	■	■	■						■				■						■		
	Club				■													■				
	Contest																				■	
	Team								■													■
	Organization					■	■						■		■	■	■		■			
	Government							■			■	■										
Skills (max)	None				■									■						■		
	Minimal		■															■				
	Operational			■						■											■	■
	Adept	■				■	■	■	■		■	■	■		■	■	■		■			
Objectives (one or more)	Copy					■	■		■			■	■		■							■
	Deny							■			■											
	Destroy				■					■	■								■			
	Damage									■	■								■			
	Take														■	■			■	■		
	All of the above/Don't care	■	■	■										■			■	■			■	

Threat Agent Library (cont.)

		Employee, Reckless	Employee Untrained	Info Partner	Anarchist	Civil Activist	Competitor	Corrupt Government Official	Data Miner	Employee Disgruntled	Government Cyberwarrior	Government Spy	Internal Spy	Irrational Individual	Legal Adversary	Mobster	Radical Activist	Sensationalist	Terrorist	Thief	Vandal	Vendor
Visibility (min)	Overt		■		■			■							■		■	■				
	Covert	■				■										■			■		■	
	Clandestine			■			■		■			■	■							■		■
	Multiple/Don't care									■	■			■								

3.2.1.3.2 Example: The Insider Threat

The Department of Homeland Security defines the insider threat as “the potential violation of system security policy by an authorized user.” (Goldman, 2014) The insider threat is a significant problem, generally, which is why it was selected as an example.

The three most common insider activities are:

1. Theft or modification of confidential or sensitive information for personal gain.
2. Theft of trade secrets or customer information to be used for business advantage or to give to a foreign government or organization.
3. Sabotage of an organization’s data, systems, or network.

The above represent the primary working set of offenses against the asset selected, using *data* as an example.

In evaluating the insider threat, turn to the TAL shown in Table 3.10. In doing so, it is recognized that some modern definitions of “insider” can include cloud provider personnel, the makers of 3rd-party software that might be used to process the data, attackers who commandeer the system that runs the database management system, an attacker who comes in to the solution over an authorized communications link, and many other possible offenders. These parties are not explicitly called out by the TAL, though some might be tacitly included in its general insider threat agent categories. For simplicity this example follows the TAL’s threat agent list, with the understanding that anybody it leaves out of the insider category will be treated as an extremely talented outsider who has somehow managed to gain partial or complete authorized access to the target solution.

Extracting the entries related to insiders gives the full set of inside threat agents, shown in Table 3.11. This table aggregates all threat agents having internal access (which accounts for the entire row labeled Access: Internal being solid). The table clearly delineates the objectives of each insider threat agent, as well as their skill levels, affiliations (resources), objectives, and the limits to which they will go to achieve their goals. (Note that for completeness, all rows have been retained, including the “unmarked” rows.)

Table 3.11 – Insider Threat Agents and Attributes

Intent		NON-HOSTILE			HOSTILE		
		Employee, Reckless	Employee Untrained	Info Partner	Employee Disgruntled	Thief	Vendor
Access (1)	Internal	■	■	■	■	■	■
	External						
Outcome (1-2)	Acquisition/Theft					■	
	Business Advantage						■
	Damage	■	■	■	■		
	Embarrassment	■	■	■	■		
	Tech Advantage						■
Limits (max)	Code of Conduct		■	■			
	Legal	■					■
	Extra-legal, Minor					■	
	Extra-legal, Major				■		
Resources (max)	Individual	■	■	■	■	■	
	Club						
	Contest						
	Team						■
	Organization						
	Government						
Skills (max)	None					■	
	Minimal		■				

Intent		NON-HOSTILE			HOSTILE		
		Employee, Reckless	Employee Untrained	Info Partner	Employee Disgruntled	Thief	Vendor
	Operational						
	Adept						
Objectives (one or more)	Copy						
	Deny						
	Destroy						
	Damage						
	Take						
	All of the above/Don't care						

Significant analysis of the insider threat could be undertaken to determine the likelihood of an insider attack producing one or more of the three outcomes listed above; however, to simplify this example, assume that an insider attack *will* occur in one form or another and that its goal will be to produce a high level of harm to the data.

Of the threat agents listed in Table 3.11, disgruntled employees must be considered to be the most dangerous because their objectives are to damage or destroy; they are willing to violate the law in the extreme; their skills may be significant; and the outcomes of their attacks are harm to the data. Trailing closely behind are the reckless employee and the information partner, because of their harmful objectives and relatively high skill levels. The remaining threat agents, though possibly less dangerous, are nonetheless considered significant risks. Assume that the threats posed by the presence of such threat agents in the personnel pool, the inevitability of an attack from their ranks, and the repercussions of an attack, demand that the data owners take all actions possible to detect and react to the signs and signals that might portend an insider attack, or that might suggest that an employee could be starting down a path that may lead to becoming an attacker, and protect the data and systems against insider attacks.

To show the adversary-centric component of the threat model at work, this study will now examine the first of the three insider activities listed at the start of this clause. In this case, the asset is data, so refer back to Clause 3.1.2.2 for the attack tree.

3.2.1.3.3 Theft or Modification of Confidential/Sensitive Information

An insider attack having the goal of data theft or modification can involve one or more of data's attributes shown in Table 3.4. Of the five attributes, *Non-repudiation* can be ignored because the insider is not an actor who would enter data only to later deny having done so. *Authenticity* can also be ignored because the insider is unlikely to build a false "feeder" system to load spurious data into the database. Figure 3.19 below illustrates the attributes, risks, and attack classes directly associated with data theft or modification. This target tree has been "pruned" from the target tree in Figure 3.15. To illustrate the result of pruning, the figure shows the original target tree, but grays out those nodes not relevant to an insider attack. The resulting pruned tree shows only those attributes, risk, and attack classes pertinent to an insider attack.

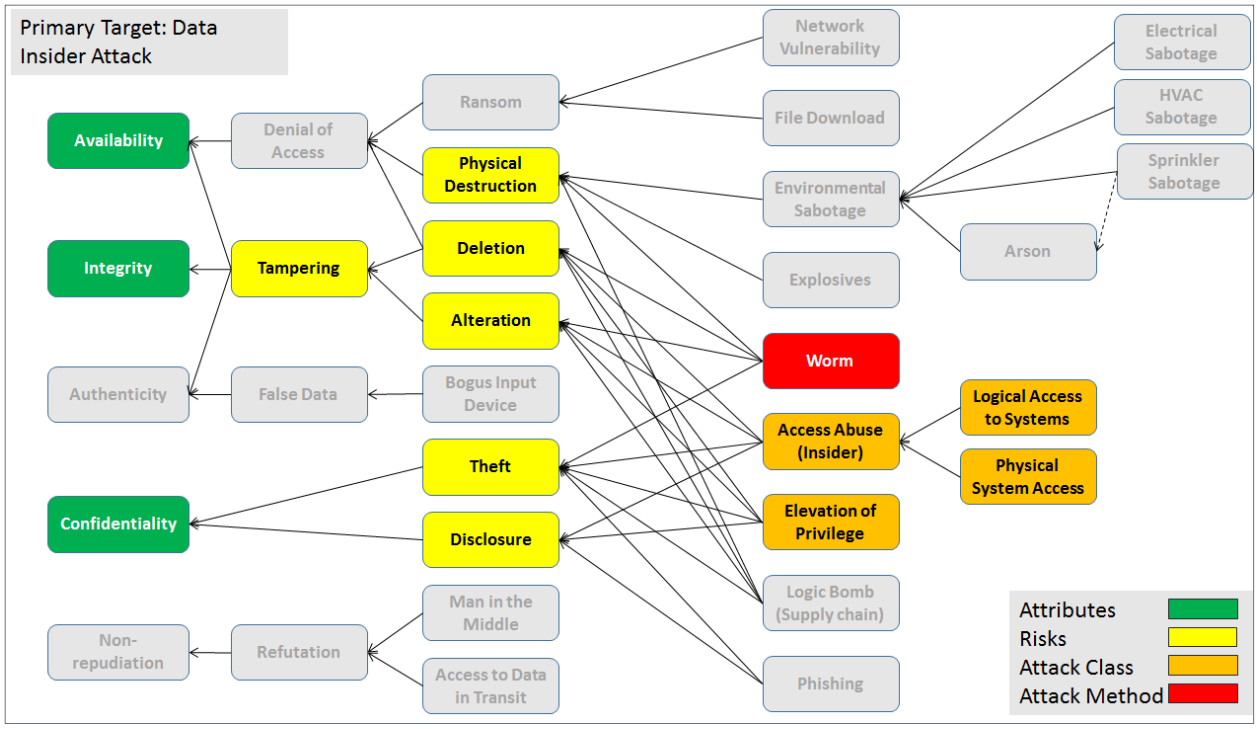


Figure 3.19 – Target Tree for Insider Attack: Data Theft or Modification

The target tree shows that the attack classes that lead to data theft or modification are characterized primarily by abuse of privilege in the form of outright misuse or by elevation of access privileges. In the figure, the attack vector labeled “Worm” represents surreptitious and malicious code, or exploitable vulnerability, produced/inserted by an employed developer, or other employee having access to the production code in the software development life-cycle.

The path metric is shown in Table 3.12. *Authenticity* and *Non-repudiation* have been omitted because the attack tree shows no effect on them.

Table 3.12 – Path Metric for Data (Insider Attack)

Attribute	Rank	Path
Availability	3	3
Confidentiality	5	7
Integrity	4	3

The weight metrics are shown in Table 3.13. For convenience and simplification, it displays only the relevant leaf nodes. It should be mentioned that “Logical Access to Systems” refers to access by an authorized administrator, while “Elevation of Privilege” refers to manipulations that give an ordinary user (or intruder) privileged access (e.g., root access).

Table 3.13 – Weight Metric for Data (Insider Attack)

Attack (Class)	Weight
Worm	12
Logical Access to Systems	12
Physical System Access	12
Elevation of Privilege	6

3.2.1.3.4 Assessment & Mitigation

The target tree shows that the attack vectors that lead to data theft or modification are characterized primarily by abuse of privilege in the form of outright misuse or by elevation of access privileges. A solution might center on strict control of access by privileged insiders, which might include stronger authentication, more detailed logging of privileged actions, a broader view of what actually constitutes privileged access, and possibly more reliance on processes that require more than one person to perform sensitive operations (separation of duties), such as duplicating databases, backing up data, and administering database functions that can provide access to the stored data itself. The high weight of "Physical System Access" suggests a need for: secure transmission of sensitive data between internal systems; intrusion-resistant cabling; a policy that forbids long-term and unattended use of protocol analyzers; other test equipment that can display or record network traffic; and practices that wipe the memories of such devices every time they are removed from the network. The high weight of the "Worm" attack suggests that the security architecture's safeguards include detection of insertions of new programs or vulnerabilities into the execution environment.

Changes to data can be undone if the modifications can be pinpointed and copies of the real data are used to reconstitute the affected portions of the original database. This suggests a need for robust, secure data change control and recordkeeping, a reliable backup system that maintains a sequence of progressively older copies, and an active checkpoint-restart capability.

This example shows how pruning the target tree to concentrate on a given class of adversary can both simplify the threat assessment process and help eliminate those threat classes that are not relevant. It also shows how to derive a high-level mitigation plan to address attacks from the adversary class.

3.2.1.4 Additional 3rd Party Considerations

Applying the threat model to 3rd Party components and platforms in Service Provider solutions requires no additions to the model itself, but does require that the modeling effort take into account the following:

1. Components provided by 3rd Parties can extend the attack surface of the solution. Any trusted and untrusted platforms, interfaces, data flows, and functions for such components might need to be assessed as primary or priority secondary assets.
2. The threat agent set might need to be enlarged to include threat agents who have physical or logical (e.g., remote) access to the 3rd Party components.
3. The possibility of the Service Provider solution becoming collateral damage in an attack against another customer collocated on 3rd Party components or premises might need to be assessed.

The threat assessment model described here represents a preliminary effort to develop a threat modeling methodology that can be applied to most IT security threat scenarios, and perhaps to other disciplines outside the IT world. The objective in developing this methodology is to remove some of the ambiguity that can accompany the threat modeling process, by focusing on security-related attributes of an asset instead

of on the asset itself. By compartmenting the asset in this fashion, a clearer indication can be obtained of the risks to the asset by discovering how each attribute is at risk, which leads to the attack classes that transmogrify those risks[6] into threats to those attributes, and to the asset itself. By further pruning the attack tree on the basis of an adversary-centric component, determination can be made of how much attention to give different classes of adversaries, and thus how much protection to provide to ward off given attacks or attack classes.

3.2.2 Abuse Cases

The development of abuse cases is an important step in the ARA Process as they provide real-world instantiations of the attacks that are identified as part of the threat analysis step. Abuse cases are also used to define the points in the solution architecture where the attacks can be manifested. The development of abuse cases requires a clear understanding of the possible threats/attacks, the solution's architecture and creativity in creating a complete, and sometimes non-intuitive, application of attacks that realize the identified threats.

The following considerations should be applied in the development of a complete set of abuse cases for a solution:

- *Abuse Cases* are created to define how the attacks that are identified in the solution's *Threat Model* may be manifested at the *Attack Points* identified on the *Solution Architectural Diagrams*. Abuse cases and attach points that fully represent and cover attacks via 3rd Party components must also be identified.
- *Abuse Cases* should be based upon real-world, likely attack scenarios that the solution will face in production use.
- A single *Abuse Case* may potentially cover one or more manifestations of a threat at one or more *Attack Points* in the *Solution Architecture*.
- It is important that the final set of *Abuse Cases* that are created as part of an ARA fully cover the *Threats, Attack Points,* and *Use Case* functionality of the solution.
- There should be specific *3rd Party Abuse Cases* identified to verify that the *Trust Model* specified for each 3rd party component is fully realized as specified. Abuse cases that confirm that the trusted capabilities of each 3rd party component do not contain unintended vulnerabilities (i.e., a *Trust but Check approach*) are also required.
- *Abuse Case* development should be performed independently from the design of the solution and associated security controls.
    - It may be effective to use a "Red Team" approach to realize this step.

[6] In the sense of risk discussed in Clause 3.2.2.

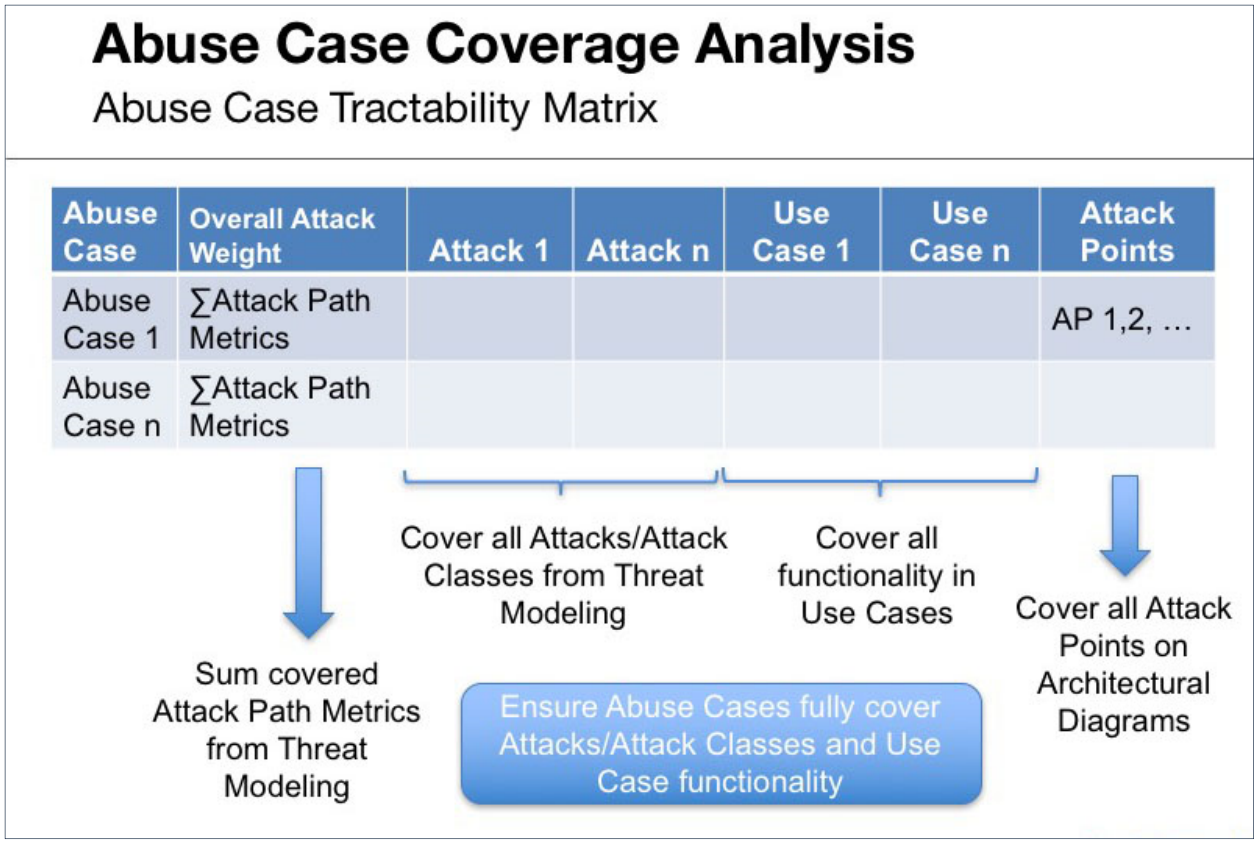


Figure 3.20 – Abuse Case Traceability Matrix

The traceability matrix (Figure 3.20) was developed as a tool to help ensure that the baseline set of abuse cases is complete and fully covers the threats/attacks, use case functionality, and possible attack points in the solution's architecture. The following guidelines are meant to aid in the completion of the traceability matrix:

- Completion of the traceability matrix as part of the abuse case step in the ARA Process requires that the abuse cases fully cover the threat/attack classes from the threat analysis and functionality in the solutions' use cases.
- All possible relevant attack point(s) (i.e., those that result from the threat analysis) in the solution's network diagram(s) must be covered by the final set of abuse cases.
- The ***Attack***, ***Use Case***, and ***Attack Point*** columns must also cover all potential vulnerabilities and risks in 3rd party components as identified in the preceding ARA Process steps.
- Path weight metrics for the covered attacks identified as part of the threat analysis step indicate the importance/risk level associated with each abuse case. This information may also be useful:
    - As a tool for test activity and associated DevOps priority setting.
    - To assess impact of any unmitigated attacks associated with the abuse case.
- Subsequent steps in the ARA process need to identify security controls to mitigate attacks associated with each abuse case.
- Some security controls may be prioritized or deferred.

The abuse case traceability matrix is a key deliverable that should be reviewed to determine the completeness of the abuse case step in the ARA process. If any of the attacks or use case columns in the matrix are not covered (i.e., nothing in the column is checked), there should be a careful review of the associated decisions to leave those areas determined to be unmitigated.

3.3 Risk Analysis & Threat Mitigation Plan

The objective of the risk analysis and mitigation activity is to assess the effectiveness of security controls and countermeasures. The intent is to identify areas of improvement to further strengthen security or to identify gaps that need additional security controls applied. Each threat identified in prior activities will be individually evaluated to ensure it is properly mitigated. Industry references, such as the Cloud Controls Matrix from the Cloud Security Alliance (see clause A.10 CSA Cloud Controls Matrix), should be used to determine that the proper controls are applied to each threat.

The risk mitigation steps and the associated analysis summary (Table 3.14) should fully address all attacks and associated attack points that can occur through or in connection with the 3rd party components identified as part of the solution ARA process. If the mitigation of a given attack is determined to be the responsibility of a 3rd party, these requirements must be carefully verified through security testing and other positive security review steps.

In rare cases, there may be no known control or mitigation to address certain threats. When this is the case, consideration should be given to reducing the impact of the threat if a successful exploit were to occur. Analysis may also lead to consideration of detection methods so that if a threat cannot be fully mitigated, detection of the attack may aid service providers in taking action to limit potential impact. Detection methods are outside the scope of the ARA activity, but this process may help teams identify where detection methods are critical to the overall security strategy.

Threats and the associated risks that are not fully mitigated should be prioritized to enable teams to take action in accordance with the severity of the threat. Several factors should be considered in determining priority rank. Among them are ease of exploit, severity of potential damage done, and cost of recovery. Identifying each of these categories as high, medium, or low is sufficient to provide initial priority. Depending on an organization's overall risk management policies and processes, the information from the ARA activity may feed into other risk activities.

The table below may be used to summarize the findings.

Table 3.14 – Risk Mitigation Coverage and Priority Analysis

Risk Mitigation Step	Overall Priority	Attacks Mitigated (from Threat Analysis)	Overall Attack Weight	Ease of Exploit	Severity of Damage Done	Cost of Recovery	Attack Points
Risk Mitgn. Step 1	H, M, L	Attack 1	∑Attack Path Metrics	H, M, L	H, M, L	H, M, L	AP 1, 2, …
Risk Mitgn. Step n							
		Attack n					

The ARA Process Example Annex (which is available as a separate document) contains an example application of the ARA process to a hypothetical service example.

4 Cybersecurity Testing Strategy

The development of a comprehensive cybersecurity testing strategy for a solution is beyond the scope of the ARA Process. That said, the ARA Process can and should provide some key inputs to the final solution for a cybersecurity test strategy and associated plans. The most useful inputs will come from the *Abuse Case*, *Risk Mitigation/Security Controls*, and *Threat Analysis* steps in the process. The risk mitigation/security controls elements should be tested to ensure that the resulting security controls' functionality performs as expected. Such testing should include steps to ensure that the trust models defined for 3rd party components are properly realized. The abuse cases provide a set of direct attack cases that can be used to test the solution from an attacker's perspective. Finally, the threats/attacks identified as part of the threat analysis step can be used to develop additional cybersecurity test cases.

The following additional considerations should be taken into account when developing the final cybersecurity test strategy and plans:

- Effective cybersecurity test strategies for a solution should include test suite elements that realize (i.e., provide a means to execute) the *Abuse Cases* and ensure that the test suite verifies the effectiveness of the *Security Controls.*
    - Leverage *Architectural Diagram* analysis and *Threat Analysis Tree Pruning* information to efficiently limit scope.
- In addition, heuristic-based or fuzzing test cases should include a focus on the specific *Attacks* and *Attack Classes* identified in the *Threat Modeling* step. These should be applied at the identified *Attack Points*.
- The associated *Threat Weighting* and *Abuse Case* priorities derived from the ARA Process should be used as an input to prioritize test suite elements and any related defect prioritization.
- The trust models for the 3rd party components in a solution should be fully verified and additional tests to verify that the trusted elements of these solutions perform as intended and do not contain unintended vulnerabilities (i.e., *Trust but Check*).
- These test plans are probably best developed using a "Red Team" approach that is independent of the developers of the *Risk Mitigation Plans*.

5 Practical Guidance

The ARA Process may be difficult to apply for first-time users. This clause provides some practical guidance to help teams cope with some common challenges.

5.1 Applying ARA to Existing Design

The ARA Process is ideally applied during the design and development of new networks or services ensuring that security is properly considered in the architecture from the beginning. However, the ARA Process can be used with existing networks or services to evaluate the security posture. When applying to existing or legacy designs, more flexibility can be exercised regarding the workflow and where to begin. A practical first step is to collect existing documentation focusing on two categories of information: diagrams that depict architectural structure or data flow and lists and descriptions of supported use cases. When diagrams are difficult to find, interviewing senior members of the team can be helpful to begin identifying the architecture that can then be documented. Legacy systems may have hundreds of functional use cases and it may seem overwhelming to teams to evaluate each one. Starting with the top 10 most common or crucial use cases is a good way to start. As described in a subsequent clause, teams can take an iterative approach to ARA by considering additional sets of use cases in future work.

5.2 Defining Security Goals

Developing security goals is a key part of the process that helps those involved define a context that makes security more understandable and the process more practical; Table 5.1 provides examples that may be helpful in understanding and crafting security goals and objectives.

Table 5.1 – Sample Security Goals

Solution Description	Sample Security Goal
A solution's primary function is to support web-based purchases for consumers accessing an Internet-based business.	Protect privacy of customer data during online purchase transaction.
A small business provides digital printing services through an online portal that allows customer to upload digital images.	Confirm integrity of digital picture file uploaded from a user portal at kiosk.
A network service provides wireless communications to local utilities for transceiving telemetry data between utility meters and readers.	Ensure all machine-to-machine service requests are from authenticated devices registered with public utility.
A service provider's back office system ensures services are provided in accordance with the customer's specific data plan.	Prohibit unauthorized account credits.
A network service provides access to online event ticket sales portals. High traffic is expected when purchase windows for popular events initially open.	Concert tickets must be available for purchase to legitimate buyers during the public offer window.

5.3 Getting Started with Partial Data

Some teams may struggle to get started with the ARA Process because complete information about the solution or network and its use cases may not be readily available. Others may struggle because no single person has a comprehensive understanding of the architecture. It is important to get started and a minimal amount of data can be sufficient. Start with the primary external interfaces that are well understood, and with a small set of primary use cases or functional operational descriptions of the primary services provided. Another strategy is to begin with a subset or sub-system of the overall solution or network. Start with what is understood best and work to address other areas over time.

5.4 Iterative Approaches

Because a comprehensive analysis may not be practical in a single activity, an incremental iterative approach may be required. Each iteration may yield actionable results and will definitely set the stage for future progress. As described in the previous clause, starting with a very simple, high-level architectural view using the primary functional use cases is appropriate. Future iterations can drill deeper into the architecture, incrementally exposing various sub-components and interfaces. Once the primary use cases have been addressed, additional use cases may be considered in future iterations. If a data-flow approach is used, focus on a single type or classification of data in each iteration. Another method of dividing the work into incremental activities may be to focus on types of users or different services provided.

5.5 3rd Party Components & Services

The use of 3rd party components and services is increasingly common in complex solutions as many companies strive for a best-of-breed approach coupled with trends in outsourcing to reduce development costs, decrease time-to-market, and stabilize ongoing operational costs. Applying ARA to a system that includes hosted services or cloud services can be difficult since the details of those 3rd party components are not known. Ideally, a joint ARA activity in collaboration with the 3rd party provider would yield the most comprehensive results. However, such an activity is not usually practical. The ARA can proceed, however, by treating the 3rd party service as a major component positioned across a trust boundary. All interfaces should be carefully documented and traffic content evaluated for security requirements. Proper controls must be applied to ensure data is protected at rest and during transit, and that all transactions are properly authenticated and authorized. The exposure of the interface should be treated as an untrusted, fully-exposed interface to the Internet.

5.6 Technology-based Threat Assessment

Clause 3.2 on Threat Identification introduces two strategies for consideration: adversary- and asset-based threat modeling. Another approach that can supplement those two methods is based on attack mechanisms or domains. Considering the technologies in use, or the domains of attack, can reveal additional insight into potential threat vectors and the security controls required to mitigate them. For example, if a solution includes a database that is accessible through a web-based GUI, care must be taken to ensure it is not vulnerable to SQL Injection attacks that could corrupt the database or inadvertently reveal its contents to an unauthorized user. Careful evaluation of the underlying technologies and domains of the solution helps in the threat assessment steps of ARA. Detailed protocol analysis can also reveal potential attack vectors. It may be necessary to seek additional input from protocol experts to help identify how obscure protocols could be abused and thereby present additional mechanisms of attack.

The Common Attack Pattern Enumeration and Classification (CAPEC) website hosted by Mitre Corporation at https://capec.mitre.org/index.html provides information on hundreds of attack patterns. They can be sorted by mechanism or domain and can help teams develop a customized threat library.

6 Conclusions & Additional Work

The ARA Process, when used with the best practices outlined in Appendix 1, provides a complete and sound approach for performing a cybersecurity architectural risk analysis of both complete, carrier-grade solutions, and the components which support them. The resulting risks are prioritized and can be mitigated using industry best practices (see Appendix 1).

There are some additional areas for possible future work which would expand the solution scope of the ARA Process and further facilitate its application. These areas include:

- Develop application notes (ANs) for using the ARA Process for certain solutions. Possible ANs include:
  - Solutions involving a service provider and a 3rd party (e.g., jointly delivered IoT solutions)
  - Solutions or components where network level security (e.g., signal protocol-based attacks) are dominant
- Analyze existing best practices in terms of their support for the ARA Process and identify any areas where new or enhanced best practices are needed.
- It would be valuable to expand upon the idea of “Technology-centric” or perhaps “Mechanism-centric” threats. This would potentially help teams identify threats associated with weaknesses in protocols, outdated cryptography (i.e., weak ciphers), poor input validation mechanisms, etc.

7 Bibliography & References

1. Casey, Timothy.*Threat Agent Library Helps Identify Information Security Risks.* s.l. : Intel Corporation White Paper, September 2007.

2. Goldman, Jeff. FBI, *DHS Warn of Surge in Insider Threats from Disgruntled Employees*. [Online] eSecurity Planet, September 25, 2014. < http://www.esecurityplanet.com/network-security/fbi-dhs-warn-of-surge-in-insider-threats-from-disgruntled-employees.html >.

3. U.S. Department of Homeland Security: Science and Technology Directorate, Cyber Security Division. *Insider Threat.* March 2016.

4. *Intelligence Arts: Advancing the Art of Human Intelligence*. [Online] June 6, 2016. < http://intelligencearts.com/?page_id=195 >.

5. The Open Group. *Technical Standard Risk Taxonomy.* January 2009. ISBN 1-931624-77-1, Document Number: C081.

6**.** Technology, U.S. Department of Commerce: National Institute of Standards and Technology. *Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach.* Washington D.C. Government Printing Office, February 2010. Special Publication 800-37.

7**.** Cigital. *The 6 Most Common Threat Modeling Misconceptions.* [Online] 2016. < https://www.cigital.com/resources/ebooks-and-whitepapers/security-threat-modeling-misconceptions/ >.

8 Acronyms

4G	4th Generation Wireless Mobile Technology
5G	5th Generation Wireless Mobile Technology
ARA	Architectural Risk Analysis
API	Applications Programming Interface
AP	Attack Point
APT	Advanced Persistent Threats
ATIS	Alliance for Telecommunications Industry Solutions
AWS	ATIS Workspaces
BP	Best Practice
C-S	Cybersecurity
CAPEC	Common Attack Pattern Enumeration and Classification
CCS	Council on Cybersecurity
CERT	Computer Emergency Response Team
CIA	Confidentiality, Integrity and Availability
CISO	Chief Information Security Officer
COBIT	Control Objectives for Information and Related Technology
CPU	Central Processing Unit
CSA	Cloud Security Alliance
CSC	Critical Security Controls
CSRF	Cross-Site Request Forgery
CSRIC	Communications Security, Reliability and Interoperability Council
DoS	Denial of Service
DDoS	Distributed Denial of Service
DHS	Department of Homeland Security
DNS	Domain Name Service
ENISA	European Networking and Information Security Agency
EPC	Evolved Packet Core
EPD	End Point Device
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
HVAC	Heating, Ventilation, and Air Conditioning
IaaS	Infrastructure as a Service
IAM	Identify and Access Management
ICT	Information and Communications Technology
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
ISAC	Information Sharing and Analysis Center

ISAO	Information Sharing and Analysis Organizations
ISO	International Organization for Standardization
IT	Information Technology
ITU	International Telecommunication Union
HTTP	HyperText Transfer Protocol
HTTPS	HTTP Secure
LE	Low Energy
LTE	Long Term Evolution
M2M	Machine-to-Machine
MVNO	Mobile Virtual Network Operator
NFV	Network Function Virtualization
OS	Operating System
PaaS	Platform as a Service
PII	Personally Identifiable Information
QoS	Quality of Service
RAN	Radio Access Network
REST	Representational State Transfer
RSA	Regional Studies Association, a UK-based learned society
SaaS	Software as a Service
SECaaS	Security as a Service
SDN	Software Defined Networking
SQL	Structure Query Language
SW	Software
TAL	Threat Agent Library
TARA	Threat Agent Risk Assessment
UC	Use Case
USB	Universal Serial Bus
USD	End-User Data
VM	Virtual Machine
VNF	Virtual Network Function

Annex A: Industry Reference Material

Throughout the development of this process numerous documents were reviewed that contributed to the collective understanding of the topics. Those identified in the following table were especially valuable and may be helpful to those executing the process as reference material. The table includes a title, brief synopsis, and an explanation of the document's potential use as part of the ARA process.

Table A.1 – Industry Practices That Support the ARA Process

Document	Synopsis	Useful For
ETSI NFV Security Documents (2)	These describe security issues and challenges as well as providing guidelines for implementing Network Functions Virtualization (NFV) to adequately identify exposure to security and trust issues.	Reference network diagrams, architecture, and security strategy.
CSA – Treacherous 12	Common pitfalls are highlighted for cloud computing deployments; also, serves as an excellent starting list for those wanting to establish and deploy a security framework for cloud computing.	Common cloud security pitfalls.
CSA – Security Guidance for Critical Areas in Cloud Computing	Provides security guidance for cloud architectures.	Security guidance for cloud architectures.
CSA – Security as a Service Implementation Guidelines	Provides details on security architectures and controls pertaining to SaaS and to cloud architectures.	Security architecture and controls.
Intel Threat Library	Provides a unique, standardized resource that offers a consistent and up-to-date reference describing the human agents that pose threats to IT systems and other information assets.	Definition of attack agents and threats.
European Networking and Information Security Agency (ENISA) Threat Landscape and Good Practice Guidance for SDNs	Overview of threats related to SDN environments with practical best practices for improving security.	Threat reference for SDN and related network architectures.
IETF Towards Secure and Dependable Software-Defined Networks	Academic paper addressing security vulnerabilities unique to SDN.	Security strategies for SDN; applicable to understanding controls.
IEEE, Center for Secure Design, Avoiding The Top 10 Software Security Design Flaws	Overview of 10 common software security design flaws and methods to avoid them.	Applicable to understanding threats.
IEEE, Center for Secure Design, WearFit: Security Design	Detailed analysis of a solution design that addresses threats and security controls.	Example of ARA-like analysis on a realistic solution that is well described.

Document	Synopsis	Useful For
Analysis of a Wearable Fitness Tracker		
CSA Cloud Controls Matrix	Comprehensive spreadsheet of security controls classified in domains. Excellent cross reference to common industry security frameworks and standards.	Understanding controls applicable to cloud architecture.

The following clauses provide a brief overview and index to aid in using the practices in Table A.1 – Industry Practices That Support the ARA Process to complete the steps in the ARA Process. Each document profile will include:

- A brief overview of the document.
- What specific clauses of the document are useful for which elements in the process.
- Information on where to locate the latest version available at the time of publication of this document.

A.1 ETSI NFV Security Documents

A.1.1 ETSI GS NFV-SEC 001 Network Functions Virtualisation (NFV); NFV Security: Problem Statement and ETSI NFV Security Documents: ETSI GS NFV-SEC 003 Network Functions Virtualisation (NFV); NFV Security; Security and Trust Guidance

These two documents are generated by ETSI for describing the security issues and challenges as well as providing guidelines for implementing NFV to adequately identify exposure to security and trust issues.

A.1.1.1 GS NFV-SEC 001 V1.1.1 Published 2014-10

This document provides a reference framework for NFV and how vulnerabilities are defined using this framework. In addition, once exposing the potential security vulnerabilities, the document tries to show if these vulnerabilities are a result of NFV deployment or existing vulnerabilities that are present even in a non-NFV deployment environment.

GS NFV-SEC 001 extends beyond the Monolithic Service Provider configuration of telecom networks and into detail about potential deployment scenarios. NFV can now be implemented in distributed fashion since functions are virtualized and can be separated by geographic and operational boundaries that expose new levels of threats. Network topology models are presented to demonstrate the levels of complexity that can be introduced without guidelines, policies, and procedures that are critical for secure deployment of NFV.

A.1.1.2 GS NFV-SEC 003 V1.1.1 Published 2014-12

This document had the goals of: (1) establishing a secured baseline of guidance for NFV operation, while describing possible measures that improve security to be appropriate with risks to confidentiality, integrity, and availability; (2) defining areas of consideration where security technologies, practices, and processes have different requirements than non-NFV solutions and operations; and (3) provide direction for the operational environment that supports and interfaces with NFV solutions and operations.

The GS NFV-SEC 003 document does an exemplary job of describing the functional components of NFV and indicating the security and trust exposure of the NFV components in operational use case scenarios. The

deployment flexibility provided by implementing an NFV architecture also can expose security and trust challenges not anticipated in traditional telecom network deployments.

A.2 CSA – Treacherous Twelve

Each year, the CSA creates the "Treacherous Twelve". Common pitfalls are highlighted for cloud computing deployments. The "Treacherous Twelve" list also serves as an excellent start for those wanting to establish and deploy a security framework for cloud computing.

Some links to important documents:

Cloud Security Alliance Releases "The Treacherous Twelve" Cloud Computing Top Threats in 2016:

Blog:

https://cloudsecurityalliance.org/media/news/cloud-security-alliance-releases-the-treacherous-twelve-cloud-computing-top-threats-in-2016/

Report:

https://downloads.cloudsecurityalliance.org/assets/research/top-threats/Treacherous-12_Cloud-Computing_Top-Threats.pdf

The Treacherous Twelve: Cloud Computing Top Threats in 2016, developed by the CSA Top Threats Working Group and sponsored by Hewlett Packard Enterprise, serves as an up-to-date guide that will help cloud users and providers make informed decisions about risk mitigation within a cloud strategy. While there are many security concerns in the cloud, this report focuses on 12 related to the shared, on-demand nature of cloud computing.

The "Treacherous Twelve" are listed below:

- Data Breaches
- Weak Identity, Credential, and Access Management
- Insecure APIs
- System and Application Vulnerabilities
- Account Hijacking
- Malicious Insiders
- Advanced Persistent Threats (APTs)
- Data Loss
- Insufficient Due Diligence
- Abuse and Nefarious Use of Cloud Services
- Denial of Service
- Shared Technology Issues

A.3 CSA – Security Guidance for Critical Areas in Cloud Computing

The CSA document on "Security Guidance for Critical Areas in Cloud Computing" provides security guidance for cloud architectures. Information on the report and a related blog can be found at:

Blog:

https://cloudsecurityalliance.org/group/security-guidance/

Report:

https://downloads.cloudsecurityalliance.org/initiatives/guidance/csaguide.v3.0.pdf

CSA Security Guidance for Critical Areas of Focus in Cloud Computing seeks to establish a stable, secure baseline for cloud operations. This effort provides a practical, actionable roadmap to managers wanting to adopt the cloud paradigm safely and securely. Domains are reviewed to emphasize security, stability, and privacy in a multi-tenant environment.

Security Guidance Version 4.0 incorporates the highly dynamic nature of IT and new developments within other CSA research projects, tying in various CSA activities into one comprehensive C-level best practice. Security Guidance v4.0 will serve as the gateway to emerging standards being developed in the world's standards organization and is designed to serve as an executive-level primer to any organization seeking a secure, stable transition to hosting its business operations in the cloud.

A.4 CSA – Security as a Service Implementation Guidelines

The "CSA-Security as a Service Implementation Guidelines" document provides details on security architectures and controls pertaining to SaaS and to cloud architectures. Information on the report and a related blog can be found at:

Blog:

https://cloudsecurityalliance.org/media/news/csa-releases-secaas-implementation-guidance/

Report:

https://cloudsecurityalliance.org/group/security-as-a-service//#_downloads

The Cloud Security Alliance (CSA) announced that its Security as a Service (SECaaS) Working Group has completed its peer review process and has published implementation guidance documents expanding upon their "Defined Categories of Service" document that was first made available in August 2011. The Working Group's Implementation Guidance now includes peer-reviewed documentation for each of the 10 service categories that were defined in the previous version. *The first category, Identity and Access Management (IAM), was released in September 2012. The following guidance documents: Data Loss Prevention, Web Security, Intrusion Management, E-mail Security, Encryption, Business Continuity & Disaster Recovery, Network Security, and Security Assessments* were presented at the CSA Summit at RSA Europe in London and can be downloaded for free at:

https://cloudsecurityalliance.org/research/secaas/#_downloads

A.5 Intel Threat Agent Library

Every threat to an information system has one or more accompanying threat agents – attackers whose motives, capabilities, and opportunities are responsible to carry out those threats. A problem with most categorizations of threat agents is the difficulty of "talking the same language" when discussing them. Terms such as "hacker" and "spy" are commonly used when discussing threats and the malefactors who carry them out, although many times the colloquial connotations attached to such terms have different meanings to different people, thus bringing more confusion than clarity to the discussion. This confusion is often compounded and prolonged because the terms in question are so familiar that the parties often fail to realize that they are using them differently and not all talking about the same thing.

The Intel Threat Agent Library (TAL) was produced by Intel's IT Threat Assessment Group as a unique, standardized resource that offers a consistent and up-to-date reference describing the human agents that pose threats to IT systems and other information assets. Threat information has historically been fragmented and sensationalized, with no standard agent definitions, making it difficult for risk managers to assess risks from specific

threat agents quickly and consistently. The TAL addresses these problems by providing a single standardized set of threat agent definitions ranging from government spies to untrained employees. To develop the TAL, Intel devised eight common threat agent attributes and defined 21 threat agents based on unique combinations of these attributes, and then rated and described the threats that each threat agent represents.

The Intel TAL defines 21 categories of threat agent, grouping them into hostile and non-hostile, and insider or not. A representative subset of the threat agents is shown in Table A.2 – Sample Threat Agents from TAL.

Table A.2 – Sample Threat Agents from TAL

Threat Agent	Common Tactics/Actions	Description
Anarchist	Violence, property destruction, physical business disruption.	Someone who rejects all forms of structure, private or public, and acts with few constraints.
Cyber Vandal	Network/computing disruption, web hijacking, malware.	Derives thrills from intrusion or destruction of property, without strong agenda.
Data Miner	Theft of Intellectual Property, PII, or business data.	Professional data gatherer external to the company (includes cyber methods).
Vendor	Theft of Intellectual Property or business data.	Business partner who seeks inside information for financial advantage over competitors.

The TAL defines eight attributes that it associates with each threat agent, to aid in the determination of the relevance and seriousness of each threat agent. Each attribute is linked to descriptive features referred to as *categories*. Table A.3 below shows each attribute and the categories associated with each.

Table A.3 – Threat Agent Attributes

	Defining Attribute	Category
9	Intent	Hostile or non-hostile.
10	Access	Internal or external.
11	Desired outcome of attack	Acquisition or theft, business advantage, damage, embarrassment (to attacked party), technical advantage.
12	Limits (legal and ethical limits that may constrain the agent)	Code of conduct, legal, minor extra-legal, major extra-legal.
13	Resources	Individual, club, contest, team, organization, government.
14	Skill level	None, minimal, operational, adept.
15	Objectives	Copy, deny, destroy, damage, take, all/don't care.
16	Visibility	Overt, covert, clandestine, multiple/don't care.

The TAL provides a compilation of the threat agents, their attributes, and their associated categories in a detailed table that, for each threat agent, indicates into which categories it falls. The table is shown below (Table A.4). The twenty-one threat agents are listed across the top, generating a set of labeled columns. Within each column, the categories into which the threat agent falls are highlighted in blue. It is important to note that no two columns are identically highlighted, indicating that each threat agent in the TAL is uniquely characterized.

When using the table, TAL users should start with the list of attributes, and determine their threat agents on the basis of the categories into which attackers might fall. Doing so eliminates any preconceptions the TAL users might have about the meanings of names such as "hacker" and "spy"; however, because the library consists of archetypes rather than precise descriptions of individuals, an exact match is not possible or perhaps even desirable. To help the TAL user to overcome the uncertainty that inexact matches might produce, the TAL methodology recommends that the TAL user firm up any issues with a focused set of relevant questions. The TAL methodology offers guidelines for generating such questions.

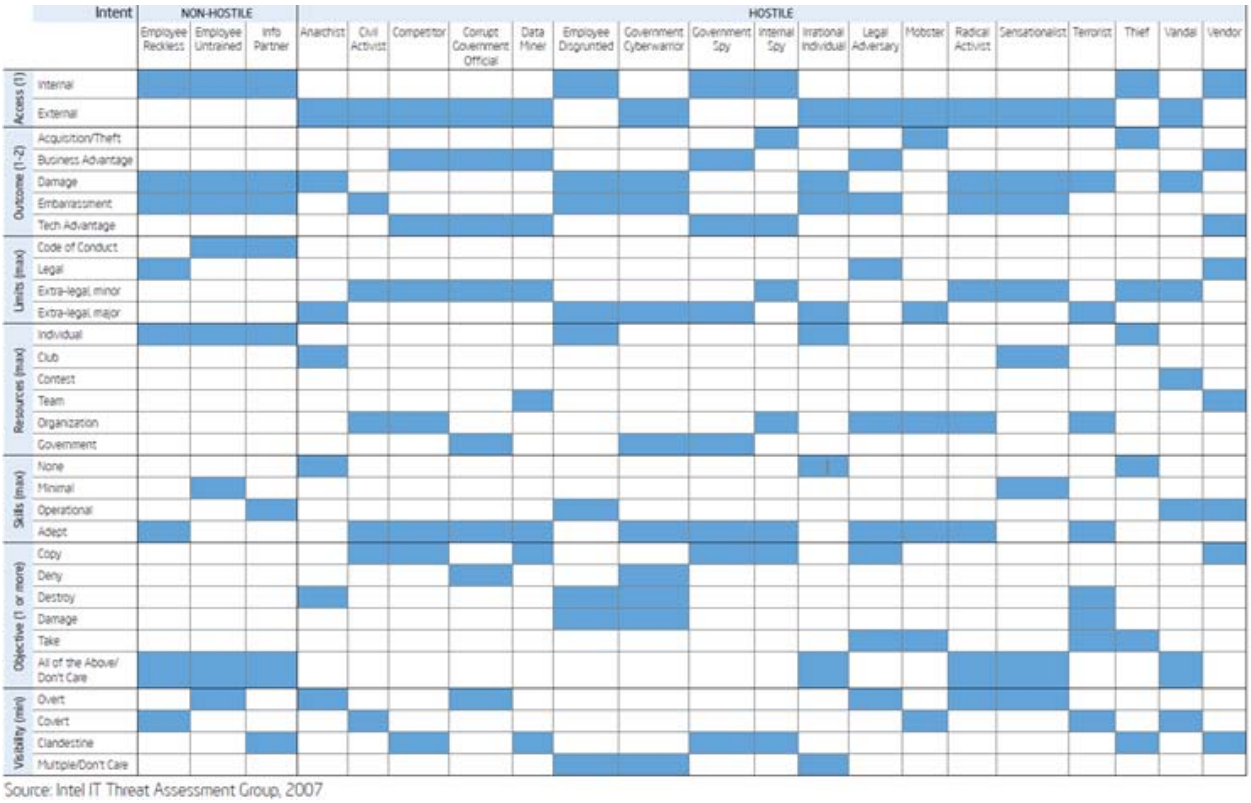
Intel uses its TAL within two risk assessment methodologies:

1. An Intel IT methodology that simultaneously addresses both general business risks and information-security specific risks. The TAL is available within a tool that supports this methodology.
2. A risk model-based methodology to perform regular security evaluations of Intel's manufacturing systems.

The TAL allows its users to evaluate the risk that these agents pose to specific assets. The agent ratings help with the risk assessment process by describing agents' activities, including which assets they target. A real value of the TAL is its relevance to the IT community at large. It represents one of the few – if not the only – comprehensive, well-developed, and currently maintained sources of threat agents, using a consistent terminology and defining each threat agent on the basis of attributes and categories. In doing so, it establishes a common vocabulary that facilitates the discussion of IT security threats.

Intel has subsequently developed and publicized a Threat Agent Risk Assessment (TARA) methodology that distills the large number of possible attacks into a digest of those exposures most likely to occur. The methodology identifies threat agents pursuing attainable objectives that could harm the target system or entity.

Table A.4 – Current Library of Threat Agents and their Defining Attributes



Source: Intel IT Threat Assessment Group, 2007

A.6 ENISA Threat Landscape & Good Practice Guidance for SDNs

A.6.1 Overview

The document addresses security threats specific to the Software Defined Network (SDN) infrastructure envisioned for 5G networks. As described in the document, it is part of a broader work that aims "to provide a contribution in the assessment of the exposure to cyber threats of the envisioned 5G networks with particular focus on their backbone networking technology represented by Software Defined Networking". This work is applicable in two ways: it addresses modern network architectures and it includes the application of virtualization technologies in networks in a security context. The paper defines categories of threats and discusses state-of-the-art methods of mitigation threats using commonly available tools.

The latest version of the document may be downloaded at the following site:

https://www.enisa.europa.eu/publications/sdn-threat-landscape

A.6.2 Clauses of the Document that are Useful in the ARA Process

- Clause 4.2 covers Categories of SDN Assets and presents them in a mind-map view. This correlates well with the asset driven threat model that is part of this process.
- Clause 5 covers threats in several key categories, then summarizes them in tables at the end of the clause. The tables identify threat types, threats, potential effect, and asset types.

- Threats are presented in a mind-map view reflecting a high-level threats taxonomy.
- The mitigation clauses are less useful as they are specifically targeted toward OpenFlow technologies.

A.7 IETF Towards Secure & Dependable Software-Defined Networks

A.7.1 Overview

This document is a brief academic paper that addresses SDN security by examining seven specific Threat Vectors. Of the seven vectors presented, only three are new or specific to SDN; but they have expanded impact or consequences in SDN. Strategies for security and dependability in SDN are also discussed.

The latest version of the document may be downloaded at the following site:

https://www.ietf.org/proceedings/87/slides/slides-87-sdnrg-2.pdf

A.7.2 Clauses of the Document that are Useful in the ARA Process

- Clause 2, which describes the seven threat vectors in detail, is very helpful in understanding threat vectors in general.
- Clause 3 on security design considerations and is applicable to any work.
- Many details are specific to SDN controllers and not directly applicable to other scenarios.

A.8 IEEE Center for Secure Design – Avoiding The Top 10 Software Security Design Flaws

A.8.1 Overview

The document is one of two initial publications from the IEEE Center for Secure Design. Unlike typical Top-10 lists, this paper presents 10 sound security practices that contribute to improved software. Each of the 10 topics are presented clearly and provide an understanding of associated threats.

The latest version of the document may be downloaded at the following site:

https://www.computer.org/cms/CYBSI/docs/Top-10-Flaws.pdf

A.8.2 What Specific clauses of the Document are Useful for which Elements in the Process?

Each of the 10 topics are required reading. Each clause includes thorough descriptions and practical advice.

A.9 IEEE, Center for Secure Design, WearFit: Security Design Analysis of a Wearable Fitness Tracker

A.9.1 Overview

The paper presents the design of a fictional wearable fitness tracker and assesses its design with respect to 10 security topics addressed in the companion paper, Avoiding the Top 10 Security Flaws. The analysis forms the basis of a threat modeling exercise and helped the working group verify the ARA process.

The latest version of the document may be downloaded at the following site:

http://cybersecurity.ieee.org/blog/2016/02/17/wearfit-security-design-analysis-of-a-wearable-fitness-tracker/

A.9.2 Clauses of the Document that are Useful in the ARA Process

- The Attack Categories clause provides clear and practical examples of threats the WearFit solution could encounter with specific sub-bullets on how each threat category could be executed along with the impact.
- The analysis clause evaluated each of the 10 security design flaws and assessed if the WearFit solution had been adequately designed to avoid those threats. The analysis also provides information about impact and attacker interest so that priorities can be derived for additional security controls.

A.10 CSA Cloud Controls Matrix

A.10.1 Overview

Published by the Cloud Security Alliance (CSA), the Cloud Controls Matrix is the first published framework that specifically addresses the cloud supply chain. The matrix covers 133 individual controls that span over 16 specific domains. The domains are listed below:

1. Application & Interface Security
2. Audit Assurance & Compliance
3. Business Continuity & Op Resilience
4. Change Control & Configuration Management
5. Data Security and Info Lifecycle Management
6. Datacenter Security
7. Encryption and Key Management
8. Governance & Risk Management
9. Human Resources
10. Identity and Access Management
11. Infrastructure & Virtualization Security
12. Interoperability & Portability
13. Mobile Security
14. Security Incident Management, E-Discovery, & Cloud Forensics
15. Supply Chain Management, Transparency, & Accountability
16. Threat & Vulnerability Management

The latest version of the document may be downloaded at the following site:

https://cloudsecurityalliance.org/download/cloud-controls-matrix-v3-0-1/

A.10.2 Clauses of the Document that are Useful in the ARA Process

- The matrix identifies the architectural relevance for each control by identifying its applicability to Physical, Network, Compute, Storage, Application, and Data assets.
- Applicability to SaaS, PaaS, and IaaS cloud models are identified for each control.
- Other common frameworks are also cross-referenced for convenience.

Annex B

B ARA Process Summary Chart

The following chart provides a single page ARA process summary.