Carnegie Mellon University
Software Engineering Institute

A Hybrid Threat Modeling Method

Nancy R. Mead
Forrest Shull
Krishnamurthy Vemuru, University of Virginia
Ole Villadsen, Carnegie Mellon University

March 2018

TECHNICAL NOTE
CMU/SEI-2018-TN-002

CERT Division

http://www.sei.cmu.edu

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

This report was prepared for the SEI Administrative Agent AFLCMC/AZS 5 Eglin Street Hanscom AFB, MA 01731-2100

DM18-0125

Table of Contents

List of Figures

List of Tables

Acknowledgements

We appreciate the earlier contributions of our FY2016 research team members: Sam Weber, Tamara Denning, Jane Cleland-Huang, Stefan Hiebl, Tadayoshi Kohno, and Janine Spears. We also acknowledge the technical review and support of Peter Beling at the University of Virginia. At the SEI, Claire Dixon did an outstanding job of editing the report, and David Biber provided many improvements to the graphics.

Abstract

In FY 2016, the research team evaluated Security Cards, STRIDE (Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service, Elevation of privilege), and persona non grata (PnG) for effectiveness in threat identification. Security Cards is an approach that emphasizes creativity and brainstorming over more structured approaches such as checklists. STRIDE involves modeling a system and subsystem and related data flows. PnGs represent archetypal users who behave in unwanted, possibly nefarious ways. The team used two scenarios: an aircraft maintenance scenario and a drone swarm scenario, both described in this technical note in detail, along with the project outcomes. No individual threat modeling method included all identified threats.

The research team subsequently developed the Hybrid Threat Modeling Method (hTMM), considering the desirable characteristics for a Threat Modeling Method. At a high level, the hTMM includes the following steps, described in detail in the technical note: (1) Identify the system you will be threat modeling. (2) Apply Security Cards according to developers' suggestions. (3) Prune PnGs that are unlikely or for which no realistic attack vectors could be identified. (4) Summarize results from the above steps, utilizing tool support. (5) Continue with a formal risk assessment method.

1 Background

Modern software systems are constantly exposed to attacks from adversaries that, if successful, could prevent a system from functioning as intended or could result in exposure of confidential information. Accounts of credit card theft and other types of security breaches concerning a broad range of cyber-physical systems, transportation systems, self-driving cars, and so on, appear almost daily in the news. Building any public-facing system clearly demands a systematic approach for analyzing security needs and documenting mitigating requirements. The challenge of security is that adversaries are intelligently trying to defeat the intent of system stakeholders—not necessarily maliciously. For example, many extremely serious security violations have been caused by hard-working employees who put highly sensitive data on USB drives in order to work from home. While security can be analyzed at the networking and code level to prevent buffer overflows, SQL injection attacks, and so on, there is value in creating a mindset of defensive thinking early in the requirements engineering process. Thinking defensively means that for every new requirement or feature, we need to think about how it could be abused or defeated by adversaries.

An illustrative example of requirements process failure is a rash of Automated Teller Machine (ATM) fraud incidents that occurred in the 1980s, as described by Ross Anderson [Anderson 1994]. ATM designers were aware of the necessity of preventing fraud and implemented security features accordingly. Unfortunately, Anderson argues, they failed to correctly anticipate their adversaries' abilities and means of attack. The system designers assumed that criminals would attempt to break the cryptography underlying the ATM system and so diligently designed the system to withstand such attacks. However, non-cryptographic attacks did not receive such consideration. The result was a system design that provided little defense against some fairly straightforward adversary actions. For example, at one bank, tellers could change a customer's mailing address to their own, request a replacement ATM card and PIN number to be sent to the customer's new address, and then change the mailing address back, all without any activity being recorded in the bank's logs.

In most systems, security requirements, if they exist at all, tend to be rather repetitive, specifying relatively obvious features such as "Only authorized users shall access personal healthcare information." Occasionally requirements are included that stem from regulatory controls, such as "An audit log must be maintained of every access to the patient's healthcare information." As the ATM fraud example shows, merely specifying security features is insufficient—one needs to anticipate ways in which a system can be misused by adversaries. It is critically important to perform systematic, rigorous, and customized threat analyses [Hilburn 2013]. Once the likely threats have been identified, the associated attack methods can be documented, for example, as misuse cases [Opdahl 2009]. With this information in hand, analysts can define and document mitigation strategies aimed at thwarting the attack as use cases that lead to appropriate written security requirements. This type of approach is standard in many existing security requirements engineering processes, such as SQUARE [Mead 2005]. As a consequence, early specification of security requirements positively impacts fundamental architectural decisions that enable security concerns to be addressed from the ground up, rather than added as late-in-the-day patches in an attempt to remediate security vulnerabilities.

Personae non Gratae (PnGs) are a means to focus attention on early-phase threat modeling. Personas have long been used for User Interaction (UX) design as a means of integrating user goals and perspectives into the design process [Cooper 1999]. Personas represent the intended archetypal users of the system. In contrast, PnGs represent archetypal users who interact with the system in unwanted ways and, as a result, undermine the system's security.

1.1 Threat Modeling

There are many definitions of threat modeling. Based on our discussions with practicing threat modelers, we adopt the definition that "a threat modeling method (TMM) is an approach for creating an abstraction of a software system, aimed at identifying attackers' abilities and goals, and using that abstraction to generate and catalog possible threats that the system must mitigate" [Shull 2016a].

There are a number of threat modeling methods. Perhaps the most well-known and widely used TMM is STRIDE, derived from six threat categories: Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service, and Elevation of privilege. It was invented in 1999 by Kohnfelder and Garg [Kohnfelder 1999], implemented at Microsoft, and widely adopted. STRIDE involves modeling a system and subsystem and the related data flows. After that, the methodology relies on a checklist evaluation approach based on the six threat categories listed above to identify specific threats to the system.

Trike was developed in 2005 to improve on perceived deficiencies of STRIDE [Saitta 2005]. It is designed for security auditing from a risk management perspective and models threats from a defensive viewpoint (i.e., in contrast to an attacker's viewpoint).

The Process for Attack Simulation and Threat Analysis (PASTA) was developed around 2012 by Morana and UcedaVelez [UcedaVelez 2012]; a companion book was published in 2015 [Morana 2015]. The PASTA is a seven-stage process that yields the impact of threats to an application and business. The PASTA process includes the steps below:

- defining business objectives and security requirements
- decomposing the application into use cases and Data Flow Diagrams (DFDs)
- introducing threat trees and abuse cases
- defining risk and business impact

Visual, Agile, and Simple Threat Modeling (VAST) was developed by A. Arguwal and is the basis for the first commercially available threat modeling tool, ThreatModeler [ThreatModeler 2017]. It is intended for large/medium organizations while supporting prevailing Agile methodology. It is designed for consistent output regardless of the implementer and it is implemented by DevOps teams during the software development lifecycle. VAST divides threat models into two categories:

- Application models: process flow diagrams are created that focus on a specific application
- Operational models: end-to-end DFDs are created that incorporate application interactions

Security Cards were developed at the University of Washington by Tamara Denning, Batya Friedman, and Tadayoshi Kohno [Denning 2013]. The method relies on physical resources, that is,

cards to assist brainstorming about potential cyber threats. The stated purpose of Security Cards is to facilitate the exploration of potential security threats for a particular system and, more broadly, to help develop a security mindset. The audience envisioned for Security Cards includes educators, students, researchers, and practitioners.

A misuse case is a TMM that extends the popular use case diagram to include security threats and countermeasures [Sindre 2008]. A use case diagram, capturing core system functionality, is transformed into a misuse case diagram by adding potential misactors (i.e., threat agents), undesirable misuse cases (i.e., threats) that could potentially disrupt or otherwise harm the desired system functionality, and use cases that counter identified threats. As such, misuse case diagrams depict the interactions between bad actors, threats, countermeasures, and the original use cases.

PnGs are inspired by the notion of personas in UX design and adopt several ideas inherent to misuse cases [Cleland-Huang 2014].

1.1.1 Personas

A persona provides a realistic and engaging representation of a specific user group. It is typically depicted through a representative image and a personal description that portrays something about the psyche, background, emotions and attitudes, and personal traits of the fictitious person [Putnam 2012, Nielsen 2013]. The task of creating a persona usually involves surveying and interviewing users, identifying optimal ways for slicing users into categories, collecting data to demonstrate that the proposed slices create distinguishable user groups, discovering patterns within the user groups, constructing personas for each group, and then creating scenarios describing how the persona might interact with the system under development. A project will typically have about five to eight personas.

While personas are typically used for purposes of UX design, there are examples in which they have been used as part of the requirements elicitation and design process. For example, within the privacy literature, Spears and Erete defined a persona to elucidate a segment of end users who are unconcerned about online behavioral tracking [Spears 2014]. In another example, Dotan and colleagues evaluated the use of personas to communicate users' goals and preferences to project members as part of a two-day design workshop for the APOSDLE project [Dotan 2009]. Similarly, Robertson and colleagues also discussed the use of personas for gathering requirements when actual stakeholders are not available [Robertson 2006]. In these examples, the focus was on eliciting a general set of end-user goals and system requirements. Cleland-Huang and colleagues introduced the notion of using personas to analyze architectural goals [Cleland-Huang 2013] and PnGs for security threats [Cleland-Huang 2014].

1.1.2 Who Does Threat Modeling?

Threat modeling is performed by a variety of organizations, including vendors such as Microsoft. Microsoft uses STRIDE and makes it freely available. U.S. government organizations such as the DoD are mandated to perform threat modeling. Various methods are in use; some are based on National Institute of Standards and Technology (NIST) standards, such as the Risk Management Framework (RMF), and some use checklists. Commercial organizations such as the automotive industry, finance, and so on also perform threat modeling. Various methods are in use, including

STRIDE, broader risk analysis approaches such as OCTAVE, and more narrowly defined methods such as development of attack trees.

2 Threat Modeling Evaluation Research Project

On our initial threat modeling evaluation project, we experimented with Security Cards, STRIDE, and PnG. Our objective was to determine whether one of these methods was clearly the most efficient and effective in identifying valid threats, while minimizing the number of false positives. We used two scenarios, an aircraft maintenance scenario and a drone swarm scenario, both described in Appendix A. We'll describe each of these in more detail, along with the project outcomes.

2.1 Security Cards

Security Cards is an approach to threat modeling that emphasizes creativity and brainstorming over more structured approaches, such as checklists, to help users identify unusual or more sophisticated attacks. The method is suitable for purposes ranging from fundamental learning about security threats to aiding advanced professionals in system design [Denning 2013]. The use of Security Cards helps to answer the following questions: If your system were compromised, what human assets could be impacted? Who might attack your system and why? What resources might the adversary have? How might the adversary attack your system? Cleland-Huang and colleagues explore how Security Cards could be used to identify threats to a technological system such as implantable cardioverter-defibrillators (ICDs) [Cleland-Huang 2016]. Marasco and colleagues describe how Security Cards can be used to explore cyber threats against biometric identity collection and management systems [Morasco 2017].

In a Security Card deck, the 42 cards are divided into four dimensions: Human Impact (9), Adversary's Motivations (13), Adversary's Resources (11), and Adversary's Methods (9). The Human Impact cards focus on the different ways an attack can affect human lives, such as financial loss and violations of privacy. Adversary's Motivations cards uncover the reasons why someone would desire to carry out an attack on an information system, such as financial gain or revenge. The ways and means for carrying out such an attack, such as tools or expertise, are addressed in the Adversary's Resources cards. Finally, Adversary's Methods cards investigate how an attacker might carry out an attack, such as through technical means or through coercion or manipulation. A card example looks like this:

Figure 1: Security Card Example

The square below contains all the options covered in each of the four dimensions:

Human Impact

- the biosphere
- emotional well-being
- financial well-being
- personal data
- physical well-being
- relationships
- societal well-being
- unusual impacts

Adversary's Motivations

- access or convenience
- curiosity or boredom
- desire or obsession
- diplomacy or warfare
- malice or revenge
- money
- politics
- protection
- religion
- self-promotion
- world view
- unusual motivations

Adversary's Resources

- expertise
- a future world
- impunity
- inside capabilities
- inside knowledge
- money
- power and influence
- time
- tools
- unusual resources

Adversary's Methods

- attack cover-up
- indirect attack
- manipulation or coercion
- multi-phase attack
- physical attack
- processes
- technological attack
- unusual methods

Figure 2: Security Card Dimensions

(*Source: http://securitycards.cs.washington.edu/assets/security-cards-information-sheet.pdf*)

Users can apply Security Cards for threat modeling in many ways, according to the Security Card designers [Denning 2013]. For example, threat modelers could begin with a system or application

they wish to protect and investigate each dimension individually to determine the extent to which each card might affect the system. Alternatively, they could also explore and assemble combinations of cards from different dimensions to uncover the most relevant and critical cyber threats.

In our previous study of three threat modeling techniques [Shull 2016b], we found that teams of participants using Security Cards demonstrated high effectiveness. The teams uncovered nearly all types of threats against the systems, but the threats identified using Security Cards varied considerably across teams and included many false positives. These results suggest that Security Cards would be useful where circumstances value wide-ranging perspectives over consistent results.

An example of a threat model developed with Security Cards can be found in Appendix C.

2.2 Persona Non Grata (PnG)

PnGs represent archetypal users who behave in unwanted, possibly nefarious ways. However, like ordinary personas, PnGs have specific goals that they wish to achieve and specific actions that they may take to achieve their goals. Modeling PnGs can therefore help us to think about the ways in which a system might be vulnerable to abuse and use this information to specify appropriate mitigating requirements.

The PnG approach makes threat modeling more tractable by asking users to focus on attackers, their motivations, and abilities. Once this step is completed, users are asked to brainstorm ideas about targets and likely attack mechanisms that the attackers would deploy. The theory behind this approach is that if engineers can understand what capabilities an attacker may have and what types of mechanisms they may use to compromise a system, the engineers will gain a better understanding of targets or weaknesses within their own systems and the degree to which they can be compromised. Some critics of this approach argue that a PnG can often take users down the wrong path. For example, for a system related to national security, users might reason that the system may be the target of a sophisticated attack from another nation-state. This conclusion, however, overlooks the fact that a nation-state might compromise a system first through a much simpler entry point and then ratchet up operations from there.

We provide examples of two PnGs in Figure 3. These PnGs were constructed manually for training purposes and target the domain of Electro-Cardio Converters. Each PnG includes an image of the persona, his or her name, a description, the assumed role (e.g., Mechanical Engineer), and a moniker (e.g., Bitter and revengeful). Furthermore, it includes a set of relevant goals and skills, and a set of misuse cases that describe specific ways in which the PnG intends to attack the system [Opdahl 2009]. From this, we can construct a threat model that includes the actor (i.e., the PnG) and the attack mechanism and target specified in the misuse cases. Attack intent is provided in the general description.

A different set of examples is provided in Appendix B.

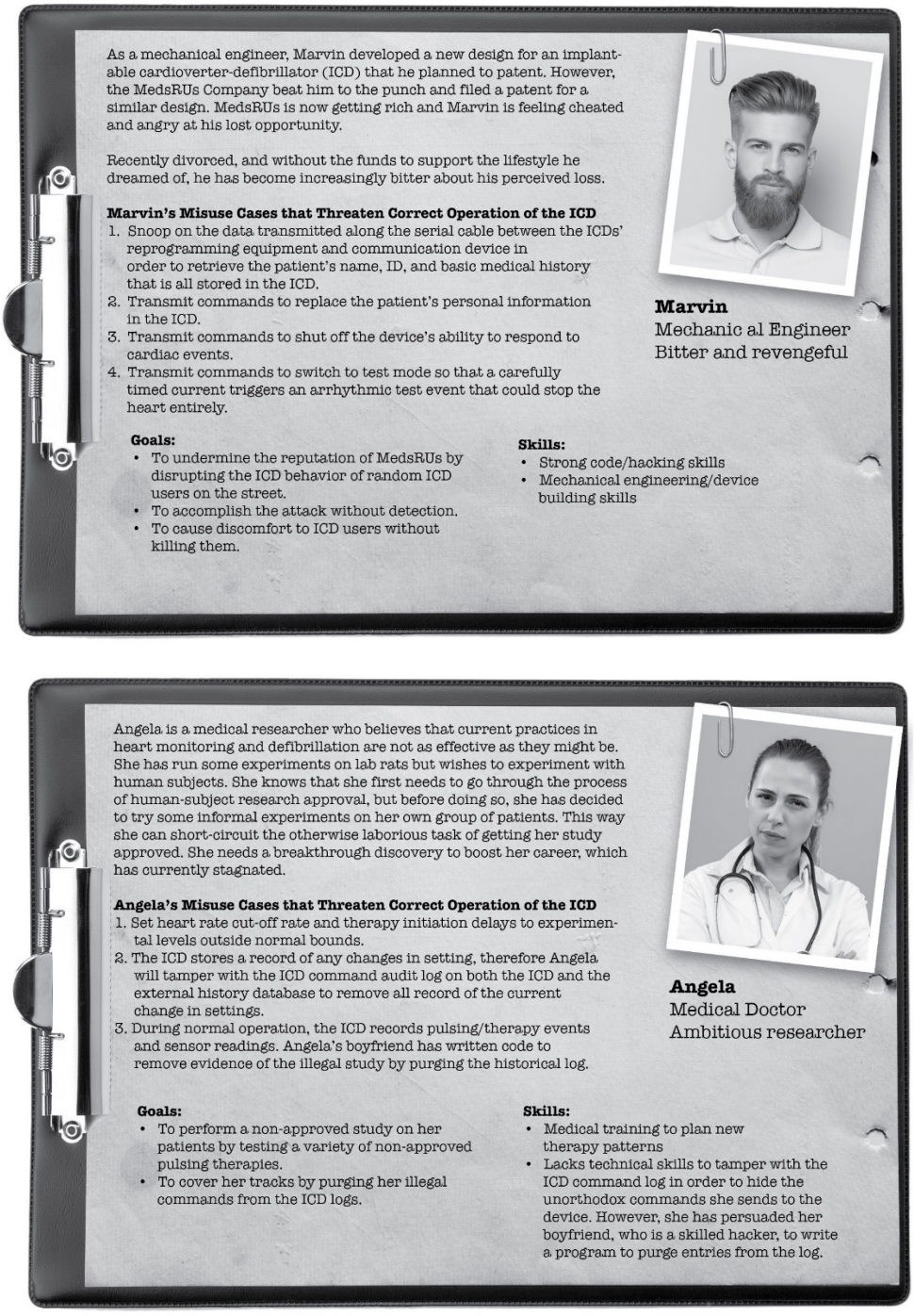

Figure 3: Two Personae Non Gratae

We asked students in two introductory information security courses, one undergraduate and the other graduate, to work in teams of three to four people to construct PnGs for one of two systems [Shull 2016b]. Here we focus on one of these systems, which utilized unmanned aerial vehicles (UAVs) to perform a rescue mission. The drones could carry payloads such as emergency supplies and were capable of autonomous operation if communication with the base station was lost. Each scenario was described in a two-page document that represented very early ideas for each of

the projects. Goals, major constraints, and high-level designs were provided, but not the lower-level implementation decisions.

We used crowdsourcing in conjunction with PnG to enhance the results. Mead describes the crowdsourcing part of the study [Mead 2017].

2.3 STRIDE

STRIDE is the most well-established TMM and represents the state of the practice. At its core, STRIDE requires breaking down a system into its various elements, assessing each of these elements for their vulnerability to threats, and then mitigating these threats [Hernan 2006]. In practice, a typical STRIDE implementation includes modeling a system with DFDs, mapping the DFD elements to the six threat categories, determining the specific threats via checklists or threat trees, and documenting the threats and steps for their prevention [Scandariato 2015]. STRIDE can be implemented manually, through the free Microsoft Secure Development Lifecycle (SDL) Threat Modeling Tool [Microsoft 2017], or through other third-party tools or implementations; for example, a lightweight variant of STRIDE was adopted by the Ford Motor Company [Ingalsbe 2008].

The table below identifies the security property associated with each of the six threat categories.

Table 1: Threat Categories and Security Properties

Threat	Security Property
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-repudiation
Information Disclosure	Confidentiality
Denial of Service	Availability
Elevation of Privilege	Authorization

Data Flow Diagrams (DFDs) are designed to show how a system works by using standard symbols to graphically represent the interaction between data stores (e.g., databases, files, registries), processes (e.g., DLLs, web services), data flows (e.g., function calls, remote procedure calls), and external entities (e.g., people, other systems) [Shostak 2014]. Once complete, each of these system elements in turn can be associated with one or more relevant threat categories, as depicted below:

Table 2: Threat Categories of DFD System Elements

Element	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Data Flows		X		X	X	
Data Stores		X		X	X	
Processes	X	X	X	X	X	X
External Entity	X		X			

In the next stage, the typical STRIDE user works through a checklist, which may be in the form of a threat tree, of specific threats that are associated with each match between a DFD element and threat category. Such checklists are accessible through STRIDE reference books or tools.

In their descriptive study of STRIDE, Scandariato and colleagues found that subjects judged it to be relatively easy to implement but time consuming [Scandariato 2015]. Their study also reported a relatively low rate of incorrect threats (false positives) and relatively high rate of overlooked threats (false negatives).

In our prior study of three threat modeling techniques [Shull 2016b], we found that subjects using the STRIDE method did not report a lot of false positives but did not reach consistent results across teams. The threats reported appeared to correlate with the makeup of specific teams and their background or experience. Based on these conclusions, STRIDE may be more appropriate for teams that lack significant security expertise because the checklist-based approach constrains users and limits the potential for false positives. However, using STRIDE for threat modeling requires an onerous and time-consuming application of checklists of potential threats to the components of the various systems and subsystems. Finally, the success of the STRIDE method depends heavily on creating accurate DFDs.

2.4 Research Project Outcomes

In our study, we compared the use of PnGs, STRIDE, and Security Cards, and found that threat models built using PnGs exhibited a higher degree of consistency than other techniques [Shull 2016b]. Nonetheless, no individual threat model included all identified threats. We therefore explored the idea of crowdsourcing the task of threat identification. Our approach used information retrieval techniques to analyze the identified threats and collate them and provided the results to a human analyst to assist with construction of a unified threat model.

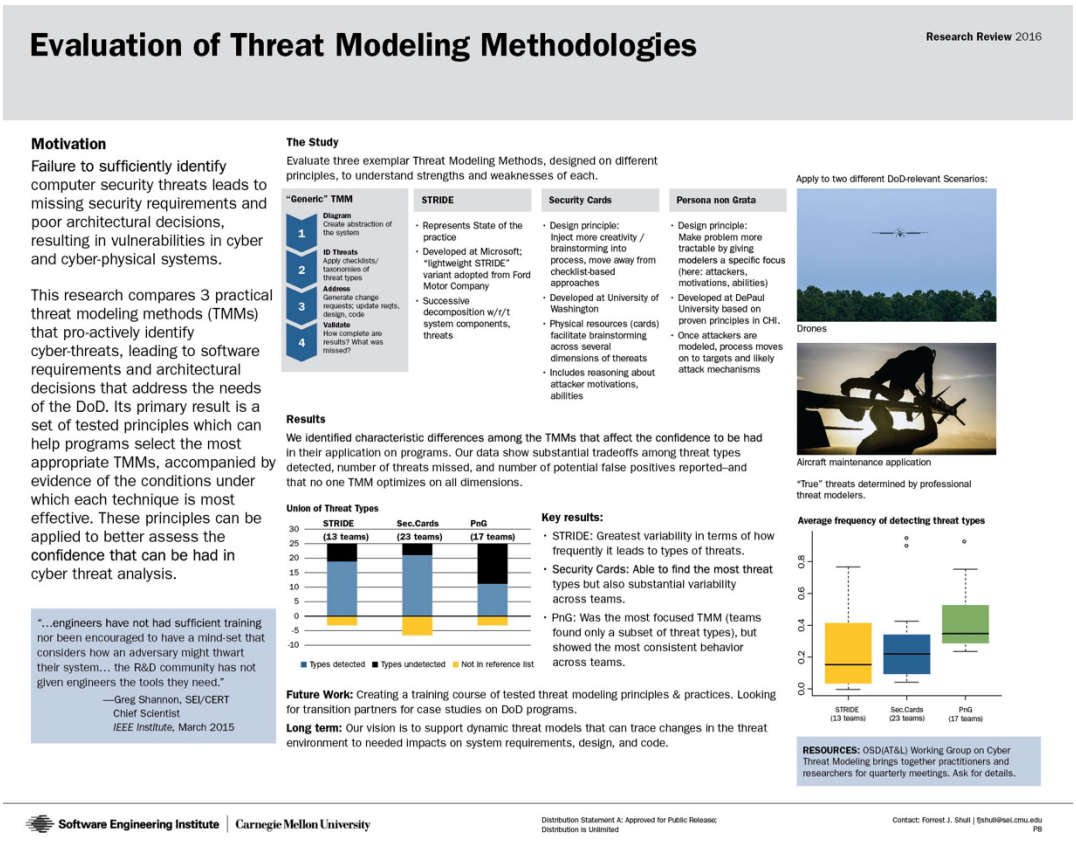

Figure 4: Evaluation of Threat Modeling Methodologies

3 Hybrid Threat Modeling Method

In developing the Hybrid Threat Modeling Method (hTMM), we considered the following desirable characteristics for a Threat Modeling Method:

3.1 Desirable Characteristics for a Threat Modeling Method

Below are the desirable characteristics:

- no false positives
- no overlooked threats
- consistent results regardless of who is doing the threat modeling
- cost-effective (doesn't waste time)
- empirical evidence to support its efficacy

3.2 Other Considerations

Here are additional characteristics to consider:

- has tool support
- suggests a prioritization scheme
- easy to learn, intuitive
- encourages thinking outside the box
- can be used by non-experts, or conversely, optimal for experts
- clearly superior for specific types of systems

3.3 Steps for the Hybrid Threat Modeling Method

The steps in the hTMM are as follows:

1. Identify the system you will be threat modeling. Execute Steps 1-3 of SQUARE or a similar security requirements method.
    a. Agree on definitions.
    b. Identify a business goal for the system, assets and security goals.
    c. Gather as many artifacts as feasible.
2. Apply Security Cards in the following way, as suggested by the developers (http://securitycards.cs.washington.edu/).
    a. *Distribute the Security Cards to participants either in advance or at the start of the activity.* Include representatives of at least the three following groups of stakeholders: system users/purchasers, system engineers/developers, and cybersecurity experts. You may find that within each of those categories, there are multiple distinct perspectives that must be represented. Other relevant stakeholders can be included as well.
        i. System users/purchasers would include those purchasing or acquiring the system, end users, and other groups with a vested interest in the system. For example, in a

scientific research organization, stakeholders could include the scientists conducting research, the executive directors of the organization, human resources, and information technologists managing the system. Each role would have its own ideas about assets that must be protected and potential attackers.

ii. Cybersecurity experts could be part of a separate specialized team or integrated into the project team. They could include roles such as system administrators, penetration testers or ethical hackers, threat modelers, security analysts, and so on.

iii. The engineer/development team members could range from systems engineers, requirements analysts, architects, developers, testers, and so on.

b. Have the participants look over the cards along all four dimensions: Human Impact, Adversary's Motivations, Adversary's Resources, and Adversary's Methods. Read at least one card from each dimension, front and back.

c. Use the cards to support a brainstorming session. Consider each dimension independently and sort the cards within that dimension in order of how relevant and risky it is for the system overall. Discuss as a team what orderings are identified. It's important to be inclusive, so do not exclude ideas that seem unlikely or illogical at this point in time. As you conduct your brainstorming exercise, record the following:

i. If your system were compromised, what assets, both human and system, could be impacted?

ii. Who are the Personae non Gratae (https://www.infoq.com/articles/personae-non-gratae) who might reasonably attack your system and why? What are their names/job titles/roles? Describe them in some detail.

1. What are their goals?
2. What resources and skills might the PnG have?

iii. In what ways could the system be attacked?

1. For each attack vector, have you identified a PnG (or could you add a PnG) capable of utilizing that vector?

3. Once this data has been collected, you will have enough information to prune those PnGs that are unlikely or for which no realistic attack vectors could be identified. Once this is done, you are in a position to take this step:

a. Itemize their misuse cases. This expands on HOW the adversary attacks the system. The misuse cases provide the supporting detailed information on how the attack takes place.

4. Summarize the results from the above steps, utilizing tool support, as follows (https://resources.sei.cmu.edu/asset_files/Presentation/2016_017_001_474200.pdf):

a. Actor (PnG): Who or what instigates the attack?

b. Purpose: What is the actor's goal or intent?

c. Target: What asset is the target?

d. Action: What action does the actor perform or attempt to perform? Here you should consider both the resources and the skills of the actor. You will also be describing HOW the actor might attack your system and its expansion into misuse cases.

e. Result of the action: What happens as a result of the action? What assets are compromised? What goal has the actor achieved?

f. Impact: What is the severity of the result (high, medium, or low)

g. Threat type: (e.g., denial of service, spoofing)

5. Once this is done, you can continue with a formal risk assessment method, using these results, and the additional steps of a security requirements method such as SQUARE, perhaps tailoring the method to eliminate steps you have already accounted for in the threat modeling exercise.

4 Rationale and Discussion

Steps 1 and 5 are activities that precede and follow the bulk of the threat modeling work. We felt it was necessary to include these, to understand where hTMM fits into lifecycle activities, specifically security requirements engineering.

Our initial thought was that we could apply Security Cards and then PnG to come up with a hybrid method. It became clear to us that we could not just do Security Cards followed by PnG, as each one is a threat modeling method in its own right. Therefore, we needed to consider specific aspects of both Security Cards and PnG and also consider what we had learned by our experiments with STRIDE. We wanted to capture the best features of all the models. Having said that, it's possible that the hTMM could change after we have done a few medium-to-large pilot threat modeling projects, in addition to the small case studies in this report, not to mention those conducted by students on our earlier research project.

Ideally, when considering stakeholders, we would like to be able to recommend a core set of important perspectives (a team might add others that are important in context but should not do this without all of the core being represented). Possibly we could do this by suggesting a particular focus that we think each perspective brings. It's worth noting that some pilot studies will probably determine whether this is feasible or whether the needed perspectives vary too widely depending on the system being studied to recommend a core set of perspectives.

We visualized the hybrid approach as using Security Cards to cast the net wide and generate lots of ideas, and then PnG to filter and weed out the poor ones. We do have a concern about possible overlap between Steps 2 and 3. After a few pilots, we may conclude that we are doing the same thing twice, or we may decide that the high-level and refined views of the threats are both necessary. Looking at it another way, using PnG in conjunction with attack vectors used by a PnG could serve as a confirmation and sanity check for the work done with Security Cards, and vice versa.

For Step 4, we felt that tool support would help to organize the results and eliminate the need for what otherwise appeared to be a copy-and-paste exercise. Ultimately, it's possible that more sophisticated tools could help with the analysis, but at a minimum tools should be used to eliminate mundane transcription and bookkeeping tasks.

4.1 Measurement Considerations

1. From a research point of view, we would like to collect data on the number and types of issues that come from each stakeholder type so we could have some evidence about what each contributes to the overall threat model. However, we are not sure if we can collect at that level of granularity.
2. Building on the comments above, it would be good to know how many items get generated in Step 2, and then how many are dropped vs. refined in Step 3. With some work we could also map those to an Oracle dataset so that we could see if the ones that got filtered were actually related to real threats or if the ones that get refined in PnG were false positives that weren't worth the effort.

5 Exemplar Scenario Application and Results

We applied the hTMM to the drone scenario described in Appendix C. We itemize the results of applying the method in the following sections, along with recommendations for improvement. The numbering below refers to the steps in the hTMM method described in Section 3.

1. System Info Gathering

The system to be studied is a drone or drone swarm that is on its way to deliver emergency supplies to flood-affected populations. It is dispatched by a team consisting of local government authorities and its drone technology contractors. See Figure 5 for an example of a drone swarm.

The drones face several potential threats, both physical and cyber in nature. We will consider potentially likely scenarios of drone attack and how those attacks affect both the people who depend on the drones and the drones themselves.

Figure 5: Example of a Drone Swarm
(*Source: http://www.ioti.com/security/drones-are-coming-take-cover*)

2. Brainstorming

2a) Involve Representative Stakeholders

The following is a ranking of various stakeholders based on the value they would add to a brainstorming session.

Table 3: Stakeholders Ranked by Value to Brainstorming Session

Stakeholder	The Expertise	Value Added
Drone Designer	Knows the safe operating range and the breakdown range its dynamics, the design specifications of each component and the drone.	High, because the information can be used to analyze both cyber threats that try to drive the drone out of its operational range and physical threats where the components fail to propel the drone in the desired path.
Drone Pilot	Knows the optimal parameter settings for safe navigation, and can program the drone to go from point A to point B without violating the drone flying regulations, if any.	High, because the knowledge of the potential routes the drone may take can be used by attackers to capture or damage the drone. This should help the analysts to learn about the locations where the attackers can intercept the drone.
Telecommunications Expert	Given a drone make and communication electronics used, this person can give information on the type of signals and their adversaries during a signal-based attack such as electromagnetic jamming.	High, because jamming of electronics signals is a known and viable attack pathway.
Local government official	Knows about the drone use cases for relief and other public service missions.	Low, because we can learn about the load carried by drones and dates of use cases, but not any technical information about the drone itself.
Threat modeler	Knowledgeable to identify new attack use cases or evaluate the attack model.	High, because this person can refine threat modeling and may contribute with alternatives.

2b) Review the Following Threat Model Dimensions

These are the dimensions reviewed:

- human impact
- threat to drone body and its electronics (GPS and Telecom)
- threat motivation
- attack channels
- adversary's resources
- adversary's methods
- design risk as a whole
- impact
- financial loss and the cost of potential investigations

Part 1: Perform ranking within each category with reason it is considered a potential threat

Human Impact Cards:

1. Emotional well-being (those suffering from the disaster are deprived of basic commodities and get depressed—the primary subject of the threat)
2. Physical well-being (health is affected due to lack of timely food supplies and medicine—the primary subject of the threat)
3. Relationships (the relations between the people, local authorities, and government is at stake if the rescue mission fails—a secondary subject)
4. Unusual impacts (loss of property, loss of life, loss of trust in local government, loss of businesses—a secondary or ternary subject)

Adversary's Motivations:

1. Money (the goods stolen from the drones and the drones/components can be resold to make money—the profitable nature makes this rank 1!)
2. Warfare (some local trouble makers may see this as a route and non-violent means of attack—allowing ease of attack, i.e., not having to face other humans in the operations makes this rank above the rest)
3. Politics (oppositions and opposition groups may intrude to bring bad name to local government—can lead to change of power, so it becomes attractive)
4. Unusual motivations (hack the drones and use them for other unauthorized purposes such as flying in restricted zones or delivery of harmful goods or simply to destroy)

Adversary's Resources:

1. Expertise (the attacker has all the expertise to hack the brand of drones used in the mission)
2. Inside knowledge (access to inside knowledge makes the attack viable)
3. Money (money flowing in for political reasons to bring down local government's reputation)
4. Inside capabilities (an insider who turns attacker can do a lot of damage to the drones)

Adversary's Methods:

1. Physical attack (shoot the drone with a drone gun)
2. Technological attack (jam the GPS or the rotors)
3. Multiphase attack (damage partially—e.g., damage one rotor and partially disable the drone to take control)
4. Manipulation or coercion (hack the drone information system and its GPS, then change the destination or send it back to the origin or make it lose the sense of direction)

Part 2: In-depth analysis of all the potential threats

Human Impact Cards:

1. Emotional well-being (those suffering from the disaster are deprived of basic commodities and get depressed—the primary subject of the threat)
2. Physical well-being (health is affected due to lack of timely food supplies and medicine—the primary subject of the threat)
3. Relationships (the relations between the people, local authorities, and government is at stake if the rescue mission fails—a secondary subject)
4. Unusual impacts (loss of life, loss of trust in local government, loss of businesses—a secondary or ternary subject)

Type	Actor	Action	Target	Purpose	Result	Impact
Denial	Attacker	Attack Methods 1-4	Drone (Physical, Cyber)	Motivations 1-4	Human Impacts 1-4	1-High 2-High 3-Low 4-Low

Adversary's Motivations:

1. Money (the goods stolen from the drones and the drones/components can be resold to make money—the profitable nature makes this rank 1!)
2. Warfare (some local trouble makers may see this as a route and non-violent means of attack—allowing ease of attack, i.e., not having to face other humans in the operations makes this an above the rest)
3. Politics (oppositions and opposition groups may intrude to bring bad name to local government—can lead to change of power, so it becomes attractive)
4. Unusual motivations (hack the drones and use them for other unauthorized purposes such as flying in restricted zones or delivery of harmful goods or simply to destroy)

Type	Actor	Action	Target	Purpose	Result	Impact
1-Capture 2,3,4-Hack	Attacker	Intrusion	Drone	Misuse Drone	Adversary's Motivations 1-4	1-High 2,3,4-Low

Adversary's Resources:

1. Expertise (the attacker has all the expertise to hack the brand of drones used in the mission)
2. Inside knowledge (access to inside knowledge makes the attack viable)
3. Money (money flowing in for political reasons to bring down local government's reputation)
4. Inside capabilities (an insider who turns attacker can do a lot of damage to the drones)

Type	Actor	Action	Target	Purpose	Result	Impact
Denial, Spoofing, Jamming, Screening	Tech Expert Hacker	Cyber attacks	Drone-physical, GPS, Accelerometer, Camera	Adversary's Motivations 1-4	Human Impacts 1-4	1,2,3,4 High
Denial, Spoofing, Jamming, Screening	Specific Attacker: a Nation	Physical, Cyber Attacks	Drone-physical, GPS, Accelerometer, Camera	Adversary's Motivations 1-4	Human Impacts 1-4	1,2,3,4 High
Screening, Shooting	Specific Attacker: a Gangster	Physical attacks	Drone, Any of its components	Adversary's Motivations 1-3	Human Impacts 1-4	1,2,3,4 High

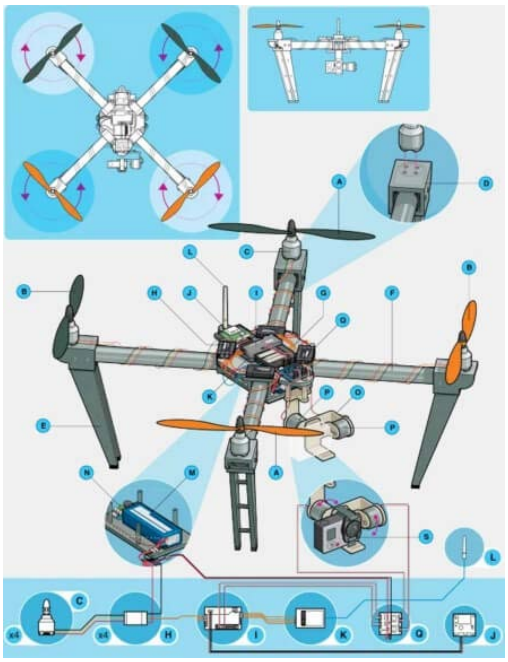

Figure 6: Example of Drone Components
(Source: https://www.dronezon.com)

Adversary's Methods:

To analyze how a drone can be subjected to an attack, let us consider an example. Figure 6 shows typical drone parts: Propellers, Brushless Motors, Motor Mount, Landing Gear, Boom, Drone Body Part, Electronic Speed Controllers, Flight Controller, GPS Module, Receiver, Antenna, Battery, Battery Monitor, Gimbal, Gimbal Motor, Gimbal Control Unit, Camera, Sensors, and Collision Avoidance Sensors. The attack can in principle be on any of the components. We consider a few most likely cases.

1. Physical attack (shoot the drone with a drone gun, direct objects, or spray a dark paint on drone's camera to blind the drone)
2. Technological attack (jam the GPS or the propellers)
3. Multiphase attack (damage partially, e.g., damage one propeller and partially disable the drone to take control)
4. Manipulation or coercion (hack the drone information system and its GPS, then change the destination or send it back to the origin or make it lose the sense of direction)

Type	Actor	Action	Target	Purpose	Result	Impact
Damaging, Capturing, Redirecting	Attacker	Shooting, Hacking, Modifying, Parameters	Drone, Physical, GPS, Accelerometer, Computer	Adversary's Motivations 1-4	Human Impacts 1-4	1,2,3,4 High

2c) Brainstorm with Attention to Mal Actors

Mal Actors (Personae non Gratae):

The insiders:

- drone physical experts, drone software experts, drone operational support staff

The outsiders:

- individuals who may think that the drone is harmful to them if it is flying nearby
- expert drone hackers with some of the motivations listed above

3. Prune Unlikely/Incomplete PnGs

Summarize the critical PnGs.

PnG	Type of Threat	Threat Risk
Drone Hacker	Cyber attack or jamming tele-communication	High, the drone may mal-function and fail the mission
Drone Pilot	Reroute the drone	High, due to loss of drone or loss of control of navigation
Drone Pirate	Capture the drone when it is closer to ground for deliveries	High, because the drone will have to make a landing to de-liver goods in remote areas during relief operations

Drone Reverse Engineer	If the drone goes through a supply chain attacker during manufacturing or when it is captured, this person can modify the components or their functions	Low, this requires supply chain attack and reengineering expertise, which is not a common skill during capture. But this can be High if it happens during manufacturing of the drone.

4. Key Threats

- (1) Cyber attack on drone autonomous navigation programming
- (2) Jamming of drone electronics such as GPS and telecommunications

4a) Actor: Drone Hacker

4b) Purpose: To disable the drone or its mission

4c) Target: The drone

4d) Attack Method: Technological Attack, Multiphase Attack

4e) Results: Damaged relationships; public-government, drone manufacturer-local officials

5. Final Assessment

Type of Impact	Financial Loss (Rough Estimates)	Social Disorder
Failed mission of the drone	If the mission fails, then an alternate attempt for deliveries will be required. This may require exploring alternate resources such as helicopters and result in more unplanned expenditures. The loss depends on the number of days of the mission. Upper limit ~$1 Million	Human Impacts: Emotional well-being Physical well-being Relationships Unusual impacts

Loss of drone navigation control	If a few drones are lost, then replacement drones needs to be dispatched or different make of drones needs to be dispatched to fulfill the mission. This may cost a few thousand dollars. Upper limit ~ $20,000	Human Impacts (as listed above) If drones fly or land in areas other than intended locations, the residents there may be distracted or affected leading to social disturbance.
Loss of drone or its supplies	The cost of the entire fleet of drones. It depends on the make and the model of drone. Upper limit ~ $200,000	Human Impacts
Drone actions do not correspond to initially programmed commands	Need to hire more manpower to locate and retrieve the drone, and allocate hours of programming to realign the commands. This leads to additional spending. Upper limit ~$50,000	Human Impacts Disorder in the drone technology department, work overload on employees and potential delays in future missions.
Replaced or reengineered drone components	If the drone is experiencing a supply chain attack, then it requires physical component diagnostics and replacing with true components. Upper limit ~$100,000	Human Impacts Disorder in the drone technology department at the management level due to the poor choice of drone manufacturers.

Final Thoughts for Further Model and Tool Development

The following are the lessons learned from the perspective of modelers. These can be considered as a passive component of the model and may help a user to expand the scope of modeling as needed during applications.

1. Specific information on the type of drone, a geographic location of the mission, and the budget available for the missions would serve as further helpful parameters to fine tune the model for specific use cases.
2. For a more efficient and routine application of the model, a Windows-based tool that can be used to build the model, for example using toolbar options such as “Select from a list of Icons” or “Dropdown lists” to add a piece of the model, would be very helpful.

3. Application of the model to other device examples should be highly encouraged for cross-culturing of ideas. Such efforts can lead to hTMM model refinement and advancement to a higher level of complexity.

6 Summary

In this report we have briefly mentioned a number of threat modeling methods in the literature. This is not an exhaustive list. We then described three threat modeling methods used in our earlier research project: Security Cards, Persona non Grata, and STRIDE. The purpose of the earlier project was to evaluate which of these methods might be considered "best of breed."

Since there was no clear "winner" in the earlier study, we developed a hybrid method that attempted to meld the desirable features of all three methods. As part of that activity, we developed some threat profile examples using our standard scenarios. We then applied the hybrid method, or hTMM, to one of our standard scenarios. A number of questions were raised, suggesting that some larger pilots will be needed to fully validate the new method.

Our plans are to engage in larger pilot studies to validate or revise the method. Ultimately, we hope to collect data to support the efficacy of the hybrid method.

Appendix A Research Project Scenarios

Aircraft Service Application Scenario

Introduction

Maintenance of aircraft is extremely important. If something goes wrong in an aircraft during flight—an engine catches fire, for example—the results are often catastrophic. As a result, there are many procedures and regulations concerning how aircraft must be maintained.

In this scenario, our organization intends to develop an application for Apple iPads™ [1] to help create and maintain aircraft maintenance records. Our "Service Application" will replace older systems that were both expensive and error prone.

Background

The maintenance records for aircraft are much more involved than those for consumer products familiar to most people. First, every aircraft is different: even airplanes that have the same model number may have been built with parts from different suppliers. For example, if two airplanes with the same model number were built at different times, the manufacturer might have changed suppliers for some parts. Also, every aircraft has highly regulated intervals for service: when only one of the propellers on a two-propeller airplane is replaced, the amount of time each propeller has spent in flight must be tracked so that inspections and replacements can be done at the proper times.

Therefore, the records for a plane include model and serial numbers of the airframe, each engine, each propeller, each rotor, and so forth. The records include the date when each part was installed, inspected, or serviced, and the name of the person who certified that the work was done properly.

When servicing an aircraft, the people doing the work are given checklists of the tasks that must be done. The person overseeing the work must have the proper certification for doing that particular task and must sign off at the end stating that the work was done correctly.

Also, many organizations consider their aircraft records to be confidential. These records are available to regulatory agencies, but companies do not want their competitors or the general public to know the details of their maintenance policies and procedures or details about their individual aircraft.

Periodically, updates to checklists or maintenance policies are made. This usually happens in response to an accident—to help determine an appropriate way to prevent a similar accident from happening again.

Typical Use Case

A typical use case for our application is as follows:

[1] iPad is a trademark of Apple, Incorporated.

A maintenance crew checks out one or more iPads at the beginning of its shift.

The crew consults the Service Application, which displays a list of tasks that must be completed. Crew members are not expected to complete all of the work tasks during their shift—the time taken to do maintenance can vary, and so they do as much as they can during their shift, after which the crew on the next shift takes over.

During a service task, the Service Application shows the crew the maintenance records for that plane, the instruction manuals for doing that particular task, and the correct checklists.

Upon completing an item on the checklist, a crew member must take some action (such as pressing the correct button on the screen) to confirm completion of that item. Sometimes checklist items require information such as the serial number of a part that replaces an old part.

Crew members can take photos of their work using the iPad camera and attach them to the service records.

When a task is complete, the Service Application requires the person responsible for the task (who must be properly certified) to sign off to verify that the task was done correctly.

Sometimes completing a task results in the creation of a new task. For example, an inspection may determine that a part must be replaced. The Service Application supports this scenario.

When their shift is over, the maintenance crew members return the iPads that they used.

The organization's main database of aircraft records is updated as service is completed.

System Description

Some key architecture decisions have already been made, as described below.

Our organization's data center will host

- the database containing all the service records for our aircraft (This database already exists: the Service Application will have to accommodate records that are created by legacy systems. Also, we won't be able to simultaneously replace our existing maintenance system, so even when the Service Application is being used, some people will still be using the older systems.)
- the master database of all the service manuals and checklists for all the equipment used by the organization
- the database of the certifications held by our service people

The iPads themselves will be Wi-Fi enabled, and Wi-Fi will be available in the organization's offices. However, Wi-Fi availability is not guaranteed out on an airport field or inside an aircraft's engine. Our Service Application will be installed on the iPads.

Because the iPad does not have enough storage capacity to hold all of the service manuals and checklists in the organization, the Service Application will provide only the manuals and checklists that are needed. Similarly, an iPad cannot hold all the service records for all the organization's aircraft, so the Service Application will provide only the pertinent records.

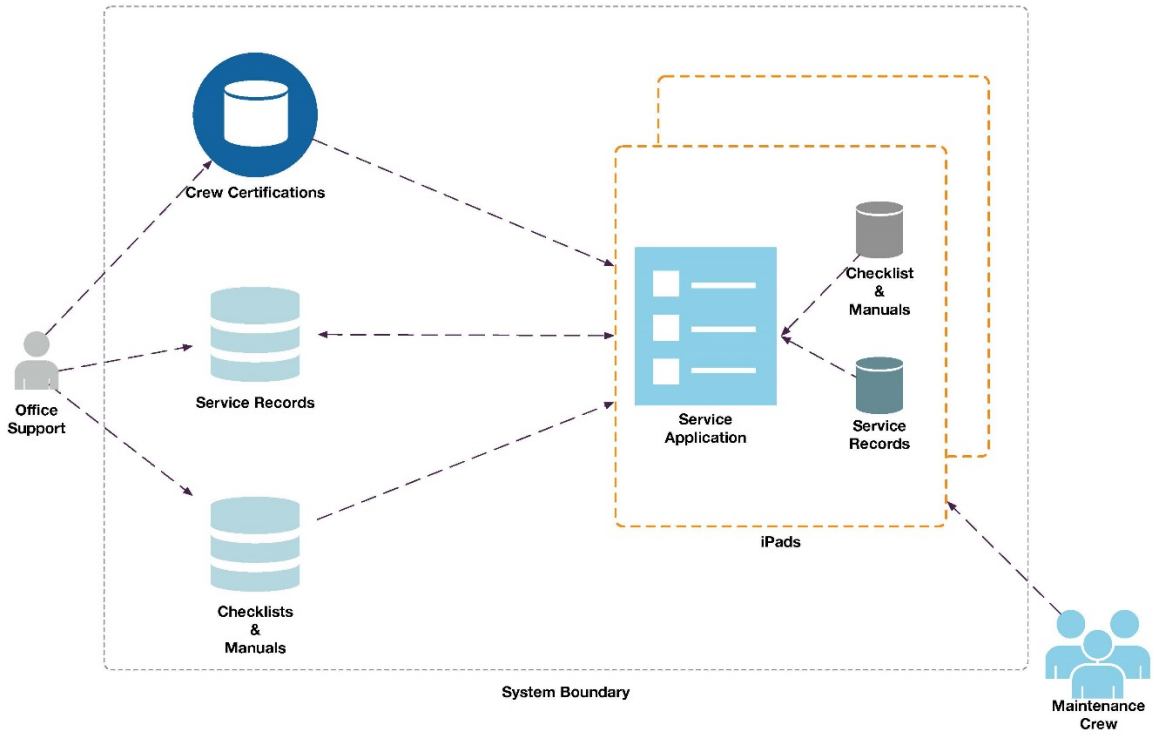

Figure 7: Overview of Service Application

Drone Swarm Scenario

Introduction

Unmanned aerial vehicles—commonly known as drones—are ideal for many rescue and emergency situations: they can fly into dangerous or uncertain conditions and go where manned vehicles cannot. We wish to design and develop fleets (or swarms, as we call them) of drones to be used in situations such as

- surveying and monitoring the extent of forest fires
- surveying the extent of earthquake damage and locating survivors
- delivering medical supplies and equipment to survivors or isolated people

Having human beings manually controlling individual drones not only requires a large number of trained personnel, but often is not even technically possible because of erratic radio communications around mountains and other terrain. As a result, each swarm must be able to act autonomously to achieve its objectives. Swarms should be capable of both national and international use.

High-Level Requirements

Figure 8 shows a deployment example of two swarms, both sent out on search and rescue missions beyond a large fire. Each swarm consists of the following:

- **One "leader"**: this drone contains radio equipment that attempts to maintain communication with a base station. The leader is the only drone that has this equipment; the other drones can communicate with each other and the leader but are not able to reliably reach the base. As a result, if the leader fails, the entire mission fails. Leaders aren't generally customized for the particular mission.
- **One or more "followers"**: these drones usually have customized equipment for the mission—video cameras, medical-equipment payload carriers, and so on. They are in radio communication with the leader, but not the base. Depending on the mission, one or more followers may be required to successfully complete an assigned task, but in general, missions can succeed even when one or more followers fail. Followers don't have the equipment necessary to be "promoted" to leader mid-mission.

Upon deployment, the leader gives the swarm a list of physical coordinates (checkpoints), received from the base. Each coordinate must be reached by a given time. The leader alerts the base when each checkpoint is reached. The leader tracks time via an onboard clock; if any checkpoint is not reached in time, the mission is aborted and the drones return to the base. This list of checkpoints may be changed mid-mission by the base. The leader alerts the base if the swarm is running low on fuel and may not be able to achieve its mission as a result. Naturally, reaching checkpoints on time isn't the only criterion for mission success: performing the survey, dropping medical supplies, and so on, are the ultimate success criteria. The mission checkpoints merely ensure that if flying conditions are much worse than anticipated, the drones won't vainly struggle to get to a location too late to be useful or become unable to return.

The drones should not collide with each other or with the ground. To avoid collisions, all drones are equipped with altimeters (to determine their height) and GPS; all of this information is periodically communicated to other drones in the swarm as well as to the base. If contact with the base

is lost, the leader attempts to re-establish communication while the swarm continues to perform based on the most recent information.

Environmental Constraints

Because swarms may fly in areas containing smoke and debris, drones are expected to operate even when they are in imperfect physical condition or become damaged during the mission.

The follower drones should fly in formations that protect the leader from bird impacts and debris as much as possible. Also, poachers and frightened observers have been known to try to shoot drones. The swarm is given a map of "dangerous flying" areas. Because flying conditions are expected to be poor in these areas, the swarm should fly slower and the followers should stick closer to the leader to protect it from harm. The flying formation and logic for preventing collisions is managed by an algorithm running onboard the drones themselves; the base does not determine the formation.

Political borders and no-fly zones (such as around buildings in Washington, DC) must be observed, and the swarm must not fly into these areas.

The drones themselves, and their low-level software, will be built and supplied by a third-party company.

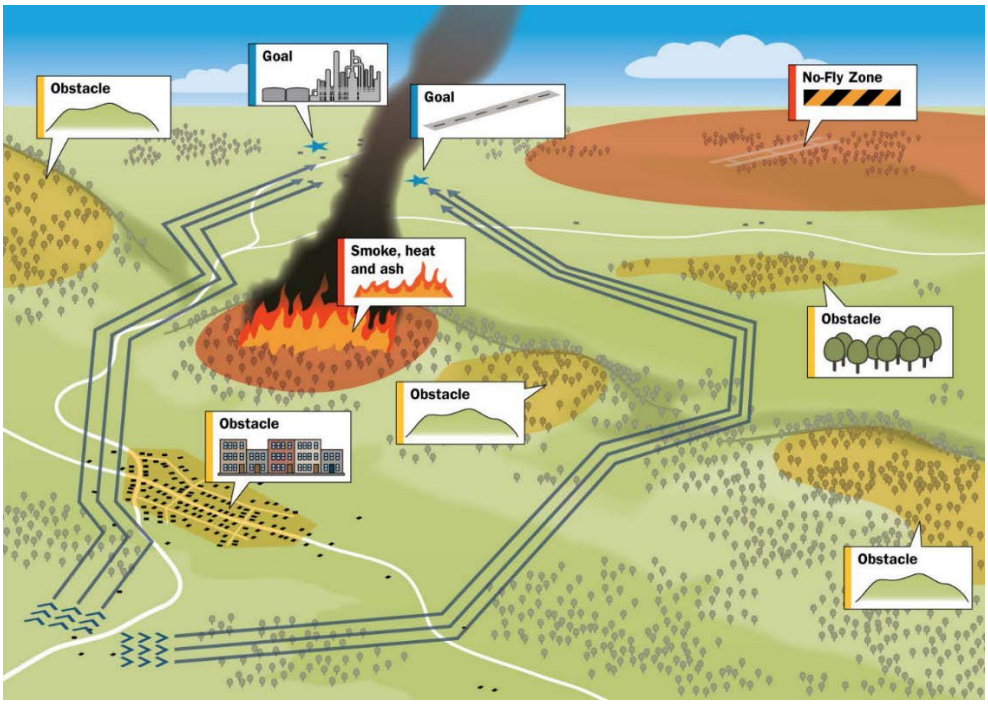

Figure 8: Deployment Example of Two Swarms

Appendix B PnG Solution Example

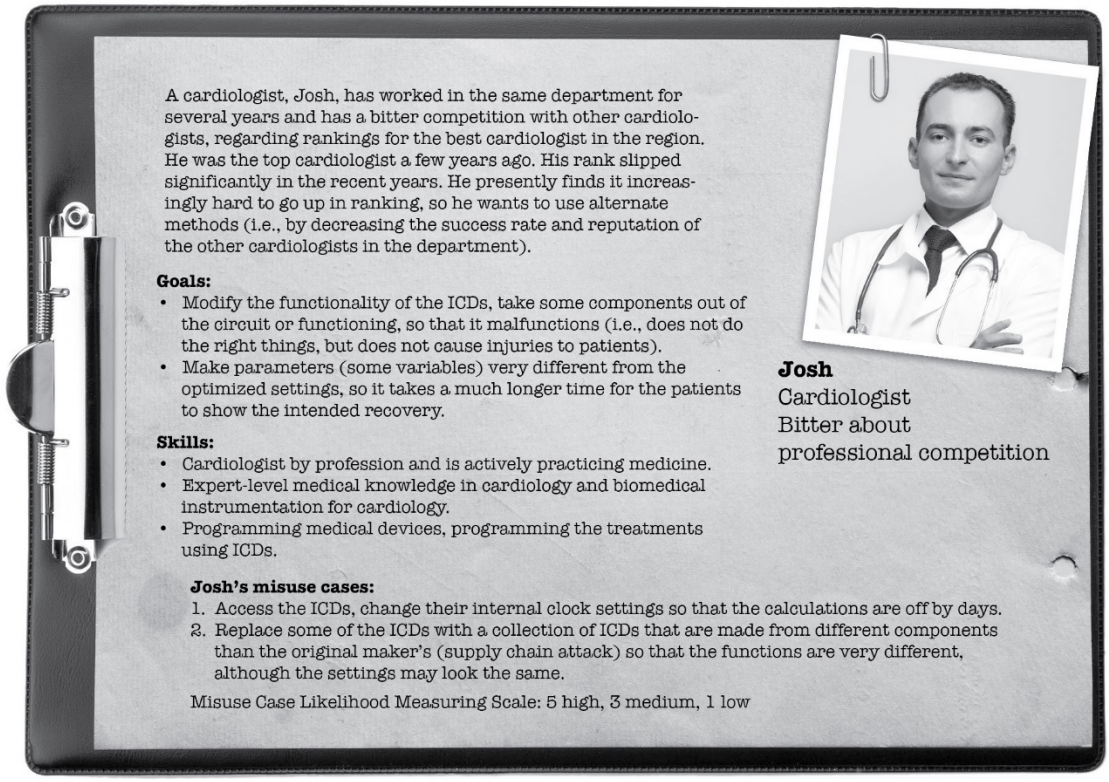

Figure 9: PnG Competing Cardiologist

Misuse Cases

The PnG profile in Figure 9 lists the following Misuse Cases:

1. Access the ICDs, change their internal clock settings so that the calculations are off by days.
2. Replace one of the ICDs with a collection of ICDs that are made from different components than the original maker's (supply chain attack) so that the functions are very different, although the settings may look the same.

Misuse Case Discussion: Case 1

How likely is this case?

Let us take a look at the function block diagram of a typical implantable cardioverter-defibrillator (ICD):

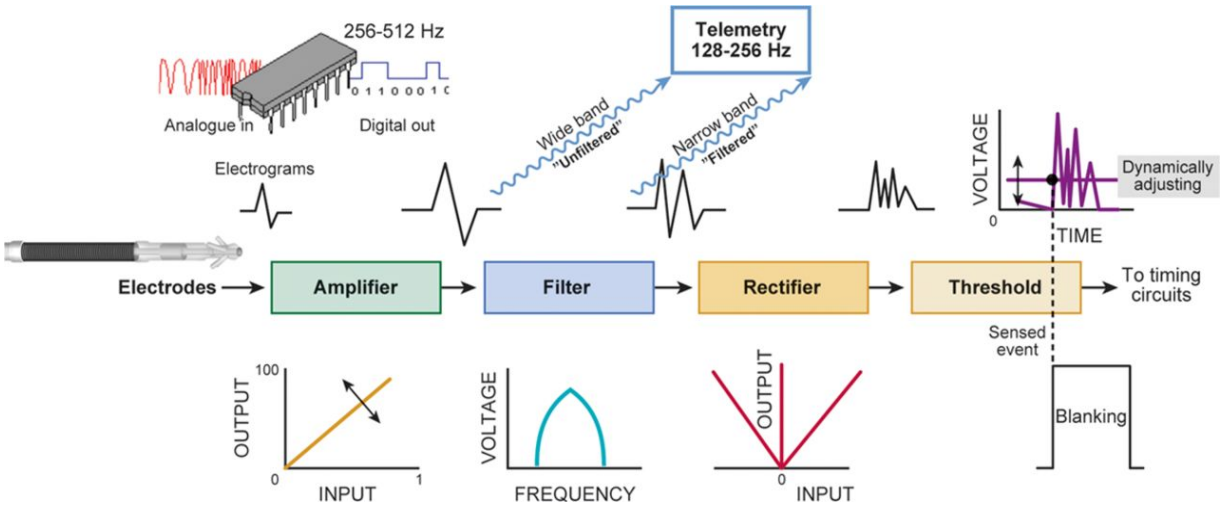

Figure 10: Functional Block Diagram of ICD
(Source: [Swerdlow 2014] http://circep.ahajournals.org/content/7/6/1237.figures-only)

Looking at the block diagram, one can imagine that the clock has a range of frequencies. By changing the frequency (i.e., making it different from what is initially set up), one should be able to modify the pulse patterns.

We can also see that there are several ways the ICD's performance can be modified from its intended operation.

Alternate modes of attack on the ICD: Change the amplifier gain, filtering parameters, rectifier parameters, change threshold.

Based on these vulnerabilities, which are not difficult to implement (if the attacker gets some time to play with the ICDs, then he could make the changes), we can easily give a high score of 4-5 for this misuse case.

What should the defender do to be prepared, and how much might it cost?

Consider some pulse shape verification algorithm or real-time pulse shape display so that the internal function of the chip can be continuously monitored to prevent a potential attack. The features are some minor changes in the programming, so it should take some time, but not too much cost if there is an internal IT team.

Misuse Case Discussion: Case 2

How likely is this case?

The block diagram shows that the ICD has three main components. It is most likely that these three components are integrated in a single chip. If this is the case, then the chip can be replaced with a duplicate one that has similar components—but its gains, filter, and rectification circuit are designed differently, so that the pulse shape is different.

Considering that the complexity of designing a chip that performs similar function as the original one involves manufacturing of a new chip, we can assign the likelihood as 2-3.

What should the defender do to be prepared, and how much might it cost?

The physician should be advised to physically examine each ICD internally to make sure the components are not compromised. If this is not possible for the physician, then an electronics engineer can do the inspection before the device is implanted into the patient. This additional protocol should not cost extra, it only requires planning to have the electronics engineer be present at the time of installing the ICD.

For additional reading on secure ICD programming, see the review article by Biffi [Biffi 2014].

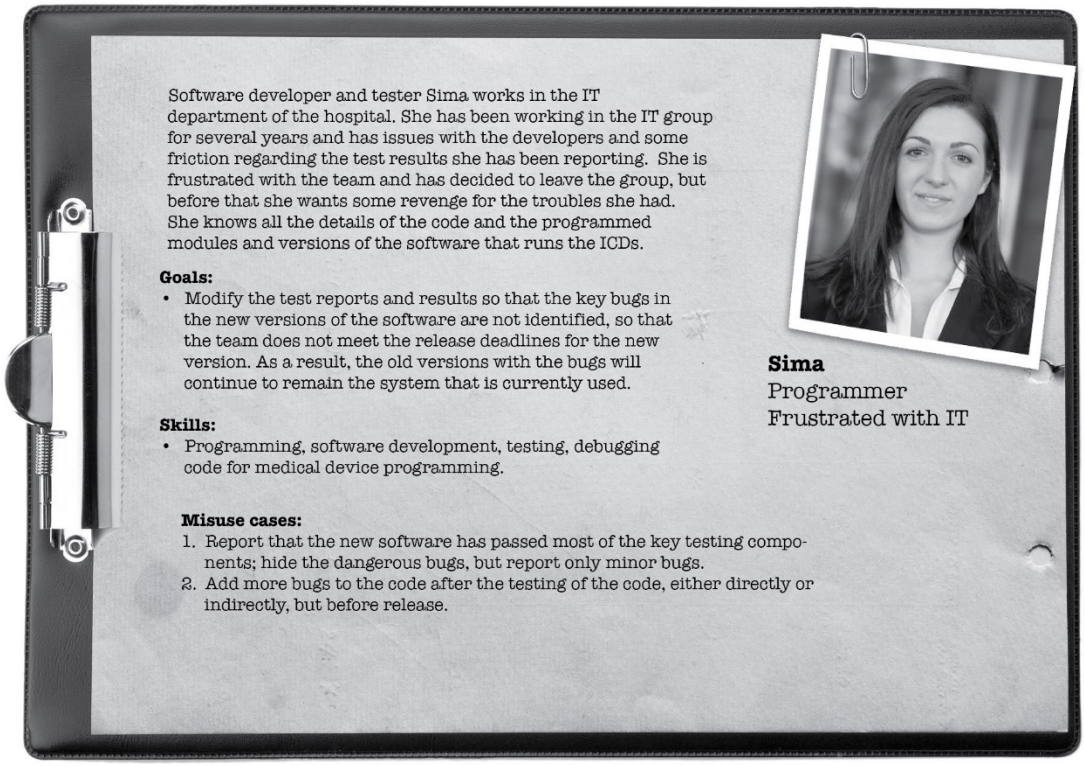

Figure 11: PnG Software Developer and Tester

Misuse Cases

The PnG profile in Figure 11 lists the following Misuse Cases:

1. Report that the new software has passed most of the key testing components; hide the dangerous bugs, but report only minor bugs.
2. Add more bugs to the code after the testing of the code, either directly or indirectly, but before release.

Misuse Case Discussion: Case 1

How likely is the case?

If there is only one tester in the team, then this case is easy to implement. We can assign a likelihood of 4-5.

What should the defender do to be prepared, and how much might it cost?

Have two independent experts do the testing. The second could be a part-time tester (either internal or external to the team) or a sub-contractor who is hired just before release for a second layer testing.

Misuse Case Discussion: Case 2

How likely is the case?

If there is only one tester in the team, and there is no additional testing as a backup as in Case 1, then this case is easy to implement. We can also assign a likelihood of 4-5.

What should the defender do to be prepared, and how much might it cost?

The solution is identical to that for Case 1. Have two independent experts do the testing. The second could be a part-time tester (either internal or external to the team) or a sub-contractor who is hired just before release for a second layer testing. For alternative ideas for testing before release, see the Software Testing Help site [Software Testing Help 2017]. In addition, simulation and modeling of the performance of the product can be undertaken as a semi-automated process to validate the new features.

Appendix C Security Cards Solution Example

Threat Modeling with Secure Cards for Drone Deliveries

We consider a drone or drone swarm that is on its way to deliver emergency supplies to the flood-affected populations after it is dispatched by the team consisting of local government authorities and their drone technology contractors. See Figure 12 for an example. The drones face several potential threats, both physical and cyber in nature. We consider a few scenarios of drone attack and how those attacks affect the drones and the people who depend on them.

Figure 12: Example of a Drone Swarm
(*Source: http://www.ioti.com/security/drones-are-coming-take-cover*)
(Same image as Figure 5)

Part 1: Ranks within each category with reason it is considered a potential threat

A Ranked Overview

Human Impact Cards:

1. Emotional well-being (those suffering from the disaster are deprived of basic commodities and get depressed—the primary subject of the threat)
2. Physical well-being (health is affected due to lack of timely food supplies and medicine—the primary subject of the threat)
3. Relationships (the relations between the people, local authorities, and government is at stake if the rescue mission fails—a secondary subject)

4. Unusual impacts (loss of property, loss of life, loss of trust in local government, loss of businesses—a secondary or ternary subject)

Adversary's Motivations:

1. Money (the goods stolen from the drones and the drones/components can be resold to make money—the profitable nature makes this rank 1!)
2. Warfare (some local trouble makers may see this as a route and non-violent means of attack—allowing ease of attack, i.e., without having to face other humans in the operations makes this rank above the rest)
3. Politics (oppositions and opposition groups may intrude to bring bad name to local government—can lead to change of power, so it becomes attractive)
4. Unusual motivations (hack the drones and use them for other unauthorized purposes such as flying in restricted zones or delivery of harmful goods or simply to destroy)

Adversary's Resources:

1. Expertise (the attacker has all the expertise to hack the brand of drones used in the mission)
2. Inside knowledge (access to inside knowledge makes the attack viable)
3. Money (money flowing in for political reasons to bring down the local government's reputation)
4. Inside capabilities (an insider who turns attacker can do a lot of damage to the drones)

Adversary's Methods:

1. Physical attack (shoot the drone with a drone gun)
2. Technological attack (jam the GPS or the rotors)
3. Multiphase attack (damage partially—perhaps damage one rotor and partially disable the drone to take control)
4. Manipulation or coercion (hack the drone information system and its GPS, then change the destination or send it back to the origin or make it lose its sense of direction)

Part 2: In-depth analysis of all the potential threats: Threat Insights

Human Impact Cards:

1. Emotional well-being (those suffering from the disaster are deprived of basic commodities and get depressed—the primary subject of the threat)
2. Physical well-being (health is affected due to lack of timely food supplies and medicine—the primary subject of the threat)
3. Relationships (the relations between the people, local authorities, and government is at stake if the rescue mission fails—a secondary subject)
4. Unusual impacts (loss of life, loss of trust in local government, loss of businesses—a secondary or ternary subject)

Type	Actor	Action	Target	Purpose	Result	Impact
Denial	Attacker	Attack Methods 1-4	Drone (Physical, Cyber)	Motivations 1-4	Human Impacts 1-4	1-High 2-High 3-Low 4-Low

Adversary's Motivations:

1. Money (the goods stolen from the drones and the drones/components can be resold to make money—the profitable nature makes this rank 1!)
2. Warfare (some local trouble makers may see this as a route and non-violent means of attack—allowing ease of attack, i.e., not having to face other humans in the operations makes this an above the rest)
3. Politics (oppositions and opposition groups may intrude to bring bad name to local government—can lead to change of power, so it becomes attractive)
4. Unusual motivations (hack the drones and use them for other unauthorized purposes such as flying in restricted zones or delivery of harmful goods or simply to destroy)

Type	Actor	Action	Target	Purpose	Result	Impact
1-Capture 2,3,4-Hack	Attacker	Intrusion	Drone	Misuse Drone	Adversary's Motivations 1-4	1-High 2,3,4-Low

Adversary's Resources:

1. Expertise (the attacker has all the expertise to hack the brand of drones that are used in the mission)
2. Inside knowledge (access to inside knowledge makes the attack viable)
3. Money (money flowing in for political reasons to bring down local government reputation)
4. Inside capabilities (an insider who turns attacker can do a lot of damage to the drones)

Type	Actor	Action	Target	Purpose	Result	Impact
Denial, Spoofing, Jamming, Screening	Attacker	Physical, Cyber Attacks	Drone, Physical, GPS, Accelerometer, Computer	Adversary's Motivations 1-4	Human Impacts 1-4	1,2,3,4 High

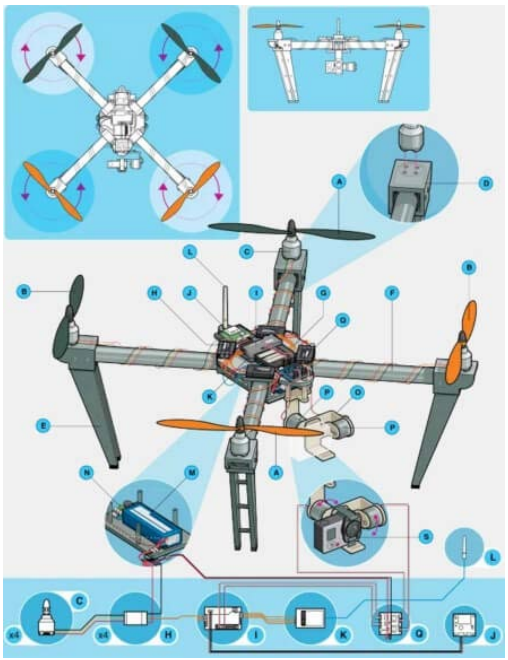

Figure 13: Example of Drone Components
(*Source: https://www.dronezon.com*)
(Same image as Figure 6)

Adversary's Methods:

To analyze how a drone can be subjected to an attack, let us consider an example. Figure 13 shows typical drone parts: Propellers, Brushless Motors, Motor Mount, Landing Gear, Boom, Drone Body Part, Electronic Speed Controllers, Flight Controller, GPS Module, Receiver, Antenna, Battery, Battery Monitor, Gimbal, Gimbal Motor, Gimbal Control Unit, Camera, Sensors, and Collision Avoidance Sensors. The attack can in principle be on any of the components. We consider a few most likely cases.

1. Physical attack (shoot the drone with a drone gun, direct objects, or spray a dark paint on drone's camera to blind the drone)
2. Technological attack (jam the GPS or the propellers)
3. Multiphase attack (damage partially, e.g., damage one propeller and partially disable the drone to take control)
4. Manipulation or coercion (hack the drone information system and its GPS, then change the destination or send it back to the origin or make it lose its sense of direction)

Type	Actor	Action	Target	Purpose	Result	Impact
Damaging, Capturing, Redirecting	Attacker	Shooting, Hacking, Modifying, Parameters	Drone, Physical, GPS, Accelerometer, Computer	Adversary's Motivations 1-4	Human Impacts 1-4	1,2,3,4 High

References

URLs are valid as of the publication date of this document.

[Anderson 1994]
Anderson, R. J. Why Cryptosystems Fail. *Communications of the ACM*. Volume 37. Issue 11. November 1994. Pages 32–40.

[Biffi 2014]
Biffi, M. ICD Programming. *Indian Heart Journal*. Volume 66. S88-S100. January, 2014. DOI:10.1016/j.ihj.2013.11.007.

[Cleland-Huang 2013]
Cleland-Huang, J. Meet Elaine: A Persona-Driven Approach to Exploring Architecturally Significant Requirements. *IEEE Software*. Volume 30. Number 4. June 2013. Pages 18–21. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6834694

[Cleland-Huang 2014]
Cleland-Huang, J. How Well Do You Know Your Personae Non Gratae? *IEEE Software.* Volume 31. Number 4. July 2014. Pages 28–31. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6834694

[Cleland-Huang 2016]
Cleland-Huang, J.; Denning, T.; Kohno, T; Shull, F.; & Weber, S. Keeping Ahead of Our Adversaries. IEEE Software. Volume 33. Number 3. May 2016. Pages 24–28. DOI: 10.1109/MS.2016.75

[Cooper 1999]
Cooper, A. The Inmates Are Running the Asylum. *Software-Ergonomie.* Berichte des German Chapter of the ACM. Volume 53. Page 17. 1999.

[Denning 2013]
Denning, T. A.; Friedman, B.; Kohno, T. Security Cards: A security threat brainstorming toolkit. 2013. http://securitycards.cs.washington.edu/index.html

[Dotan 2009]
Dotan, A.; Maiden, N. A. M; Lichtner, V.; & Germanovich, L. Designing with Only Four People in Mind? – A Case Study of Using Personas to Redesign a Work-Integrated Learning Support System. *INTERACT.* August 2009. Uppsala, Sweden. Volume 2. Pages 497–509. 2009.

[Hernan 2006]
Hernan, S.; Lambert, S.; Ostwald, T.; & Shostack, A. “Uncover Security Design Flaws Using the STRIDE Approach.” *MSDN Magazine*. Nov. 2006.

[Hilburn 2013]
Hilburn, T. B. & Mead, N. R. Building Security In: A Road to Competency. *IEEE Security & Privacy.* Volume 11. Number 5. 2013. Pages 89–92.

[Ingalsbe 2008]
Ingalsbe, J. A.; Kunimatsu, L.; Baeten T.; & Mead, N. R. Threat Modeling: Diving into the Deep End. *IEEE Software*. Volume 25. Number 1. January 2008. Pages 28–34.

[Kohnfelder 1999]
Kohnfelder, L. & Garg, P. The Threats to Our Products. *Microsoft Interface.* April 1999.

[Mead 2005]
Mead, N. R. & Stehney, T. Security Quality Requirements Engineering (SQUARE) Methodology. *ACM SIGSOFT Software Engineering Notes*. Volume 30. Number 4. July 2005. Pages 1–7.

[Mead 2017]
Mead, Nancy; Shull, Forrest; Spears, Janine; Hiebl, Stefan; Weber, Sam; & Cleland-Huang, Jane. Crowd Sourcing the Creation of Personae Non Gratae for Requirements-Phase Threat Modeling. Pages 404–409. *IEEE International Requirements Engineering Conference Proceedings*. September 2017. DOI 10.1109/RE.2017.63.

[Microsoft 2017]
Microsoft Corporation. SDL Threat Modeling Tool. *Security Development Lifecycle*. November 10, 2017 [accessed]. https://www.microsoft.com/en-us/sdl/adopt/threatmodeling.aspx

[Morana 2015]
Morana, M. M. & UcedaVelez, *T. Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis.* Wiley-Blackwell. 2015.

[Morasco 2017]
Marasco, E.; Cukic, B.; Shehab, M.; & Usman, R. Attack Trees for Protecting Biometric Systems against Evolving Presentation Attacks. *16th Annual IEEE International Conference on Technologies for Homeland Security (HST) 2017*. Boston, MA. April 2017. Pages1–6. http://wpage.unina.it/emanuela.marasco/Marasco_HST17.pdf

[Nielsen 2013]
Nielsen, L. Personas – User Focused Design. *Human-Computer Interaction Series*. Volume 15. Springer, 2013.

[Opdahl 2009]
Opdahl, A. L. & Sindre, G. "Experimental comparison of attack trees and misuse cases for security threat identification." *Information & Software Technology*. Volume 51. Number 5. 2009. Pages 916–932. 2009.

[Putnam 2012]
Putnam, C.; Kolko, B. E.; & Wood, S. Communicating about users in ICTD: leveraging HCI personas. *Proceedings of the Fifth International Conference on Information and Communication Technologies (ICTD 2012)*. Atlanta, Georgia. March 2012. Pages 338–349.

[Robertson 2006]
Robertson, S. & Robertson, J. *Mastering the Requirements Process*. Addison Wesley. 2006.

[Saitta 2005]
Saitta, P.; Larcom, B.; & Eddington, M. *Trike v.1 Methodology Document* [Draft]. July 2005. http://octotrike.org/papers/Trike_v1_Methodology_Document-draft.pdf

[Scandariato 2015]
Scandariato, R.; Wuyts, K.; & Joosen, W. A descriptive study of Microsoft's threat modeling technique. *Requirements Engineering*. Volume 20. Number 2. Jun. 2015. Pages 163–180.

[Shostack 2014]
Shostack, A. *Threat Modeling: Designing for Security*. John Wiley & Sons, 2014.

[Shull 2016a]
Shull, F. Evaluation of Threat Modeling Methodologies. Software Engineering Institute, Carnegie Mellon University. 2016. http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=474197

[Shull 2016b]
Shull, F. & Mead, N. Cyber Threat Modeling: An Evaluation of Three Methods. *SEI Blog*. November 11, 2016. https://insights.sei.cmu.edu/sei_blog/2016/11/cyber-threat-modeling-an-evaluation-of-three-methods.html

[Sindre 2008]
Sindre, G. & Opdahl, A. "Misuse Cases for Identifying System Dependability Threats." *Journal of Information Privacy and Security*. Volume 4. Number 2. 2008. Pages 3–22.

[Software Testing Help 2017]
7-step practical implementation of manual testing before product release. *Software Testing Help*. April 17, 2017. http://www.softwaretestinghelp.com/practical-implementation-of-manual-testing/

[Spears 2014]
Spears, J. & Erete, S. "i have nothing to hide; thus nothing to fear": Defining a framework for examining the 'nothing to hide' persona. Pages 1-5. In *Tenth Symposium on Usable Privacy and Security (SOUPS 2014)*. Menlo Park, CA. July 2014.

[Swerdlow 2014]
Swerdlow, C. D.; Asirvatham, S. J.; Ellenbogen, K. A.; Friedman, P. A. Troubleshooting Implanted Cardioverter Defibrillator Sensing Problems I. *Circ. Arrhym Electrophysiol.* Volume 7. Number 6. December 2014. Pages 1237–1261. http://circep.ahajournals.org/content/circae/7/6/1237.full.pdf

[ThreatModeler 2017]
Real World Threat Modeling. *ThreatModeler*. December 11, 2017 [accessed]. http://threatmodeler.com/

[UcedaValez 2012]
UcedaVelez, T. *Real World Threat Modeling Using the Pasta Methodology*. Technical report. Open Web Application Security Project (OWASP). 2012.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	March 2018	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
A Hybrid Threat Modeling Method	FA8721-05-C-0003

6. AUTHOR(S)

Nancy R. Mead, Forrest Shull, Krishnamurthy Vemuru (University of Virginia), Ole Villadsen (Carnegie Mellon University)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2018-TN-002

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFLCMC/PZE/Hanscom Enterprise Acquisition Division 20 Schilling Circle Building 1305 Hanscom AFB, MA 01731-2116	n/a

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

In FY 2016, the research team evaluated Security Cards, STRIDE (Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service, Elevation of privilege), and persona non grata (PnG) for effectiveness in threat identification. Security Cards is an approach that emphasizes creativity and brainstorming over more structured approaches such as checklists. STRIDE involves modeling a system and subsystem and related data flows. PnGs represent archetypal users who behave in unwanted, possibly nefarious ways. The team used two scenarios: an aircraft maintenance scenario and a drone swarm scenario, both described in this technical note in detail, along with the project outcomes. No individual threat modeling method included all identified threats.

The research team subsequently developed the Hybrid Threat Modeling Method (hTMM), considering the desirable characteristics for a Threat Modeling Method. At a high level, the hTMM includes the following steps, described in detail in the technical note: (1) Identify the system you will be threat modeling. (2) Apply Security Cards according to developers' suggestions. (3) Prune PnGs that are unlikely or for which no realistic attack vectors could be identified. (4) Summarize results from the above steps, utilizing tool support. (5) Continue with a formal risk assessment method.

14. SUBJECT TERMS	15. NUMBER OF PAGES
persona non grata, PnG, Security Card, STRIDE, misuse case	52

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18
298-102