SESSION ID: LAB3-R02

THREAT MODELS: INTO THE DEEP!

DEMYSTIFYING THREAT MODELING: THE ATASM PROCESS†

Brook Schoenfield
Author, Principal Engineer, Technical Lead, Curious Questioner McAfee

† ATASM from Security Systems, © 2015, Brook S.E. Schoenfield, used with permission
Brook S.E. Schoenfield, MBA
Principal Engineer

• >1000 security reviews
• 17+ years security
• 30+ years software/hitech
• Real-time OS, TCP/IP stack down to hardware
• 5th virtual team lead, 9th technical leader role
• 4th software security lead @ 4th organization

• SANS Institute Featured Speaker, contributor GWEB certification
• Author:
  – Core Software Security, CRC Press, 2014
  – Avoiding The Top 10 Security Design Flaws, IEEE
  – Various SANS Smart Guides, papers, etc.

• Founding member IEEE Center for Secure Design
• Featured security architect at Bletchley Park Cyber Museum
• >1000 security reviews
• 16+ years security
• 30+ years software/hitech
• Real-time OS, TCP/IP stack down to hardware
• 5th virtual team lead, 9th technical leader role
• 4th software security lead @ 4th organization

• SANS Institute Featured Speaker, contributor GWEB certification
• Author:
  – Core Software Security, CRC Press, 2014
  – Avoiding The Top 10 Software Security Design Flaws, IEEE
  – Various SANS Smart Guides, papers, etc.

• Founding member IEEE Center for Secure Design
• Featured security architect at Bletchley Park Cyber Museum
First Comments

- ATASMS\(^1\) is a pedagogy
- ATASMS is a mnemonic, a high level abstraction for a process that is often non-linear, and highly recursive
- “Peel the onion” is close to reality, but harder to learn and to coach: a real threat model process is often quite fractal
- You are encouraged to use ATASMS; if useful, please teach

---

1. ATASMS\slash\textregistered\ material © Brook S.E. Schoenfield, 2010-2015, all rights reserved, CRC Press, 2015. Used with permission
Possible Outcomes From This Session

• Newbies: Introduction to threat modeling concepts
• Learners: Dig deeper, gain consistency, increase comfort, practice
• Practitioners: Conceptualization and explanation; a method that fosters inclusion
• Experts: Articulate in order to present and to teach
• Everyone: Hopefully, some new tricks of the trade?
What Can Be Said About This “Architecture”?
Is This Any Better?
Is there a recurring design problem: Have you heard about or experienced poor security design choices?

Break into work groups (4-6) to brainstorm examples of insecure design.
Design brainstorm report back from workgroups
Yes, That Jeep Cherokee

Please consider this design...

Entertainment system (linux), externally connected to untrusted network is given full access to the internal network (bus) for life/death systems like ignition and brakes.
WiFi Regulated Hearts?

“I realized that my heart was now wired into the medical Internet of Things, and this was done without informing me or asking for my consent.” MARIE MOE, 03.14.16, GO AHEAD, HACKERS. BREAK MY HEART

• “…pacemaker technology and is designed to communicate wirelessly with a nearby external programmer in the 175 kHz frequency range. After partially reverse-engineering the ICD's communications protocol with an oscilloscope and a software radio, we implemented several software radio-based attacks that could compromise patient safety and patient privacy.”

Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses, Daniel Halperin; Univ. of Washington, Seattle, WA; Thomas S. Heydt-Benjamin; Benjamin Ransford; Shane S. Clark more authors, 2008 IEEE Symposium on Security and Privacy (sp 2008)
It’s not immediately clear why Target would have given an HVAC company external network access, or why that access would not be cordoned off from Target’s payment system network.
Current Events!

- Meltdown
- Spectre

- Vulnerabilities?
- Design flaws?
- Poor design?
- Intentional hiding?
“Kernel-memory-leaking Intel processor design flaw forces Linux, Windows redesign”

- John Leyden and Chris Williams 2 Jan 2018 at 19:29, The Register

Just because The Register says it, and twitter explodes, doesn’t make it so...
Consider The Bronze Age

This “design” (technology) worked as intended for ~2000 years
Until “Research” Found A Weakness: Strength (technique == iron)

This “design” worked for 1000 years steadily improving steel, which then fell to gun powder…
My Crystal Ball Isn’t That Good

- Threat modeling must account for known attack types
- Threat modeling accounts for known and imagined *credible* actors
- Threat modeling imagines new attack scenarios based upon existing attack types
- Threat modeling is not typically sufficiently prescient to account for research advances in the distant future

Important: threat modeling must be grounded in reality in order to remain relevant
• Threat modeling must account for known attack types
• Threat modeling accounts for known and imagined actors
• Threat modeling imagines new attack scenarios based upon existing attack types
• Threat modeling is not typically sufficiently prescient to account for research advances in the distant future

Important: threat modeling must be grounded in reality in order to remain relevant
A start...


Creative Commons Attribution-ShareAlike 3.0 license.
Threat modeling is a technique to identify the attacks a system\(^1\) must resist and the defenses that will bring the system to a desired defensive state

1. “system” is defined inclusively
The Timing Component: Earlier Is Better

Early requirements get the "worm"
Acknowledgement and Planning

Or at least, early requirements capture team mindshare
Start the threat model at the point when the architecture is sufficiently defined such that most if not all of the major functional components are understood.

Revisit the threat model when the architecture materially changes.

Threat models are living documents.

Threat modeling is compatible with iterative development practices. Iterate the threat model appropriately.
Tool Belt

Architecture is a tool for structuring complexity

Architecture provides a playground for potential changes

Architecture Risk Assessment (ARA) is a tool for applying a threat landscape to an architecture to uncover security needs

Threat modeling is a tool for secure design.

A threat model is not a “design”

Threat model is dependent upon architectural understanding

Threat models output security requirements
Everything You Know About Security...

Architecture is a tool for structuring complexity

Architecture is a tool for playing with potential changes

Architecture Risk Assessment (ARA) is a tool for applying a threat landscape to an architecture to discover security needs

--- Threat modeling is applied security architecture

Threat modeling is a part of architecture design.

A threat model is not a “design”

Threat model is dependent upon architectural understanding

Threat models output security requirements
“Secure” Software Must:

• Be rid of implementation errors with exploitable effects
• Contain customers’ security features
• Be self-protective
• Minimize consequences of successful attack ("fail well")
• Install with sensible, “closed” defaults
Design Secure Software

- Be rid of implementation errors with exploitable effects
  - Contain customers’ security features
  - Be self-protective
  - Minimize consequences of successful attack (‘‘fail well’’)
  - Install with sensible, ‘‘closed’’ defaults

*ARA & threat modeling help with these tasks*
What Can We Infer?
What More Do We Need And Why?
Prework: The 3 S’s

Strategy
- Threat landscape
- Risk posture

Structures
- Possible controls
- Existing limitations

Specification
- Data sensitivity
- Runtime/execution
- Deployment
Background Information Is Critical

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Structures</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat landscape</td>
<td>Existing limitations</td>
<td>Data sensitivity</td>
</tr>
<tr>
<td>Risk posture</td>
<td></td>
<td>Runtime/execution</td>
</tr>
</tbody>
</table>

There is no book of “correct” answers. Each organization is different. Investigate yours...
A threat model is a crossroads of knowledge from architecture experts, domain experts, and security experts. Absence of one or more stakeholders cripples the model and its usefulness.
Personal accelerometer

- Measures movement of body
- Worn on wrist
- Extremely low power
- Long battery life
- Minimal operating system and CPU
Low Power Bluetooth Specification

• There is no protection of communications!
  • E.g., TLS not used over open air
• Authentication by device ID only
• Devices only exchange movement data
Understanding system architecture is key

Differing views and levels of detail each solve distinct problems

There is no perfect view or level
A Threat Model From This?
Web-Sock-A-Rama Web Business

• Successful Online Sock store
• All kinds of socks: men, women, children, specialty, sports, artistic, novelty
• Takes no public political positions
• Relatively risk averse
• Customers are considered the business’ greatest resource
  • Customer security and privacy prime objectives
• Assume that the support and administration of the infrastructure is run rigorously:
  • strong access controls and need-to-know (e.g., NIST-800-53): networks, hosts, execution environments
  • External network separated from internal
  • Administration is via highly controlled management network
Start At High Level

- What is the organization’s purpose?
- What is this system’s purpose?
- How does this system contribute to the organization goals? To an enterprise architecture?
- What are the major functions?
- How do these interact? Why?
- Derive, if you can, the intended risk posture
Web-Sock-A-Rama
Find an attack vector
Logical Or Component View
ATASM

Architecture  Threats  Attack Surfaces  Mitigations

Threat Modeling = Applied security architecture

Substeps

1. Architecture
   - Understand the Logical and Component Architecture of the System
   - Understand every communication flow and any valuable data that is moved or stored
   - Apply new security controls to the set of attack surfaces for which there isn't sufficient mitigation
   - Build a Defense in Depth

2. Threats
   - List all the possible Threat Agents for this type of System
   - List the Ultimate goals for each of these Threat Agents
   - List the typical attack methods for these type of Threat Agents
   - List the system level objectives of Threat Agents using their attack methods
   - Decompose the Architecture to a level that exposes every attack surface
   - Apply Attack methods for expected objectives to the attack surfaces
   - Filter out Threat Agents who have no attack surfaces applied to their typical attack methods
   - List all existing security controls for each attack surface
   - Filter out attack surfaces for which there is sufficient existing protection

3. Attack Surfaces

4. Mitigations
A Highly Recursive Process
Threat modeling is a technique to identify the attacks a system must resist and the defenses that will bring the system to a desired defensive state.

By Sung Lee; concept by David Wheeler; used with permission
Adapted from original by Sung Lee; concept by David Wheeler; used with permission
The outcome, the “point” is to discover the necessary security requirements

“Security needs tend to get expressed as system constraints” – Erik Simmons
Digging into threat agents and their methods

Focus on human actors

Adversaries are adaptable and creative
### Table 2.1 Summarized Threat Attributes

<table>
<thead>
<tr>
<th>Threat Agent</th>
<th>Goals</th>
<th>Risk Tolerance</th>
<th>Work Factor</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyber criminals</td>
<td>Financial</td>
<td>Low</td>
<td>Low to medium</td>
<td>Known proven</td>
</tr>
<tr>
<td>Industrial spies</td>
<td>Information and disruption</td>
<td>Low</td>
<td>High to extreme</td>
<td>Sophisticated and unique</td>
</tr>
<tr>
<td>Hacktivists</td>
<td>Information, disruption, and media attention</td>
<td>Medium to high</td>
<td>Low to medium</td>
<td>System administration errors and social engineering</td>
</tr>
</tbody>
</table>
• Divide paper into 5 columns
  **Threat Agent, Goals, Technical ability (Methods), Risk tolerance, Work Factor**
  • Start by asking, “Who will want to attack this piece of software, when it runs under its normal, expected deployment?”
  • No discussion about the merits of an idea. (brainstorm principle)
  • Collate brainstorm ideas into the larger buckets. Be stereotypical
  • Derive your group’s threat agent landscape
Table 2.1 Summarized Threat Attributes

<table>
<thead>
<tr>
<th>Threat Agent</th>
<th>Goals</th>
<th>Risk Tolerance</th>
<th>Work Factor</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyber criminals</td>
<td>Financial</td>
<td>Low</td>
<td>Low to medium</td>
<td>Known proven</td>
</tr>
</tbody>
</table>

- Divide paper into 5 columns
- **Threat Agent, Goals, Technical ability (Methods), Risk tolerance, Work Factor**
- Start by asking, “Who will want to attack this piece of software, when it runs under its normal, expected deployment?”
- No discussion about the merits of an idea. (brainstorm principle)
- Collate brainstorm ideas into the larger buckets. Be stereotypical
- Derive your group’s threat agent landscape
Put our matrices together
Set priority with risk

Risk != vulnerability
Threat != vulnerability
Threat != exploit
Set Priority With Risk

A usable risk rating method *should* be:
- Lightweight
- Fast
- Simple and understandable
- Intuitive

Risk = Probability * Annualized loss
Set Priority With Risk

• A usable risk rating method *should* be:
  • Lightweight
  • Fast
  • Simple and understandable
  • Intuitive

\[ \text{Risk} = \text{Probability} \times \text{Annualized loss} \]

We haven’t actuarial tables
Successful attack requires an active threat agent who has the motivation to attack that particular system and who uses the threat agent's skills (exploit) via an exposed (attack surface) exploitable weakness (vulnerability) that can be manipulated by that attacker.

**Threat + exploit + exposure + vulnerability**

**The “kill chain”**
"Credible Attack Vector" (CAV)

CAV == Threat & Exploit & Exposure & Vulnerability
(consider each term as essentially boolean for this narrow context)

Interrupt any CAV term and you interrupt the kill chain

E.g., Threat & Exploit & !Exposure & Vulnerability !≠ CAV
(break the kill chain; eliminate the CAV; lower risk exposure)

Given 0 < CAV < 1, then risk might be expressed as:

Risk Rating = CAV * Impact
(essentially how JGERR rates risks)
Now That You Know, Let’s Have Some Fun…

- Threats
- Architecture
- Attack Surfaces
- Mitigations

Our goal is to experience each step, not to finish the threat model

- Yes, I switched the order for teaching purposes
Threat modeling Examples

Fictitious but based on real systems
Logical Or
Component View
Analytics Data Harvesting
Analytics Data Harvesting Attack Surface
System Components
Holistic Analysis Will Protect Data Sources
System Components Attack Surfaces
System Components Attacks At Assets
User Roles & Flows
User Roles & Flows Attack Surfaces
Identity Services: Who’s In Danger?
Mobile Security System

- Must reside in OS application sandbox
- Intercept in system to grab privileged events of interest for examination
- Intercept at higher privilege
- Intercept initialized early boot
- Intercept proxies events to engine
- Engine contains decision logic
- UI starts engine & communicate
- Files parsed/normalized by file opener
- External communications proxied through communicate
- Notifications from cloud through OS push notifications
- All message exchanges are initiated from device
  - Response to notification
- Device certificate/private key issued at enrollment
- Messages/updates signed by cloud services
Mobile Security App
Break Into Teams To Analyze
Consider Server-side Components
After the threat model, then what?

Prioritization is one of the hardest problems
Simple Threat Model Steps (Won’t likely be linear)

Enumerate CAV
Define and score impacts
Enumerate existing mitigations
Develop requirements to bring system to desired posture
Prioritize
Share and review with entire team + product management
How Do We Prioritize?

- Risk rate each CAV+impact that is not sufficiently protected
- Decide where the risk “line” is (low, medium, only high)
- Engage stakeholders in constructive dialog
- Write exceptions for items that should get implemented but must wait
Getting requirements implemented is an entirely different story...

(feel free to ask)
Getting requirements implemented is an entirely different story...

(feel free to ask)

Composing requirements that can and will be built is a distinct discipline
Hints For Clearer Requirements

• Separate the functional need from any solution
  • Tell designers what function is required, not how to build that function

• Give implementers room for creativity and innovation
  • Don’t lock in obsolescence

• The amount of detail required is inverse to the amount of skill and knowledge of the implementers
  • Skilled teams need only understand what functionality is required
  • Unskilled implementers will need lots of detail
Closing thoughts
What kind of docs must you have?

• The most important document is the requirements output from threat modeling
  • A threat model may be inferred from a thorough security requirements document
  • You must produce a requirements document

• If visual, one or more visual depictions of the architecture
  • Different domains often require different views onto the same system
  • Views might include:
    • attack surfaces
    • Assets
    • Mitigations and controls
Next Steps

- Find an experienced mentor and 2 peer reviewers
  - Independence helps

- Just do it!
  - Threat models need to be revisited when architecture, threat landscape, and security features change

- Experienced threat modelers, please make yourself available!

- Start with your own projects
  - Gather the team and wrestle with the threat model
  - To gain experience, you can help with other threat models
    - Modeling diverse architecture deepens abilities
Shameless Self-promotion

https://www.facebook.com/securingsystems
RSA Book Store, South Lobby
1:30PM Today, Thursday
A Threat Modeling Library

https://www.facebook.com/securingsystems
Some Resources

https://www.owasp.org/index.php/Application_Threat_Modeling
https://www.owasp.org/index.php/Threat_Risk_Modeling
https://www.owasp.org/images/a/aa/AppSecEU2012_PASTA.pdf
http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=427321
https://www.facebook.com/securingsystems
https://www.facebook.com/softwaresec
www.amazon.com/Core-Software-Security-Source
Where To Find Me

brook_schoenfield@mcafee.com
http://www.brookschoenfield.com
brook@brookschoenfield.com
@BrkSchoenfield

1. I apologize in advance. I only Linkedin with people with whom I've had meaningful interaction. Thanks.
1. I apologize in advance. I only LinkedIn with people with whom I’ve had meaningful interaction. Thanks.
Q & A