Practical Threat Modeling

BRUCE POTTER

Bruce Potter, CTO of the KEYW Corporation, has over 20 years of experience focused on tackling high-end information security research and engineering problems. During his career, Bruce has built and led teams focused on hard problems in information security such as cybersecurity risk analysis, high assurance system and network engineering, advanced software analysis techniques, wireless security, and IT operations best practices. Bruce is also the founder of The Shmoo Group, a nonprofit think tank comprising security, privacy, and crypto professionals who donate time to information security research and development. Bruce assists in the organization of ShmooCon, an annual computer security conference in Washington, D.C. Bruce has authored many publications and has delivered numerous presentations at various security and network conferences and private events, including DefCon, BlackHat USA, ShmooCon, the United States Military Academy, Johns Hopkins University, and the Library of Congress. bpotter@keywcorp.com

Threat modeling is a key component to modern-day cybersecurity risk management, but the processes of creating a threat model can be complex and overwhelming. By understanding the components that make up statements of threat, such as threat actors, assets, and malicious actions, we can turn threat modeling into a management process that can be performed by a variety of practitioners. With some practice and awareness of your enterprise, you can start creating threat models that will have a large impact on the quality of your risk management decisions.

Information security is really all about risk management. Building provably secure systems is cost prohibitive and serves as a barrier to innovation. Most modern-day systems not only operate in a constantly changing environment but are incredibly complex and diverse collections of hardware and software with many interfaces and numerous use cases.

Rather than secure all parts of a system equally, we must invest our time and resources wisely to secure a system in the places that actually matter. The idea of addressing the areas of highest concern first is really the core concept behind risk management. Without understanding the risk to the system, we might as well roll dice to determine what to focus our efforts on.

Why Is Threat Modeling Important?

There are several ways to think about risk. While distilling risk down to a simple equation has some dangers associated with it, the core concepts behind a risk equation are a useful foil to discuss risk. For our purposes, assume understanding a risk follows the equation below:

Risk = ((Threat x Vulnerability) / Countermeasure) x Impact

Understanding each of the values that go into this equation is its own discipline. For instance, to understand vulnerabilities in a system, you might employ product evaluators or penetration testers who will examine your system from top to bottom to find security vulnerabilities and document them. To understand countermeasures, you could perform a security control audit to find out where all your security controls are, if they are configured correctly, and how effective they are. Finally, to examine impact, you might meet with various business managers within your organization to better understand how critical each system is and the overall value to the business.

Understanding the threat has been a more elusive problem. The concept of threat feels like it has more basis in intelligence gathering than technical analysis. The idea of someone in an organized crime ring sitting in a dark room halfway across the world writing custom malware targeting your Web site sounds like something out of a spy movie. But understanding the threats facing your systems doesn't require you to hire a security intelligence service and go deep underground to find all the organizations wishing to do you harm.

Threat modeling is a process that is used to develop and rank specific threats against your system. The resulting threat model is a document with a similar audience as a technical risk or vulnerability assessment. The threat model can be used by developers to understand what attackers might try to do to the system when they are determining how to code defensively. The threat model can be used by system and network operators to help determine what network controls should be put in place based on potential adversarial actors. The model can even be used by management to assist in understanding the threat landscape and adjust development and IT spend.

The process of threat modeling can be very complex. Microsoft, as part of its well documented and publicly available security development life cycle, published a book [1] that documents their threat modeling process. Adam Shostack's *Threat Modeling: Designing for Security* puts forth a great process for threat modeling in the development process. The book is comprehensive and can be applied in structured development environments. However, the process described in *Threat Modeling* can be very heavyweight, especially in lean or understaffed development environments. And the process is very difficult to modify for suitability for use for non-developers. The threat modeling process put forth in this article has been influenced by a number of sources, including *Threat Modeling*. However, I have created this process to be useful and applicable to a broad audience of practitioners, not just system developers.

All that said, what exactly is a threat?

The Syntax of a Threat

In the context of threat modeling, I've found it is useful to think about threats using a very specific syntax:

$ACTOR does $ACTION to $ASSET for $OUTCOME because $MOTIVATION

A threat model is a collection of threat statements that follow this basic syntax listed in ranked order. In order to create a specific threat in a threat model, we must have specific knowledge about an instance of each variable. The process of creating a threat model involves identifying interesting values for each variable and determining which are important to your organization.

The first three variables ($actor, $action, and $asset) are somewhat self-explanatory and will be covered later in this article. The outcome is critically important to each threat. There are many actions that an adversary could take involving your system. However, if the action results in no bad outcome, there is no consequence to the action. For instance, if an attacker can anonymously log in to your FTP server but the only data on the server is public data, then there's no bad outcome. In fact, your adversary is acting with the same privilege and access as your regular users. Anonymous FTP access to your FTP server in this case is not a threat action.

The last variable, motivation, is somewhat optional. The motivation of an adversary is not necessarily of interest to every organization. Some organizations are interested in what is motivating their attackers and use that information to develop deterrence strategies. Other organizations do not have as robust an understanding of adversaries and only care about the outcome, not the motivation. As you go through this process, you will get a sense of what you and your organization cares about and can decide whether capturing the attacker's motivation is important to you.

Threat Actors

While there are specific bad actors in the world that may wish to harm your systems, you don't necessarily need to identify them by name. Rather, thinking of threat actors in broad categories helps you understand motivations and resourcing and how that would impact what they can and would do. The following five major threat actor categories are a useful starting point for you to develop an understanding of threat actors and the role they will have in your model.

Nation State

Nation state actors are very well resourced and may maintain operations for months or even years. These actors are motivated by national interests such as intelligence gathering, military action, critical infrastructure control, and industrial espionage. Nation states are generally very difficult to defend against.

Organized Crime

Organized crime actors are moderately well resourced with operations that may last for months. These actors are generally motivated by financial gain or access to information that can lead to financial gain such as personal information or credit card data. Due to the focused nature of organized crime, they can be very difficult to defend against, although the information they are interested in is often more limited than that of nation states.

Insiders

Insiders are as well resourced as you let them be. Insiders will utilize whatever access is available to accomplish their objective. Given the state of internal security of most organizations, insiders are far over-accessed and can cause great harm. Motivation for insiders can range from ideological issues to profit to revenge. Insiders are difficult to defend against as they may dedicate their lives to pursuing their objective.

Hacktivists

Hacktivists have limited resources and run operations that last weeks to months. They are generally motivated by ideological issues and target organizations very specifically. Hacktivists often publicly discuss their objectives and hide behind anonymity services such as Tor. Organizations with well-run IT operations can often defend against hacktivists, although social engineering can be the Achilles heel of enterprises under hacktivist attack.

Script Kiddies

Script kiddies have very limited resources. These adversaries are often motivated by curiosity and simple malicious intent. Their tooling often only consists of publicly available tools, hence the "script kiddie" moniker. These attackers will look for targets of opportunity and can be defended against using normal IT security best practices.

Others

As you work through several threat models, you may find that you identify specific threat actors that are unique to your organization. Maybe you have been targeted by an organized crime group in the past with particular interest in your company. Or maybe Bob from Accounting seems like he might be up to no good. Whatever the reason, feel free to add to the list of threat actors as your models evolve. However, be cautious of creating too many specific actors; if their resourcing and motivations are similar, there's little utility in splitting out multiple actors.

System Representation

The first step in the threat modeling process is to create a system representation. In Microsoft's process, the system is represented formally through Data Flow Diagrams (DFD). These DFDs capture all interfaces, assets, and data flows through very specific iconography. While Microsoft DFDs are in some contexts a universal language, they are also time-consuming to create and at such a low level to not be useful to non-developers.

Rather than completely decomposing the system you are threat modeling, capture the system in a manner that is convenient for you and your team. These might be network diagrams, system architectures, or even basic scribbling on a whiteboard. What's important in capturing the representation is that it captures the assets and capabilities you are trying to protect. For instance, if you are threat modeling a CRM (customer relationship management) solution, your system representation should include your sales force, the types of systems they use to access the solution, transport and storage mechanisms, the CRM servers themselves, and any external data sources. Whether the servers are circles or squares really doesn't matter; what does matter is that all the components of the system are represented. Ultimately, these diagrams capture all the $assets that you will use to create the threat model.

Brainstorming

The real meat of the threat modeling process is brainstorming. This is the part of the process that requires a little bit of creativity and security knowledge. Start with an asset represented in your system representation. Pick a threat actor and then think of the bad things the attacker may do to that asset and how that would affect your organization. Write down that threat in the syntax described above, and then think of another bad thing that threat actor could do…and another…and another. Write down ideas for that threat actor until you've run out of ideas, then go to another threat actor.

It may seem like you could just write a program that does something like

```
Foreach (ASSET)
        Foreach (ACTOR)
                Foreach (ACTION)
                        Print "$ACTOR does $ACTION to $ASSET
```

and BOOM you'd have a threat model. While this is true, in reality the number of threats you would come up with is astronomical. This is where common sense comes in to play. The idea of a script kiddie launching a highly sophisticated attack involving a large amount of resources is nonsensical. Similarly, there are numerous attacks a nation state wouldn't carry out because they aren't motivated to, such as defacements and social media attacks. There are several techniques you can use to help target your threat statements.

Use Your Knowledge

Only write down threats that you think are real issues. There's a great deal of knowledge of contemporary attack techniques and motivations. Based on the line of work that your business is engaged in, the specifics of your assets, and the capabilities of various threat actors, use your knowledge of security and attacks to capture threats that make sense. For instance, if you run a large retail operation: we know that in the attack against Target and other institutions, attackers went after the point-of-sale terminals. Therefore, when thinking of threats against point-of-sale systems, a threat like this is appropriate:

- Organized crime group places RAM scraper on point-of-sale terminals in order to steal mag stripe data to facilitate fraud

You'll note that this threat adheres to our threat syntax

- Organized crime group ($actor) uses physical access to place RAM scraper ($action) on point-of-sale terminals ($asset) in order to steal mag stripe data ($outcome) to facilitate fraud ($motivation)

This threat is contemporary and is likely of high concern to retail organizations.

A Threat Is Specific as It Needs to Be

Sometimes a threat does not need all the syntactical parts in order to be useful. Take potential threats against a network router. As we brainstorm what different threat actors would do, we might find that we end up with a list of threats such as these:

- **Nation state** performs denial of service on **router** to stop all access to internal services from Internet
- **Organized crime** performs denial of service on **router** to stop all access to internal services from Internet
- **Insider** performs denial of service on **router** to stop all access to internal services from Internet

Note that multiple threat actors may do the exact same action to your router. While the motivation may be different, the actual attack is the same. Unless you are really worried about the motivation, you can distill these three threats into one:

- Network-based denial of service against router stops all access to internal services from Internet

This threat is still actionable even though a specific threat actor is not represented.

Taking a Break

Brainstorming threats is tough work. It can be difficult to be creative in developing a long list of threats during the early stages. Work your way through all the components in the system representation in the first pass, then take a break. Go do something else, take a walk, drink a beer, or just go home for the day. Whatever you need to do to get away from creating threats, do it. From our experience, taking three or four cycles on adding threats to the list is when we hit the point of diminishing returns. After several sessions of listing threats, you've generally run through all the knowledge you have of a system and likely attack scenarios. Any more sessions will have very little impact on the quality of the resulting model.

Cutting Down the List

After several brainstorming sessions, you should have a list of between 50 and 200 threats, depending on the size of the system under analysis and the types of interfaces it has. A list that big is unusable. To be practical, a threat model should consist of between five and 20 top-line threats. These threats are the top threats the system faces and can be kept in your head as you build and operate the system. Twenty threats can be printed on a single piece of paper and taped on the office wall of every developer and operator in an organization to remind them of what they're defending against.

The first thing to do is to look through the list for threats that are just not realistic. In the process of brainstorming, we should allow ourselves creative license to dream and imagine all manner of bad actions an adversary might take. That process can sometimes lead us to strange places and result in threats that are really just nonsense. Remove these threats from the list. Mechanically, this doesn't mean deleting them. Rather, put them somewhere else. As you threat model more and more systems, having a list of threats to look back on to jar your memory is important. So never delete threats, just move them to a different document or tab.

The next thing to do with the list is to order by variables (actor, action, asset, etc.) to see if there is a way to distill multiple threats into a single threat using the idea that a "threat is as specific as it needs to be." If there are numerous threats that look like basically the same thing, spend time deciding whether they really are different or whether they have the same implication on the enterprise and can be condensed into something similar.

Finally, once you have nonsense threats removed and similar threats condensed, you can start sorting through the threats and ranking them. Ranking does not need to be a formal process; rather, you can use the basic calculus of the risk equation at the beginning of this article to think about what countermeasures you have in place, what the impact of the attack would be, and how likely the threat actor is to carry out the attack. Take your time and play around with the ordering until you think it's correct.

Once the threats are in order, look for a "cut line." Find a place in the list where threats above the list are likely to be important to developers and operators in your organization and below where folks are unlikely to care. There is no right size for the list of threats above the cut line, but generally the list of important threats should fit on a single piece of paper (printing in 6-point font doesn't count). Once you have found your cut line, take those threats, print them out, and tape them to your wall. Congratulations, you have made your first threat model!

Using the Threat Model

Now what? Unlike a risk assessment or vulnerability assessment, threat models tend to change slowly over time. As vulnerabilities are patched, a vulnerability assessment loses its utility. Vulnerability assessments may have a useful life of a few weeks. Risk assessments, which tend to focus on higher-level issues, may have a lifetime of a few months. Threat models can often live on for a year or two without any changes. Threats to an organization change slowly over time as economic and political systems evolve. After a year, you should reexamine your threat model to determine whether it needs updating; if your organization's situation is generally the same as it was, you can let it ride.

The threat model should be presented to developers and operators throughout your organization. Developers can use the model to help them write more defensible code. Operators can use the model to help implement and configure better security controls. The threat model can serve as the foundation of future risk assessments and help penetration testers understand what you are really concerned about. The threat model is a foundational piece of a risk-based approach to cybersecurity.

Threat modeling is still an art form. It is an important part of any cybersecurity program, but performing threat modeling is not a well understood process. The process and ideas put forth here are guidelines; they are not meant to be the hard and fast method you must use to create a threat model. Rather, they are a starting point. As you work through the process a few times, you will find ways to optimize and customize it to have more utility for you and your organization. Use these customizations to create even better and more relevant threat models to help secure your systems. Further, share your improvements with those around you so that we can all learn as we advance the discipline of threat modeling.

Reference

[1] A. Shostack, *Threat Modeling: Designing for Security* (Wiley, 2014).