Draft NIST Special Publication 800-154

Guide to Data-Centric System Threat Modeling

Murugiah Souppaya
Karen Scarfone

COMPUTER SECURITY

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Draft NIST Special Publication 800-154

Guide to Data-Centric System Threat Modeling

Murugiah Souppaya
Computer Security Division
Information Technology Laboratory

Karen Scarfone
Scarfone Cybersecurity
Clifton, VA

March 2016

U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Willie May, Under Secretary of Commerce for Standards and Technology and Director

Authority

This publication has been developed by NIST in accordance with its statutory responsibilities under the Federal Information Security Modernization Act (FISMA) of 2014, 44 U.S.C. § 3541 *et seq.*, Public Law (P.L.) 113-283. NIST is responsible for developing information security standards and guidelines, including minimum requirements for federal information systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130.

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official. This publication may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright in the United States. Attribution would, however, be appreciated by NIST.

National Institute of Standards and Technology Special Publication 800-154
Natl. Inst. Stand. Technol. Spec. Publ. 800-154, 25 pages (March 2016)
CODEN: NSPUE2

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at http://csrc.nist.gov/publications.

Public comment period: *March 14, 2016* through *April 15, 2016*

All comments are subject to release under the Freedom of Information Act (FOIA).

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
Email: 800-154comments@nist.gov

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Abstract

Threat modeling is a form of risk assessment that models aspects of the attack and defense sides of a particular logical entity, such as a piece of data, an application, a host, a system, or an environment. This publication examines data-centric system threat modeling, which is threat modeling that is focused on protecting particular types of data within systems. The publication provides information on the basics of data-centric system threat modeling so that organizations can successfully use it as part of their risk management processes. The general methodology provided by the publication is not intended to replace existing methodologies, but rather to define fundamental principles that should be part of any sound data-centric system threat modeling methodology.

Keywords

data security; information security; risk assessment; risk management; threat modeling; threats; vulnerabilities

Trademark Information

All trademarks or registered trademarks belong to their respective organizations.

Acknowledgments

The authors, Murugiah Souppaya of the National Institute of Standards and Technology (NIST) and Karen Scarfone of Scarfone Cybersecurity, wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content.

Table of Contents

List of Appendices

Executive Summary

Threat modeling is a form of risk assessment that models aspects of the attack and defense sides of a particular logical entity, such as a piece of data, an application, a host, a system, or an environment. The fundamental principle underlying threat modeling is that there are always limited resources for security and it is necessary to determine how to use those limited resources effectively. There are many types of threat modeling; for example, system threat modeling is threat modeling performed for operational systems to improve their overall security. This publication focuses on one type of system threat modeling: data-centric system threat modeling. Data-centric system threat modeling is threat modeling that is focused on protecting particular types of data within systems.

Threat modeling is needed because of the dynamic nature of security. The attack and defense sides of security are constantly changing. As part of handling this change, organizations should continually reassess and evolve their defenses. This includes adopting continuous monitoring practices, security automation technologies, and threat intelligence feeds to detect new vulnerabilities and attacks in near-real-time, allowing rapid risk mitigation. Another key component of handling the constant change in security is having security metrics; these can be used for more informed decision making, again often relating to risk management in general and risk mitigation in particular.

Increasingly, simply following general “best practices” for security is insufficient for safeguarding high-value data. Best practices are largely based on conventional wisdom intended to mitigate common threats and vulnerabilities. By their very nature, such best practices are generalized, especially for ubiquitous products (web browsers, server and desktop operating systems, etc.) They do not take into account the unique characteristics of each system. Also, most best practices are geared toward preventing host compromise and do not take into account the security needs for particular data (again, a more generalized goal versus a specific one). So, for a particular situation, best practices may not include security controls that are necessary to effectively reduce risk.

Data-centric system threat modeling allows organizations to consider the security needs of each case of interest, instead of relying solely on generalized “best practice” recommendations. Organizations with strong capabilities in continuous monitoring, security automation, and security metrics should consider adding data-centric system threat modeling based on the principles presented in this publication to supplement these capabilities and achieve demonstrably better security for data of particular importance.

1. Introduction

1.1 Purpose and Scope

Organizations often plan, implement, maintain, and assess the security controls for their systems without performing a methodical analysis of risk involving system threat modeling. The purpose of this publication is to provide information on the basics of system threat modeling so that organizations can successfully use it as part of their risk management processes.

There are many forms of threat modeling. This publication's scope is limited to data-centric system threat modeling, which involves focusing on the security of particular instances of data (such as client information stored on a field agent's laptop) instead of focusing on the security of particular hosts, operating systems (OS), applications, etc.

This publication provides a general methodology that organizations can use. The intent is not to replace existing methodologies, but rather to define fundamental principles that should be part of any sound data-centric system threat modeling methodology.

1.2 Audience

This document is intended for security managers, security engineers/architects, system administrators, and others who are responsible for planning, implementing, and maintaining data and system security controls. Auditors and others who need to assess the security of data and systems may also find this publication useful.

1.3 Document Structure

The remainder of this document is organized into the following major sections and appendices:

- Section 2 discusses attack and defense basics. The terminology and concepts defined in this section are fundamental to understanding the rest of the publication.
- Section 3 provides an introduction to general system threat modeling.
- Section 4 presents a basic methodology for data-centric system threat modeling, with simplified examples illustrating the use of the methodology.
- Appendix A contains an acronym and abbreviation list.
- Appendix B lists the references for the document.

2. Attack and Defense Basics

This section establishes a foundation for the rest of the document by covering the basic concepts of security relevant to threat modeling. It defines fundamental terminology, such as what vulnerabilities and attack vectors are, because of the lack of consensus in the security community as to what such terms mean. It also explains how these concepts work in practice. The discussion is divided into two parts: the attack side (Section 2.1) and the defense side (Section 2.2). This section can be thought of as explaining the problem that threat modeling is trying to solve. The term "threat modeling" is defined in Section 3.

2.1 The Attack Side

This section defines the basic concepts related to the attack side of threat modeling, grouped by core terms: *vulnerability*, *exploit and attack*, *attack vector*, and *threat*. Where applicable, it enumerates the major categories of each entity (such as major categories of vulnerabilities). These enumerations are not intended to be comprehensive or authoritative, but instead to help illustrate the potential scope of threat modeling activities.

2.1.1 Vulnerability

The term "vulnerability" has been defined in many ways over years. This document proposes that a *vulnerability* is any trust assumption involving people, processes, or technology that can be violated in order to exploit a system. Types of vulnerabilities include the following:

- A *software flaw vulnerability* is caused by an error in the design or coding of software. One example is an input validation error, such as user-provided input being trusted, and thus not being properly evaluated for malicious character strings and overly long values associated with known attacks. Another example is a race condition error that allows the attacker to perform a specific action with elevated privileges. A race condition is possible because the software does not expect certain patterns of activity to occur, in effect trusting that users will not cause such patterns.
- A *security configuration issue vulnerability* involves the use of security configuration settings that negatively affect the security of the software if taken advantage of by users. A security configuration setting is an element of a software's security that can be altered through the software itself. Examples of settings are an operating system offering access control lists that set user privileges for files, and an application offering a setting to enable or disable the encryption of sensitive data stored by the application. Many configuration settings increase security at the expense of reducing functionality, so using the most secure settings could make the software useless or unusable.
- A software feature is a functional capability provided by software. A *software feature misuse vulnerability* is a vulnerability in which the feature also provides an avenue to compromise the security of a system. These vulnerabilities are caused by the software designer making trust assumptions that permit the software to provide beneficial features, while also introducing the possibility of someone violating the trust assumptions to compromise security. For example, email client software may contain a feature that renders HTML content in email messages. An attacker could craft a fraudulent email message that contains hyperlinks that, when rendered in HTML, appear to the recipient to be benign, but actually take the recipient to a malicious web site when they are clicked on. One of the trust assumptions in the design of the HTML content rendering feature was that users would not receive malicious hyperlinks and click on them. Another example of a software feature misuse vulnerability is an attacker stealing a user's credentials and reusing them to impersonate the user; the trust assumption was that only the legitimate user would use those credentials. Misuse vulnerabilities are inherent in software

features because each feature is based on trust assumptions—and those assumptions can be broken, albeit involving significant cost and effort in some cases. [1]

No system is 100 percent secure: every system has vulnerabilities. At any given time, a system may not have any known software flaws, but security configuration issues and software feature misuse vulnerabilities are always present and cannot even be readily enumerated at this time. A system's vulnerabilities are likely to have a wide variety of characteristics. Some will be very easy to exploit, while others will only be exploitable under a combination of highly unlikely conditions. One vulnerability might provide root-level access to a system, while another vulnerability might only permit read access to an insignificant file.

2.1.2 Exploit and Attack

To *exploit* a vulnerability is to use it to violate security objectives, such as confidentiality, integrity, and availability (see Section 2.2.3 for more information on security objectives). The program code or other commands used to exploit a vulnerability are generically referred to as an *exploit* or an *attack*. These meanings, which are specific to the commands, are not to be confused with the verb forms of "exploit" and "attack", which have different meanings; "to exploit" implies a successful security violation, while "to attack" implies an attempted security violation but not its success or failure. Noun forms of these verbs, referring to the actions, have the same distinction in meanings; an attack (action) that succeeds can also be called an exploit.

This document uses the term *attacker* to refer to a party who attacks a host, network, or other IT resource. However, not all attacks are intentional. Some attacks are performed by users who accidentally or otherwise unintentionally violate security policies, requirements, etc., to the point of compromising security. Because intent often has no relation to the impact of a compromise, this document uses the term "attack" to refer to both intentional and inadvertent compromises. Sections 2.1.2.1 and 2.1.2.2 discuss the individuals who perform attacks in both categories.

2.1.2.1 Intentional

Attackers who intend to exploit vulnerabilities are motivated by various reasons, ranging from the desire to make political or social statements to financial gain and cyberwarfare. Some attackers are focused on a short-term action, such as a financial transaction, but many attackers, especially those interested in gaining access to sensitive information, may be more interested in long-term infiltration and data gathering. These attacks are often targeted, such as focusing on exploiting a high-value system or individual.

The skill sets of attackers vary as widely as their motivations. At one extreme are unskilled individuals who purchase attack toolkits that make basic exploitation almost trivial to perform. At the other extreme are highly skilled individuals who are capable of discovering new vulnerabilities on their own and figuring out how to exploit them. Another important group of attackers to consider is malicious insiders; even though they may not be particularly skilled in exploitation, their level of access and detailed knowledge of the organization's systems makes them particularly effective at data theft and manipulation. A malicious insider may also work in conjunction with an external attacker, such as insiders selling their usernames and passwords to third parties.

Another category of intentional attacks is not directly associated with a particular person or group—for example, malware may have been propagating from system to system for some time and is not being directed or otherwise controlled by anyone. This is not meant to imply that no one is responsible for the malware, but rather that there is not a person specifically launching each instance of the malware.

Because the malware is designed to exploit vulnerabilities, this publication considers all malware-based attacks to be intentional attacks.

2.1.2.2 Inadvertent

Attackers who inadvertently exploit vulnerabilities are acting either by accident or through a lack of understanding, such as performing actions that they do not know are security violations or do not consider a "real" security problem.

2.1.3 Attack Vector

An *attack vector* is a segment of the entire pathway that an attack uses to access a vulnerability. Each attack vector can be thought of as comprising a *source* of malicious content, a potentially vulnerable *processor* of that malicious content, and the nature of the malicious *content* itself. Examples of attack vectors are:

- Malicious web page content (content) downloaded from a web site (source) by a vulnerable web browser (processor);
- A malicious email attachment (content) in an email client (source) rendered by a vulnerable helper application (processor);
- A malicious email attachment (content) downloaded from an email server (source) to a vulnerable email client (processor);
- A network service with inherent vulnerabilities (processor) used maliciously (content) by an external endpoint (source);
- Social engineering-based conversation (content) performed by phone from a human attacker (source) to get a username and password from a vulnerable user (processor);
- Stolen user credentials (content) typed in by an attacker (source) to a web interface for an enterprise authentication system (processor); and
- Personal information about a user harvested from social media (content) entered into a password reset website by an attacker (source) to reset a password by taking advantage of weak password reset processes (processor).

The characteristics of attacks vary widely. Some involve exploitation of a single vulnerability using a single attack vector, while others involve multiple vulnerabilities and multiple attack vectors, or even a single vulnerability and multiple attack vectors. And the vulnerabilities and attack vectors may be spread across multiple hosts, compromising one host in order to compromise another, further complicating the composition of an attack.

Here is an example of a single possible attack involving one vulnerability, decomposed into its attack vectors:

1. Malicious email attachment (content) sent from a host (source) to the organization's primary mail server (processor);

2. Malicious email attachment (content) sent from the organization's primary mail server (source) to antivirus server (processor);

3. Malicious email attachment (content) sent from the antivirus server (source) to the organization's internal mail server (processor);

4. Malicious email attachment (content) sent from the organization's internal mail server (source) to the user's email client (processor); and

5. Malicious email attachment (content) rendered (processor) by the vulnerable email client (source).

Note that although there are five attack vectors, the actual exploitation only occurs during the last of the five (when the malicious attachment is rendered by a vulnerable email client). However, each attack vector affords an opportunity to detect and stop the attack before it goes any farther.

Although this example focuses on vulnerabilities in technologies, many attacks include non-technological attack vectors. For example, attackers often use social engineering methods to trick users into revealing their passwords, performing actions that unknowingly grant attackers remote access to systems, and otherwise enabling security to be compromised. Similar attacks may be performed on IT personnel, such as an attacker impersonating a legitimate user and convincing a help desk agent to reset the user's password to a password selected by the attacker.

Because the focus of this publication is modeling threats against a targeted (vulnerable) system, the compromises of greatest interest are those against the ultimate target itself. There are often intermediate hosts used as jumping off points for attacks—in botnets, for example. Analyzing how those intermediate hosts become compromised would fall within the scope of performing an analysis of those individual intermediate hosts themselves as targets, and is outside the scope of analyzing the ultimate target. So, in the previous example with five attack vectors, with the ultimate target being the host with the vulnerable email client, it would be irrelevant to the target how the host in step 1 that originally sent the malicious email attachment was compromised.

Other terms are useful when discussing attack vectors. For example, an *attack model* comprises a scenario that may occur and a single path (one or more attack vectors in sequence) that could be taken for that scenario. The attack models for a scenario and the security controls attempting to disrupt those attack models collectively constitute a threat model.[1] Another way to analyze attack vectors is to analyze all of the attack vectors directly against a particular system; this is known as the system's *attack surface*.

2.1.4 Threat

A *threat* is defined in NIST Special Publication (SP) 800-30 as "any circumstance or event with the potential to adversely impact organizational operations and assets, individuals, other organizations, or the Nation through an information system via unauthorized access, destruction, disclosure, or modification of information, and/or denial of service." [2, p. B-13] Threats may be intentional or unintentional. A *threat source* is the cause of a threat, such as a hostile cyber or physical attack, a human error of omission or commission, a failure of organization-controlled hardware or software, or other failure beyond the control of the organization.

---

[1] Section 3 contains a more formal definition of the term "threat modeling."

A *threat event* is defined in NIST SP 800-30 as “an event or situation initiated or caused by a threat source that has the potential for causing adverse impact.” [2, p. B-13] The distinction between a threat and a threat event is subtle, but basically a threat event is caused by a particular threat source, while a threat is more generic (not caused by a particular threat source).

2.2 The Defense Side

This section explains the basic concepts related to the defense side of threat modeling, grouped by core terms: *risk*, *security controls*, and *security objectives*.

2.2.1 Risk

Committee on National Security Systems Instruction (CNSSI) 4009, *National Information Assurance (IA) Glossary*, defines *risk* in general as “a measure of the extent to which an entity is threatened by a potential circumstance or event, and typically a function of: (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence” [3, p. 104] and *information security risks* specifically as “those risks that arise from the loss of confidentiality, integrity, or availability of information or information systems and reflect the potential adverse impacts to organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, and the Nation” [3, p. 65].

Risk management is defined in CNSSI 4009 as “the program and supporting processes to manage information security risk to organizational operations….” [3, p. 104]. Part of risk management is *risk assessment*, which is defined in NIST SP 800-30 as “the process of identifying, prioritizing, and estimating risks to organizational operations (including mission, functions, image, reputation), organizational assets, individuals, other organizations, and the Nation, resulting from the operation of an information system” [2, p. B-9]. Risk assessment considers the possible threats and vulnerabilities, and determines what security controls (see Section 2.2.2) should be used to *mitigate* them, which means to reduce their risk to an acceptable level.

2.2.2 Security Controls

CNSSI 4009 defines *security controls* as “the management, operational, and technical controls (i.e., safeguards or countermeasures) prescribed for an information system to protect the confidentiality, integrity, and availability of the system and its information” [3, p. 110]. Although technical controls can be fully automated, which makes them an obvious choice for stopping attacks, management and operational controls also play important roles. For example, users must be trained on their security responsibilities so that they are less likely to violate security policies, be tricked by phishing attacks, and otherwise decrease the organization’s security posture. Ultimately an organization’s security relies on a combination of people, processes, and technology.

2.2.3 Security Objectives

Organizations usually think of their security objectives for data in terms of protecting its confidentiality[2], integrity, and/or availability.[3] In many cases, the security objectives for an instance of data should not all have equal importance, and in some cases, an organization may want to focus its threat modeling efforts

[2] For the purposes of this publication, privacy objectives will be considered a subset of confidentiality objectives.

[3] Some organizations may choose to replace the security objectives with one or more security requirements for threat modeling purposes. Security requirements are more specific and granular than security objectives, and they often come directly from the organization’s policies or from regulations or security compliance initiatives that the organization is subject to.

on a single objective. For example, if a risk assessment shows that the risk of a breach of confidentiality is unacceptably high, then performing threat modeling for confidentiality only may be most helpful for mitigating the confidentiality breach risk to an acceptable level. Similarly, information that has already been released to the public may still need its integrity and availability protected, but not its confidentiality.

Because this publication is addressing data-centric system threat modeling, only security objectives for data—not systems—are pertinent. This means that operational unavailability of systems caused by attacks is out of scope, for example, while operational unavailability of data caused by attacks is in scope.

The security objectives relate directly to the risk management, assessment, mitigation, and security control concepts discussed in Sections 2.2.1 and 2.2.2. For example, if the results of risk assessment show that the risk of a breach of confidentiality is unacceptably high, then additional security controls or changes to existing security controls may be needed to mitigate the confidentiality breach risk to an acceptable level. The same is true for integrity and availability.

3. Introduction to System and Data-Centric System Threat Modeling

This section provides an overview of threat modeling in general, with an emphasis on system and data-centric system threat modeling. *Threat modeling* is a form of risk assessment that models aspects of the attack and defense sides of a particular logical entity, such as a piece of data, an application, a host, a system, or an environment. The fundamental principle underlying threat modeling is that there are always limited resources for security and it is necessary to determine how to use those limited resources effectively.

There are many types of threat modeling, and they can be distinguished based on three related characteristics:

1. The *logical entity* being modeled (data, software, system, etc.);
2. The *phase* of the system lifecycle (for example, modeling security for software during its initial design versus modeling security for already-implemented off-the-shelf software, for example); and
3. The *goal* of the threat modeling (to reduce software vulnerabilities, to thwart particular classes of attackers, to improve overall system security, to protect particular types of data, etc.).

A common form of threat modeling is *software threat modeling*, which is threat modeling performed during software design to reduce software vulnerabilities. There are many established methodologies for performing software threat modeling.

Another common form of threat modeling is known as *system threat modeling*, which is threat modeling performed for operational systems to improve their overall security. Compared to software threat modeling, system threat modeling tends to be largely informal and ad hoc.

Data-centric system threat modeling is a particular type of system threat modeling, and its basics are described in Section 4. This type of threat modeling is focused on protecting particular types of data within systems.

Threat modeling is needed because of the dynamic nature of security. Security would be difficult enough to tackle if it was a one-time endeavor. Unfortunately, the attack side is constantly changing; new vulnerabilities are discovered, new attacks are created, and new threats arise. Long-term changes happen, too—new classes of vulnerabilities are discovered, attacker motivations change, and other transformations occur over years. The defense side of security is also constantly changing—security controls are improved and enhanced, new types of security controls are developed, etc. Change is inevitable; as a particular class of vulnerabilities becomes well mitigated, for example, attackers simply identify another class of vulnerabilities that are not as well mitigated to exploit, and defenders adjust security controls accordingly.

As part of handling this constant change, organizations should continually reassess and evolve their defenses. This includes adopting continuous monitoring practices [4], security automation technologies [5], and threat intelligence feeds to detect new vulnerabilities and attack attempts in near-real-time, allowing rapid risk mitigation. Another key component of handling the constant change in security is having security metrics; these can be used for more informed decision making, again often relating to risk management in general and risk mitigation in particular.

Organizations with strong capabilities in continuous monitoring, security automation, and security metrics should consider adding data-centric system threat modeling as described in Section 4 to supplement these capabilities and achieve demonstrably better security for data of particular interest. Quantitative security metrics are more accurate than qualitative ones, but quantitative metrics are presently very difficult for most organizations to collect. Using high-quality qualitative metrics is far better than using no metrics at all.

Increasingly, simply following general "best practices" for security is insufficient for safeguarding high-value data. Best practices are largely based on conventional wisdom intended to mitigate common threats and vulnerabilities. By their very nature, such best practices are generalized, especially for ubiquitous products (web browsers, server and desktop operating systems, etc.) They do not take into account the unique characteristics of each system. Also, most best practices are geared toward preventing host compromise and do not take into account the security needs for particular data (again, a more generalized goal versus a specific one). So, for a particular situation, best practices may omit security controls that are necessary to effectively reduce risk.

Data-centric system threat modeling allows organizations to consider the security needs of each case of interest, instead of relying solely on "best practice" generalized recommendations. Organizations are already very familiar with applying best practices to operating systems and individual applications, such as securing a web server (host) or web server software. What is considerably more challenging for organizations to tackle is determining how to secure a particular chunk of data. It is not that securing a piece of data is so difficult, but that traditionally security professionals, system administrators, and others responsible for securing operational systems have focused on securing systems, not data. The rest of this publication focuses on data security.

This differentiation between system and data security has parallels to existing NIST publications. A system security approach starts with a FIPS 199 [6] low/moderate/high impact rating for a system, and then selects the corresponding security controls from NIST SP 800-53 [7]. In contrast, a data security approach starts with a publication such as NIST SP 800-60 [8] for categorizing the type of information.

4. Basics of Data-Centric System Threat Modeling

Data-centric system threat modeling brings together the attack and defense side information for data of interest in a standardized model that facilitates security analysis, decision making, and change planning. This section provides information on the fundamentals of data-centric system threat modeling. This publication is not trying to define a new threat modeling methodology, but rather to educate organizations on the fundamentals of data-centric system threat modeling and to make recommendations related to the use of this type of modeling.

The data-centric system threat modeling approach presented in this publication has four major steps:

1. Identify and characterize the system and data of interest;
2. Identify and select the attack vectors to be included in the model;
3. Characterize the security controls for mitigating the attack vectors; and
4. Analyze the threat model.

Sections 4.1 through 4.4 provide more information on performing each of these steps. Each step is also illustrated by examples, which are denoted by a border around the text. The same example is continued throughout all of the steps. Note that the data presented in the example is hypothetical and meant solely for providing a simplified illustration of the steps.

Section 4.5 explains in detail that this approach to data-centric system threat modeling is intended to be customized to meet the needs of each organization, and it shows how easily this customization can occur.

4.1 Step 1: Identify and Characterize the System and Data of Interest

The first step is to identify and characterize the system and data of interest. The system and data should be defined narrowly, pertaining to a particular logical set of data on a particular host or small group of closely related hosts and devices.

Once the system and data are defined, they need to be characterized, which refers to understanding the system's operation and usage to the extent needed for the organization's data-centric system threat modeling approach. **At an absolute minimum, characterization should include the following:**

- **The authorized locations for the data within the system.**[4] This will include some or all of the following:
  - *Storage*: all places where data may be at rest within the system boundaries;
  - *Transmission*: all ways in which data may transit over networks between system components and across the system's boundaries;
  - *Execution environment*: e.g., data held in local memory during runtime, data processed by virtual CPUs, etc.;
  - *Input*: e.g., data typed in using the keyboard; and

[4] The methodology identifies just the authorized locations because someone with access to the data could store, transmit, output, or otherwise place the data in any accessible location, authorized or not, including locations outside the system boundaries.

    - *Output*: e.g., data printed to a physically attached printer, data displayed on the laptop screen, etc.

- **A basic understanding of how the data moves within the system between authorized locations.** For example, a file might be held in memory while it is being created and is only written out to storage when the user directs the system to do so. Depending on the complexity of the system, gaining this understanding may require first understanding the system's functions and processes, users and usage scenarios, workflows, trust assumptions, and other aspects of people, processes, and technology related to the system.
- **The security objectives (e.g., confidentiality, integrity, availability) for the data.** In many cases, some objectives are more important than others; in other cases, an organization may want to focus on a single objective for a particular threat model.[5]
- **The people and processes who are authorized to access the data in a way that could affect the security objectives.** For example, if an organization has selected confidentiality as its sole objective for a particular threat model, the authorized people and processes should include all users, administrators, applications, services, etc. who are allowed to read the data.

Example Scenario

Summary: The data of interest is a spreadsheet containing personally identifiable information (PII) for employees who have received workers' compensation.

The system of interest comprises:

- a human resource specialist's laptop (spreadsheet is stored on and used from the laptop);
- a USB flash drive (spreadsheet is backed up onto the USB flash drive); and
- a printer (spreadsheet can be printed from the laptop to the printer).

The authorized locations for the data of interest are as follows:

- Storage: Spreadsheet kept on a laptop hard drive, backup of spreadsheet kept on a USB flash drive;
- Transmission: Sent to a printer over a wireless network;
- Execution environment: Local laptop memory and processors;
- Input: Typed in using the laptop keyboard; and
- Output: Displayed to the screen.

Description: Data is input through the keyboard into the spreadsheet, which is temporarily held in the execution environment. As the user updates the spreadsheet, the data is displayed to the screen. When the user has completed editing the spreadsheet, the user directs the system to save the spreadsheet to the laptop hard drive. The user may also load the spreadsheet into the execution environment and print the spreadsheet to a nearby printer through a wireless network connection. Finally, the user occasionally copies the latest version of the spreadsheet from the laptop hard drive to a USB flash drive as a backup.

[5] Some organizations may choose to replace the security objectives with one or more security requirements. Security requirements are more specific and granular than security objectives, and they often come directly from the organization's policies or from regulations or security compliance initiatives that the organization is subject to.

Although confidentiality, integrity, and availability all matter for the data of interest, confidentiality is considered so much more important that the organization has decided to perform its trust modeling in terms of confidentiality only.

In this highly simplified example, the human resource specialist is the only person who is authorized to access the data.

4.2 Step 2: Identify and Select the Attack Vectors to Be Included in the Model

The second step involves identifying the potential attack vectors, as discussed in Section 2.1.3, that could be used to negatively impact one or more of the identified security objectives for one of the authorized locations for the data. Once the attack vectors are identified, it may be necessary to select only a subset of those attack vectors to be included in the threat model. Although it is generally preferable to include all the attack vectors, there are often too many to be addressed with limited resources. Possible criteria to consider include the relative likelihood of the attack vector being used and the most likely impact of a successful attack. See Section 4.5 below for more information on the selection process.

Location 1: Stored in a spreadsheet on the local hard drive.

- *Vector 1a*: Attacker gains unauthorized physical access to the laptop, uses forensic tools or other utilities to copy the file (without authenticating to the OS).
- *Vector 1b*: Attacker gains unauthorized physical access to the laptop, exploits vulnerabilities to gain OS access (impersonating user/admin).
- *Vector 1c*: Attacker steals and reuses user/admin/service credentials.
- *Vector 1d*: Attacker gains access to/control over user’s session/device.
- *Vector 1e*: User forwards the file to an unauthorized recipient (user was tricked via social engineering, user is malicious, user made a mistake, etc.).
- *Vector 1f*: Attacker accesses unsecured network service (e.g., connects to unsecured file share) and gains access to the file.

Location 2: Stored in a spreadsheet on a flash drive backup.

- *Vector 2a*: Attacker gains unauthorized physical access to the flash drive, mounts the drive and copies the file.
- *Vector 2b*: Attacker steals and reuses user/admin/service credentials for laptop while flash drive is mounted.
- *Vector 2c*: Attacker gains access to/control over user’s session/device while flash drive is mounted.
- *Vector 2d*: User forwards the file to an unauthorized recipient.

Location 3: Printed to a nearby printer over a wireless network connection.

- *Vector 3a*: Attacker monitors unencrypted or weakly encrypted wireless network communications and captures the data being sent to the printer.
- *Vector 3b*: Attacker views a printout of the spreadsheet.

Location 4: Processed locally.

- *Vector 4a*: Attacker gains access to/control over user's session/device.

Location 5: Input locally.

- *Vector 5a*: Attacker watches the information being typed in to the laptop.
- *Vector 5b*: Attacker uses keystroke logger on laptop to monitor keystrokes.

Location 6: Output locally.

- *Vector 6a*: Attacker views the information on the laptop screen.
- *Vector 6b*: Attacker uses malware on laptop to take screen shots.

Selected attack vectors (based on the possibility and the likelihood of each attack vector being used to completely compromise confidentiality):

- *Vector 1c*: Data is stored in a spreadsheet on the local hard drive; attacker steals and reuses user/admin/service credentials.
- *Vector 1d*: Data is stored in a spreadsheet on the local hard drive; attacker gains access to/control over user's session/device.
- *Vector 2b*: Data is stored in a spreadsheet on a flash drive backup; attacker steals and reuses user/admin/service credentials for laptop while flash drive is mounted.
- *Vector 2c*: Data is stored in a spreadsheet on a flash drive backup; attacker gains access to/control over user's session/device while flash drive is mounted.
- *Vector 4a*: Data is processed locally; attacker gains access to/control over user's session/device.

4.3 Step 3: Characterize the Security Controls for Mitigating the Attack Vectors

The third step of the methodology is, for each attack vector selected in Step 2, to identify and document security control alterations (additions to existing security controls, reconfigurations of existing security controls, etc.) that would help mitigate the risk associated with the attack vector and that are reasonably feasible to accomplish. Note that it is not necessarily to enumerate every single applicable control, such as having a security program and policies, because these controls should already be in place for the entire enterprise and are not normally customized to take a particular attack vector into consideration.

Next, for each selected security control alteration, estimate how effectively it would address exploitation of each applicable attack vector. This could be as simple as ranking effectiveness as low, medium, or high, or as complex as estimating the percentage of attacks against the attack vector that would be stopped by this mitigation. Whatever method is selected, it should be comparable across mitigations and attack vectors.

The counterpart to estimating the effectiveness of each security control alteration is estimating the negative implications. Factors to consider may include cost (perhaps the order of magnitude or the range of costs for acquisition and implementation, and for annual management/maintenance) and reductions in functionality, usability, and performance. These can be particularly hard to determine accurately for future mitigations, so it may be necessary to develop very rough estimates for them using a simple low/medium/high type scale that is particular to the organization.

Feasible security control alterations:

1. Require strong password with strongly encrypted password hash (vectors *1c* and *2b*).

- Effectiveness: Low
- Acquisition and implementation costs: Low
- Annual management/maintenance costs: Low
- Impact on functionality: Low
- Impact on usability: Low
- Impact on performance: Low

2. Require multifactor authentication (vectors *1c* and *2b*)

- Effectiveness: High
- Acquisition and implementation costs: Moderate
- Annual management/maintenance costs: Moderate
- Impact on functionality: Low
- Impact on usability: Moderate
- Impact on performance: Low

3. Use antivirus software, spam filtering, real-time blacklists, user awareness, web reputation software, etc. (vectors *1c*, *1d*, *2b*, *2c*, and *4a*)

- Effectiveness: Moderate
- Acquisition and implementation costs: Moderate
- Annual management/maintenance costs: Moderate
- Impact on functionality: Moderate
- Impact on usability: Moderate
- Impact on performance: Moderate

4. Patch vulnerabilities (vectors *1c*, *1d*, *2b*, *2c*, and *4a*)

- Effectiveness: Low
- Acquisition and implementation costs: Moderate
- Annual management/maintenance costs: Moderate
- Impact on functionality: Moderate
- Impact on usability: Low
- Impact on performance: Moderate

4.4 Step 4: Analyze the Threat Model

The final step of the methodology is to analyze all the characteristics documented during the previous steps, which collectively comprise the threat model, to assist in evaluating the effectiveness and efficiency of each security control option against the selected attack vectors. It is far too simplistic to say that a control should be applied because it lowers risk. In addition to their financial costs in terms of acquisition, implementation, and management/maintenance, security controls can negatively impact functionality, usability, and performance, among other factors. Any assessment of security controls should include considerations of all significant relevant factors.

The most challenging part of threat model analysis is determining how to consider all of these characteristics together. It is straightforward to compare an individual characteristic, such as the annual management/maintenance costs, across attack vectors and mitigations. However, it is not straightforward at all to compare the entire set of characteristics for an attack vector against the entire set of characteristics for another attack vector. Yet such comparisons are absolutely critical to determining how risk can best be reduced across all the attack vectors, in a cost-effective manner that has an acceptable negative impact on the organization's operations. Each organization needs to determine how to compare the characteristics for each attack vector/security control pair, as a basis for comparing attack vector characteristics and security control characteristics.

One approach to facilitating these comparisons is to assign scores and weightings to each characteristic. For example, the narrative descriptions of threat consequence could be converted to numerical values on a three-point scale. Three-point scales could also be used for other characteristics in place of low, medium, and high ratings. Even complex characteristics, such as cost, could be mapped to a simple scale.

In addition to assigning scores to each characteristic's possible values or value ranges, the organization also needs to consider the relative weights of each characteristic. Perhaps the effectiveness against attacks is considered much more important than other characteristics; if so, this could be conveyed by doubling or tripling its score. Similarly, all other characteristics could be assigned a multiplier that would increase or decrease their scores or keep them the same. Then after applying the multipliers, the organization would add up the results and have a relative score for each attack vector/security control pair.

Another scoring approach that could be followed in addition to the previously described approach is to set thresholds or rules for certain criteria and eliminate from further consideration any attack vector/security control pairs that do not meet these. A simple example is eliminating all pairs that have a cost of $100,000 or more over a period of three years. A more complex example is eliminating all pairs that have a cost of $50,000 or more over a period of three years AND a high impact on usability AND low or medium effectiveness against attacks.

After much debate, the organization decides to set the following scores for the characteristics and weigh them all evenly:

- No security control effectiveness = 0
- Security control effectiveness of low = 1
- Security control effectiveness of moderate = 2
- Security control effectiveness of high = 3
- Negative implication of high = 1

- Negative implication of moderate = 2
- Negative implication of low = 3

The organization calculates the total of the negative implication scores for each security control (see first table below), then multiplies these totals by the score of the security control effectiveness per attack vector (see second table below) to reach a score for each attack vector/security control pair (see shaded area of the second table below). The higher the score, the more "bang for the buck" the security control will provide against the corresponding attack vector. This information is now ready for use in decision making.

Possible Security Controls	Acquisition and Implementation Costs	Annual Management/ Maintenance Costs	Impact on Functionality	Impact on Usability	Impact on Performance	Total for Security Control
Require strong password with strongly encrypted password hash	3	3	3	3	3	**15**
Require multifactor authentication	2	2	3	2	3	**12**
Use antivirus software, spam filtering, real-time blacklists, user awareness, web reputation software, etc.	2	2	2	2	2	**10**
Patch vulnerabilities	2	2	2	3	2	**11**

Possible Security Controls	Security Control Effectiveness Per Attack Vector					Negative Implication Total	Security Control Effectiveness Times Negative Implication Total Per Attack Vector				
	1c	1d	2b	2c	4a		1c	1d	2b	2c	4a
Require strong password with strongly encrypted password hash	1	0	1	0	0	15	15	0	15	0	0
Require multifactor authentication	3	0	3	0	0	12	36	0	36	0	0
Use antivirus software, spam filtering, real-time blacklists, user awareness, web reputation software, etc.	2	2	2	2	2	10	20	20	20	20	20
Patch vulnerabilities	1	1	1	1	1	11	11	11	11	11	11

4.5 Customizing the Data-Centric System Threat Modeling Approach

This publication presents a primarily qualitative approach to data-centric system threat modeling. A quantitative approach would lead to more precise and accurate results than a qualitative approach, but quantitative approaches would also be much more resource-intensive and would not scale well for modeling large and complex systems unless the metrics and methodologies are mostly automated. Because such automation is not yet widely available, if at all, this publication focuses on qualitative modeling, which is still quite beneficial. In the future, as more automated quantitative metrics and methodologies become available, organizations should reconsider the feasibility of using quantitative modeling.

Most of the actions within the methodology can be addressed in a wide variety of ways in terms of both content (what information is captured) and format/structure (how that information is captured). There is no “right” method, and the examples are purely illustrative. What is important is recording sufficient information to provide the necessary input for subsequent steps and a basis for making actionable recommendations.

A prime example of the flexibility of the methodology is Step 2. Step 2 uses the list from Step 1 of authorized locations for the data of interest. In the example, each attack vector is defined in a narrative way, such as “Attacker gains unauthorized physical access to the laptop, uses forensic tools or other utilities to copy the file (without authenticating to the OS).” This single statement actually conveys three pieces of data: 1) a source of malicious content, 2) a potentially vulnerable processor of that malicious content, and 3) the nature of the malicious content itself.

Some organizations may prefer a more narrative approach to defining attack vectors because it is easier for others to understand, while other organizations may want a more thorough or technically-based approach and therefore want to go through the threat consequences and actions as a taxonomy for identifying the attack vectors. And, of course, there are many other ways of defining attack vectors that individual organizations may prefer to use because of existing processes and tools or for other reasons. Another factor to consider is the granularity of the attack vectors; one organization may only have the resources to consider the attack vectors at a truly high level, while another organization may want to do a deep dive and make the attack vectors as narrow as possible.

Organizations may also want to scope their threat modeling so it takes less effort. Using Step 2 as an example, an organization may decide to eliminate any attack vectors that do not merit further consideration. For example, an organization may decide that attack vectors with the lowest relative likelihood should be ignored because there are far too many other attack vectors to be considered. Similarly, an organization may only be interested (at least initially) in attack vectors that are likely to lead to a complete compromise of confidentiality, integrity, and availability. Another possibility is to eliminate attack vectors that do not have any feasible mitigations. Ideally an organization should analyze all attack vectors before winnowing out any—for example, an unlikely attack vector may turn out to be incredibly easy and inexpensive to mitigate, or a single mitigation may address multiple attack vectors—but realistically this may not be feasible in some cases.

Of course, an organization can skip any of the elements of the methodology that are not relevant for a particular situation or environment, and likewise an organization can add characteristics if other factors are also important to the organization.

Appendix A—Acronyms and Other Abbreviations

Selected acronyms and other abbreviations used in the guide are defined below.

CMSS	Common Misuse Scoring System
CNSSI	Committee on National Security Systems Instruction
FIPS	Federal Information Processing Standard
FISMA	Federal Information Security Modernization Act
FOIA	Freedom of Information Act
HTML	Hypertext Markup Language
IA	Information Assurance
IR	Interagency Report
IT	Information Technology
ITL	Information Technology Laboratory
NIST	National Institute of Standards and Technology
OMB	Office of Management and Budget
OS	Operating System
PII	Personally Identifiable Information
RFC	Request for Comments
SCAP	Security Content Automation Protocol
SP	Special Publication
USB	Universal Serial Bus

Appendix B—References

This appendix lists the references for the document.

[1] E. LeMay, K. Scarfone, and P. Mell, National Institute of Standards and Technology (NIST) Interagency Report (IR) 7864, *The Common Misuse Scoring System (CMSS): Metrics for Software Feature Misuse Vulnerabilities*, July 2012. http://dx.doi.org/10.6028/NIST.IR.7864

[2] Joint Task Force Transformation Initiative, National Institute of Standards and Technology (NIST) Special Publication (SP) 800-30 Revision 1, *Guide for Conducting Risk Assessments*, September 2012. http://dx.doi.org/10.6028/NIST.SP.800-30r1

[3] Committee on National Security Systems (CNSS), CNSS Instruction No. 4009, *National Information Assurance (IA) Glossary*, April 6, 2015. https://www.cnss.gov/CNSS/issuances/Instructions.cfm

[4] K. Dempsey, N. Chawla, A. Johnson, R. Johnston, A. Jones, A. Orebaugh, M. Scholl, and K. Stine, National Institute of Standards and Technology (NIST) Special Publication (SP) 800-137, *Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations*, September 2011. http://dx.doi.org/10.6028/NIST.SP.800-137

[5] S. Quinn, K. Scarfone, and D. Waltermire, National Institute of Standards and Technology (NIST) Special Publication (SP) 800-117 Revision 1 (Draft), *Guide to Adopting and Using the Security Content Automation Protocol (SCAP) Version 1.2*, January 2012. http://csrc.nist.gov/publications/PubsSPs.html#800-117-R1

[6] National Institute of Standards and Technology (NIST) Federal Information Processing Standard (FIPS) Publication (PUB) 199, *Standards for Security Categorization of Federal Information and Information Systems*, February 2004. http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf

[7] Joint Task Force Transformation Initiative, National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53 Revision 4, *Security and Privacy Controls for Federal Information Systems and Organizations*, April 2013 (including updates as of January 22, 2015). http://dx.doi.org/10.6028/NIST.SP.800-53r4

[8] K. Stine, R. Kissel, W. Barker, J. Fahlsing, and J. Gulick, National Institute of Standards and Technology (NIST) Special Publication (SP) 800-60 Revision 1, *Volume 1: Guide for Mapping Types of Information and Information Systems to Security Categories*, August 2008. http://dx.doi.org/10.6028/NIST.SP.800-60v1r1