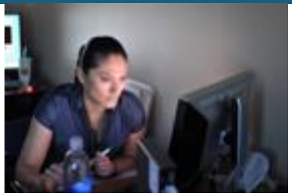
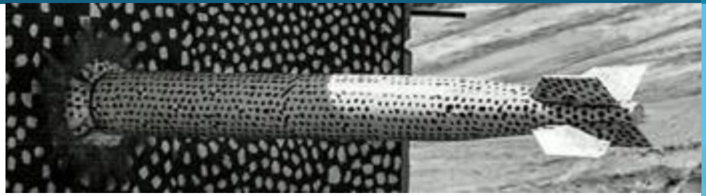
Red-Teaming and Cyber Risk Assessment at Sandia





PRESENTED BY

Taylor McKenzie

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Background

- Received Ph.D. in Economics from the University of Oregon in 2017
  - Focus in industrial organization and econometrics
  - Efficiency analyses (technical and allocative)
  - Structural modeling of pricing and economies of scale
- Experience interning at Pacific Northwest National Laboratory
  - Disease modeling
  - Nuclear proliferation pathway analysis
  - Social media analytics
  - Social network analysis for cyber vulnerabilities
- Currently a Senior Cybersecurity Researcher at Sandia National Laboratories
  - Risk and resilience analyses
  - Statistical analyses for variety of projects

Outline

1. Risk, resilience, and vulnerability analysis
2. Red-teaming and the Sandia's process
   - How we approach vulnerability analysis
   - Limitations of our approach and what we've done to address customer needs (blue slides)
3. Lessons learned

Risk, Resilience, and Vulnerability Analysis

- Motivating question: How can weaknesses in a system affect ability to perform its mission?
- Risk analysis: Threats (from any source), vulnerability to those threats, and consequence
- When all values are quantitative and relevant probabilities are known:

$$Risk = \sum_{T} \Pr(T) \times \left(\sum_{V} \Pr(V|T) \times \left(\sum_{C} C \times \Pr(C|T,V) \right) \right)$$

- Quantitative risk analysis can be difficult, especially for cyber systems
  - Broad threat landscape
  - Vulnerability set is not well-understood; many vulnerabilities exist but are not known
  - Can be difficult to determine how threats and vulnerabilities lead to consequences

Risk, Resilience, and Vulnerability Analysis

- There are several alternative methods to address difficulties with quantitative risk analysis
- Resilience analysis: Vulnerability to a specific threat and resulting consequences; focus on system's ability to withstand and quickly recover from disruption
- Vulnerability analysis: Specific consequence of concern and vulnerabilities that can lead to that consequence
- Qualitative risk analysis: Score threat, vulnerability, and consequence on a qualitative scale; develop method to combine qualitative scores into overall risk
  - Useful when some information is available, but not enough to use quantitative risk framework

Qualitative Risk Assessment: Prioritizing Improvements

- The Cyber Security Advisor (CSA) group under the Department of Homeland Security performs cybersecurity assessments with the goal of improving security measures for high-risk organizations
- CSA has relatively limited resources, wanted a method and tool to prioritize engagements (conferences, industry working groups, etc.)
- We focused on measuring cyber risk posed to organization and the likelihood that engagements will lead to eventual assessments

Qualitative Risk Assessment: Prioritizing Improvements

- Likelihood of successful engagement could be assessed quantitatively using historical engagement data
  - Some problems with endogeneity and separating effects (e.g., is an advisor particularly successful, or do most engagements with a particular industry lead to assessments?)
- Relied on qualitative measures of threat, vulnerability, and consequence
  - Some intuition on types of targets adversaries would target, vulnerabilities present, and potential impacts
  - Example: Larger organizations face higher threat and greater consequence than smaller organizations
- Developed initial method of combining measures into overall risk
- Polled CSA with hypothetical engagements, asked them to score each component and overall risk
  - Used for improving and validating methodology

Vulnerability Analysis and Red Teaming

- Vulnerability analysis often takes the form of red teaming
  - Red teaming: Assuming the role of an adversary to identify vulnerabilities and their consequence
- There are many flavors of red teaming
- Sandia's main red teaming effort is unique in a few ways
  - Information Design Assurance Red Team (IDART)
  - Considers a realistic adversary for a given target
    - An adversary would not launch a sophisticated attack to break into your home wireless network
  - Considers a realistic attack path, including a combination of cyber and physical actions
    - Why do something the hard way?
  - Documents the process and areas of improvement

IDART Process

1. Identify consequences of concern (nightmare scenarios)
2. Define a model of a realistic adversary, including skills and tolerance for risk
3. Collect information about the system, including potential weaknesses and attack vectors
4. Identify and document easiest (low effort, low likelihood of detection) attack paths that result in consequences of concern
5. (Optional) Demonstrate attack paths to show feasibility

IDART Process

1. Identify consequences of concern (nightmare scenarios)
2. Define a model of a realistic adversary, including skills and tolerance for risk
3. Collect information about the system, including potential weaknesses and attack vectors
4. Identify and document easiest (low effort, low likelihood of detection) attack paths that result in consequences of concern
5. (Optional) Demonstrate attack paths to show feasibility

Nightmare Consequences

- What consequences can the customer absolutely not tolerate?
- Advantages:
  - Impacts are clear and understandable to the decision-maker
  - Improving vulnerabilities that lead to high-consequence impacts can have greatest return
  - Can be easier to identify and explain attack paths that result in high-consequence outcomes
- Disadvantages:
  - There can be other consequences that are high-impact but not "the worst thing that can happen"
  - High-impact consequences the customer is less familiar with may be overlooked

Categorizing Consequences

- It can be helpful to categorize consequences to identify potential attack paths
- For information systems, consequences/capabilities measured in terms of[1]
  - Confidentiality: Ability to protect information from unauthorized access
  - Integrity: Ability to protect information from unauthorized changes
  - Availability: Ability to provide access to system or services when requested
  - (Less frequently) Accountability/non-repudiation: Ability to track actions on a system and associate them with specific actors[2]
  - (Less frequently) Assurance: Ability to trust metrics and system analysis to ensure it performs its mission
- Nightmare consequence examples
  - Systems that store personal information: Confidentiality
  - Power system: Availability
  - Power system information/control infrastructure: Integrity and Availability

1. See Stoneburner, Gary, Clark Hayden, and Alexis Feringa. *Engineering principles for information technology security (a baseline for achieving security)*. Booz-Allen and Hamilton Inc Mclean VA, 2001.
2. For example, see RAND Report 2395: Olympic-Caliber Cybersecurity

Consequences More Generally

- For other projects/applications, we want to quantify consequence impact beyond "the worst thing that can happen"
  - These consequences may be less understood by decision-makers (e.g., system-level impacts)
  - It may be difficult to determine exactly how consequences propagate
  - It can be difficult/impossible to numerically quantify consequences, necessitating qualitative measures
- Involved in a few projects that dealt with these issues

Projects Quantifying Consequence: Network simulation

- Quantifying resilience of enterprise network to various cyber attacks
- Emulytics™: Simulation of real cyber system with high fidelity
  - Ability to simulate individual components, interaction between components
  - Ability to plug in real hardware
  - See how the system would respond to a disruption without affecting real operations
- Able to simulate network and observe system-level values (e.g., latency, connectivity, uptime)
  - Included hosts (e.g., running Windows and additional software), servers, communications
  - Tested various disruption scenarios: Denial of service, random/systematic outages
- Some problems:
  - System-level values are noisy
  - Decision-makers need higher-level information
  - Difficult to understand exactly how system-level impacts affect ability to perform mission

Projects Quantifying Consequence: Network simulation

- Performed state estimation to estimate true quantities of interest
  - Aimed to describe ability of hosts to communicate with servers and each other
  - Example: If a given host made a request to a given server, with what probability could it expect the response is received? How long could the host expect to wait for a reply?
- Deliberated with team and people familiar with system to convert system-level metrics to higher levels
  - Measures of confidentiality, integrity, and availability for services/capabilities provided by system
  - Measures of ability to perform specific actions given system attributes

IDART Process

1. Identify consequences of concern (nightmare scenarios)
2. Define a model of a realistic adversary, including skills and tolerance for risk
3. Collect information about the system, including potential weaknesses and attack vectors
4. Identify and document easiest (low effort, low likelihood of detection) attack paths that result in consequences of concern
5. (Optional) Demonstrate attack paths to show feasibility

Adversary Models: Generic Threat Matrix

- For IDART, hypothetical adversaries are qualitatively rated along six attributes
  - Intensity: Risk tolerance for getting caught and negative consequences
  - Stealth: Ability to maintain secrecy through attack
  - Time: Amount time the adversary is willing to spend planning, developing, and deploying attack
  - Technical personnel: Size of technical team
  - Knowledge: Level of cyber and other knowledge possessed by adversary
  - Access: Ability to access facility or system, either by opportunity, force/coercion, or insider assistance
- Attribute ratings are then aggregated into an overall adversary level using the generic threat matrix (1-High, 8-Low)
  - This is a subjective, deliberative process, often end up with more than one possible categorization
- This process is based on SAND Report 2007-5791: Categorizing Threats
  - Reviews actual attacks and characteristics of real adversaries
  - Develops abstract method of categorizing threats that can be used openly

Generic Threat Matrix

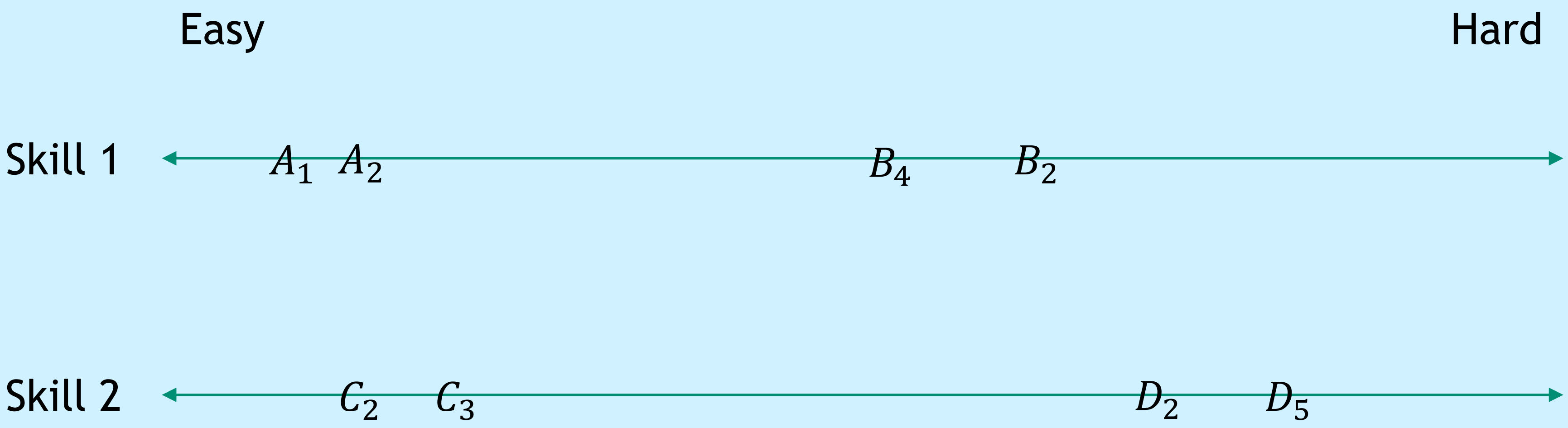
Threat Level	Threat Profile						
	Commitment			Resources			
	Intensity	Stealth	Time	Technical Personnel	Knowledge		Access
					Cyber	Kinetic	
1	H	H	Years to Decades	Hundreds	H	H	H
2	H	H	Years to Decades	Tens of Tens	M	H	M
3	H	H	Months to Years	Tens of Tens	H	M	M
4	M	H	Weeks to Months	Tens	H	M	M
5	H	M	Weeks to Months	Tens	M	M	M
6	M	M	Weeks to Months	Ones	M	M	L
7	M	M	Months to Years	Tens	L	L	L
8	L	L	Days to Weeks	Ones	L	L	L

Adversary Models: Aggregation

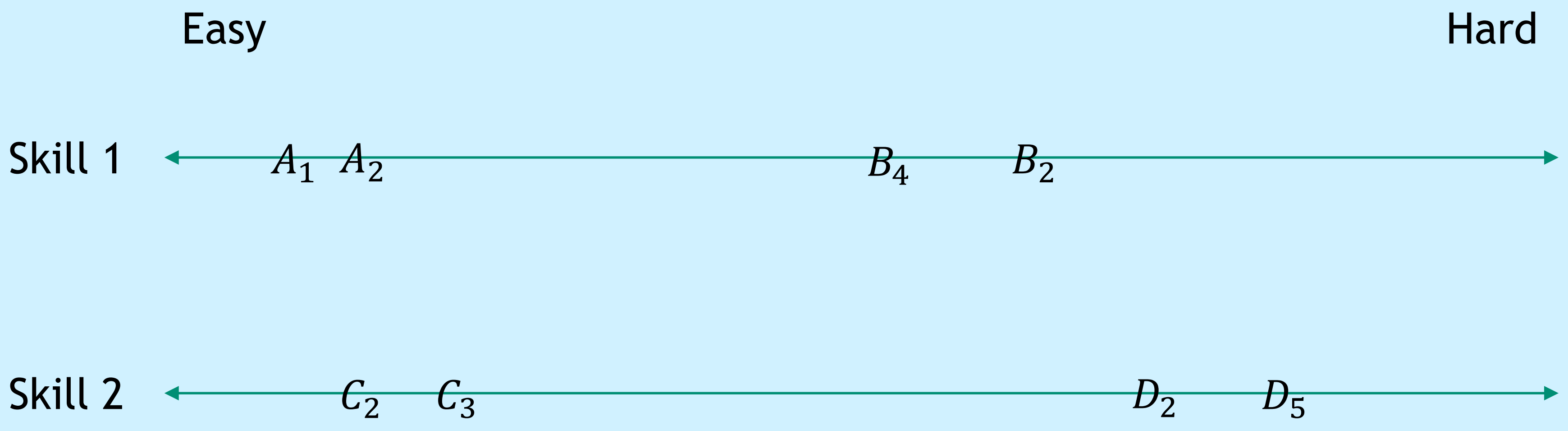
- Why use an aggregate skill level rather than ratings for each attribute?
  - Attribute scores are correlated in real-world adversaries, aggregating ensures hypothetical adversary is more realistic
  - Avoids overfitting adversary characteristics
  - Easier to quickly evaluate attack path difficulties and determine whether the path is feasible to a given adversary
- Disadvantages:
  - Scoring and aggregation can collapse skill sets (e.g., compromising two similar vulnerabilities requires less skill/resources than compromising two very different vulnerabilities)
  - Subjective, may not rely directly on subject matter expertise

Analysis to Select Adversarial Model

- We have been working to make this process less subjective, taking our deliberation out of the process, and relying more on subject matter expertise
- For physical systems, we have surveyed subject matter experts over a variety of dimensions/capabilities to determine what is required to overcome given security measures
- We have taken a couple approaches to using this data
  - Example data: Security measures indexed by letters, experts indexed by subscripts

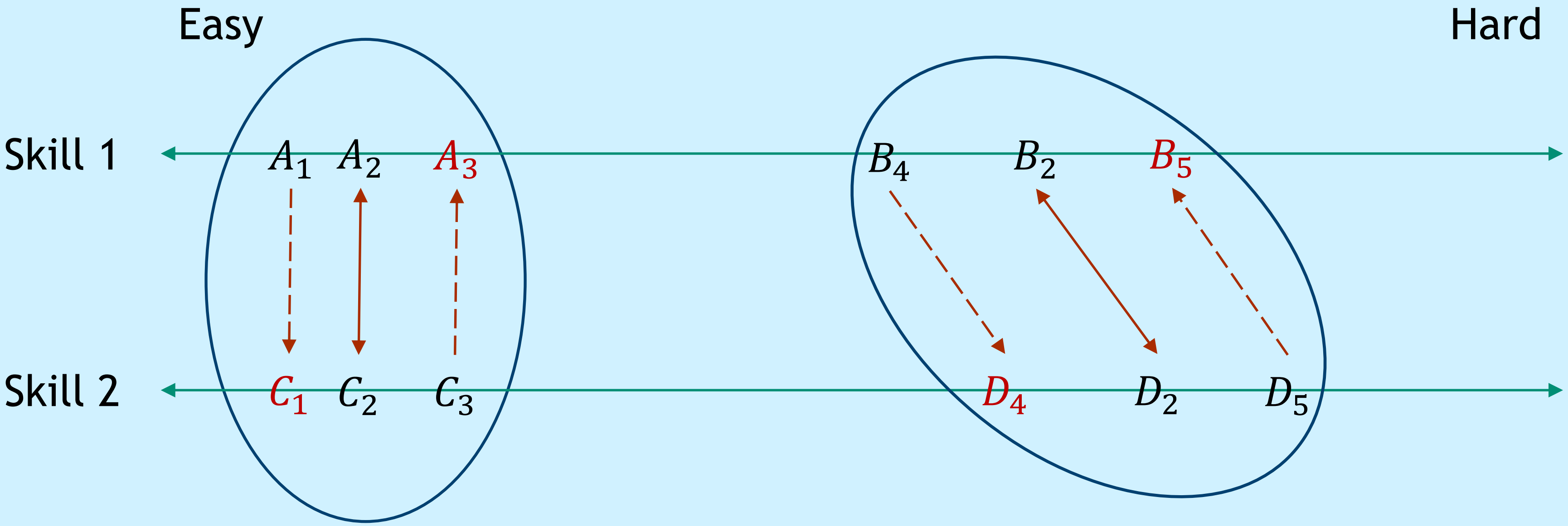


Security Measure Rating Example



- Previous (deliberative) methodology can/has produced aggregate ratings for scenarios (collection of security measures)
  - Classification model can relate security measure ratings (expert) to aggregate ratings (deliberation)
  - Example: $S = (A, C, D)$; $R(S) = f(A_i, C_i, D_i)$
  - Advantages: Resulting predictions are on same scale as past ratings, directly draw on expertise when possible
  - Disadvantage: Resulting predictions are dependent on past (subjective) aggregate ratings

Security Measure Rating Example



- Cross-domain knowledge used to establish relationship between security measures
- Clusters of security measures with similar difficulties (across skills) grouped together to form "difficulty rating"
- Scenarios can be probabilistically assigned to a group
  - Example: $S = (A, C, D)$; $R(S) = (75\%\ Group\ 1, 25\%\ Group\ 2)$
- Method may be specific to the physical systems we are examining

IDART Process

1. Identify consequences of concern (nightmare scenarios)
2. Define a model of a realistic adversary, including skills and tolerance for risk
3. Collect information about the system, including potential weaknesses and attack vectors
4. Identify and document easiest (low effort, low likelihood of detection) attack paths that result in consequences of concern
5. (Optional) Demonstrate attack paths to show feasibility

System Information and Weaknesses

- Approach: Tour facility, gain as much knowledge of the system as possible, document
- Key takeaway: There are frequently differences between how people think a system is configured vs. how it is actually configured
  - Misunderstanding of terms
  - Outdated system information
  - Plan vs. implementation
- Asking the right questions, asking in several different ways
- Examples:
  - "Our system is air-gapped"
    - How do you update software on your system?
    - Can employees check their personal email? How do you read the news?
    - Is the same account information used on internal (air-gapped) and external (not air-gapped) networks?
  - Network diagram vs. network map results

IDART Process

1. Identify consequences of concern (nightmare scenarios)
2. Define a model of a realistic adversary, including skills and tolerance for risk
3. Collect information about the system, including potential weaknesses and attack vectors
4. Identify and document easiest (low effort, low likelihood of detection) attack paths that result in consequences of concern
5. (Optional) Demonstrate attack paths to show feasibility

Attack Paths

- Key assumption: An adversary will not do something difficult or easily detectable if the same result can be accomplished by doing something easy or less detectable
  - Example: Produce/procure identical badge and insert credentials into system OR create a look-alike, dress as staff, and ask someone to let you in
- Basic strategy
  - Find a large open wall and post-its
  - Starting points on one side, nightmare consequences on the other
  - Brainstorm steps at any point in the attack sequence based on ideas/expertise
  - Prune steps that are too difficult for adversary model or more difficult than other steps that accomplish same result
  - Chain together steps to create attack path
- Detailed and complete information about the system is critical
- May require some iteration

IDART Process

1. Identify consequences of concern (nightmare scenarios)
2. Define a model of a realistic adversary, including skills and tolerance for risk
3. Collect information about the system, including potential weaknesses and attack vectors
4. Identify and document easiest (low effort, low likelihood of detection) attack paths that result in consequences of concern
5. (Optional) Demonstrate attack paths to show feasibility

Demonstrating Attack Paths

- Main goal: Show it's possible to make a nightmare consequence happen under the adversarial model
- Showmanship/cool factor never hurts
- Many customers think attacks need to be sophisticated
  - Systems are secured against sophisticated attacks, leave open easier attack paths
  - Showing them an attack can be easy gets them thinking other low-hanging fruit
  - We like to perform assessments in design phase; red-team mentality becomes ingrained in design process
  - We're happy to help identify vulnerabilities, but we like to teach customers to think about potential system weaknesses on their own

Lessons Learned

- Real-life security measures are often not as advanced as expected
  - Attacks do not need to be sophisticated to be successful
  - Security measures tend to focus on one area, leaving open other attack vectors
  - Often not a question of if a system will be compromised, but when; importance of resilience
- Frequent differences between ideas of system configuration and actual configuration
- Quantitative risk analysis can be difficult or infeasible
  - Highlighting vulnerabilities is more approachable and can achieve similar operational goals
  - Other analyses (e.g., resilience analysis) can be useful for other scenarios