Game Theoretic Approaches to Attack Surface Shifting and Reduction

Pratyusa K. Manadhata
HP Labs
manadhata@hp.com
Context: Attack Surface Measurement (ASM)

How can we quantify a software system’s security?

Measure the system’s attack surface [MW10]
Attack Surface Reduction (ASR) Mitigates Risk

Traditional industry approach: code quality improvement

Software will ship with known and future vulnerabilities

Reduce attack surface to increase the difficulty and decrease the impact of future exploitation
Code Quality and ASR Complement Each Other

- **Bad Code Quality**
  - Low Attack Surface Measurement: Low Security Risk
  - High Attack Surface Measurement: Medium Security Risk

- **Good Code Quality**
  - Low Attack Surface Measurement: Medium Security Risk

410/13/2011
ASR in the Industry

- Microsoft
- SAP
- MuSecurity
- OpenSSH
- Firefox
- ...

10/13/2011
Moving Target Defense [GPS09, JGSWW11]

• **Shift** the attack surface
  – “Attacks only work once if at all”
Outline

• Introduce the notion of attack surface reduction

• **Formalize** the notion of attack surface shifting

• Explore *game theoretic approaches* to shift and reduce the attack surface
Intuition Behind Attack Surfaces

A system’s attack surface is the ways in which an adversary can enter the system and potentially cause damage. Hence we define a system’s attack surface in terms of the system’s resources (i.e., methods, channels, and data items).
Model of a System and its Environment

A system, $s$, and its environment, $E_s = \langle U, D, T=\{t_1, t_2\} \rangle$. 
I/O Automata [LT89]

• Action Signature
  – Input, Output, Internal actions
  – Pre and Post conditions
    m.pre and m.post

• Composition
  – \( E_s = (U_{io} \ || \ D_{io} \ || \ ( \ || t_{io} )) \)
  – \( P = s_{io} \ || \ E_s \quad t_{io} \quad T_{io} \)
Not All Resources Are Part of the Attack Surface

- Only those resources that the attacker can use to send data into or receive data from the system are relevant.

- We introduce the formal entry point and exit point framework to identify the relevant resources.
Entry Point and Exit Point Framework

- Entry Points/Exit Points
  - Direct (input/output action)
  - Indirect (internal action)

- Channels (e.g., sockets and pipes)
  - $c \in \text{Res(m.pre)}$

- Untrusted Data Items (e.g., files)
  - $d \in \text{Res(m.post)}, \ d \in \text{Res(m.pre)}$
Attack Surface Definition

• Definition
  - \textbf{M}: set of entry points and exit points
  - \textbf{C}: set of channels
  - \textbf{I}: set of untrusted data items.

\[
\text{attack surface} = \langle M, C, I \rangle
\]
Larger Attack Surface Leads to More Attacks

Attacks \((s)\) = The set of executions of \((s \parallel E_s)\) that contain either an input action or output action of \(s\).

Theorem: Given an environment, \(E\), if \(AS(A) \geq AS(B)\), then \(\text{Attacks}(A \parallel E) \supseteq \text{Attacks}(B \parallel E)\).
Not All Resources Contribute Equally to the Attack Surface

- Contribution $\propto$ Damage Potential

- Contribution $\propto (\text{Attacker Effort})^{-1}$

- Contribution = $\frac{\text{Damage Potential}}{\text{Attacker Effort}}$
Attack Surface Measurement (ASM)

- $\text{ASM}(A) \geq \text{ASM}(B)$ if there exists a nonempty set, $R$, of resources s.t.
  \[ \forall r \in R. \text{contribution}(r, A) \geq \text{contribution}(r, B). \]

Theorem: Given an environment, $E$, if $\text{ASM}(A) \geq \text{ASM}(B)$, then $\text{Attacks}(A \parallel E) \supseteq \text{Attacks}(B \parallel E)$. 

10/13/2011
Quantitative Attack Surface Measurement

• Assume \textit{der}: method $\rightarrow Q$.
  
  – Similarly, for channel and data.

\[
\text{ASM} = \left\langle \sum_{m \in M} \text{der}(m), \sum_{c \in C} \text{der}(c), \sum_{d \in I} \text{der}(d) \right\rangle
\]
## Numeric Damage Potential-Effort Ratio

<table>
<thead>
<tr>
<th>Resource</th>
<th>Damage Potential</th>
<th>Attacker Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>Privilege</td>
<td>Access Rights</td>
</tr>
<tr>
<td>Channel</td>
<td>Protocol</td>
<td>Access Rights</td>
</tr>
<tr>
<td>Data Items</td>
<td>Type</td>
<td>Access Rights</td>
</tr>
</tbody>
</table>

Impose a **total ordering** among the values of the attributes and assign numeric values accordingly, e.g., root = 5 and auth = 3.
Attack Surface Measurement Method

1. Identify a set, $M$, of entry points and exit points, a set, $C$, of channels, and a set, $I$, of untrusted data items.

2. Estimate each relevant resource’s damage potential-effort ratio, $\text{der}$.

3. Compute Attack Surface Measurement $= \left\langle \sum_{m \in M} \text{der}(m), \sum_{c \in C} \text{der}(c), \sum_{d \in I} \text{der}(d) \right\rangle$. 
Shifting the Attack Surface

• Scenario: A system’s defender is trying to protect the system from an attacker.

• Goal: Shift the attack surface such that old attacks don’t work any more
  – may introduce new attacks
Not All Changes Shift the Attack Surface

• Changing the attack surface by changing features
  – Add/remove resources
  – Change existing resource’s contribution

• Shifting the attack surface
  – Remove at least one existing resource
  – Reduce an existing resource’s contribution
Definition of Shifting

- $R_o$: old attack surface
- $R_n$: new attack surface
- $r_o$: a resource, $r$’s, contribution to $R_o$
- $r_n$: $r$’s contribution to $R_n$

\[ \Delta AS = |R_o \setminus R_n| + |\{r: (r \in R_o \cap R_n) \land (r_o > r_n)\}| \]
Shifting Prevents Old Attacks

- Given a system, $S$, an environment, $E$, and $S$’s attack surface, $R$, the set of attacks on $S$ is $\text{Attacks}(S_R \parallel E)$.

**Theorem:** Given an environment, $E$, an old attack surface, $R_o$, a new attack surface, $R_n$, if $\Delta AS > 0$, then $\text{Attacks}(S_{R_o} \parallel E) \setminus \text{Attacks}(S_{R_n} \parallel E) \neq \phi$. 

Old attacks  New attacks
Disable Features: AS Shift and ASM Reduction
Enable and Disable Features: AS Shift and ASM Reduction/Addition/Identical
Enable Features: No AS Shift and ASM Addition
## Summary of Scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Feature</th>
<th>AS Shift</th>
<th>ASM</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Disabled</td>
<td>Yes</td>
<td>Reduction</td>
</tr>
<tr>
<td>B</td>
<td>Enabled and Disabled</td>
<td>Yes</td>
<td>Reduction</td>
</tr>
<tr>
<td>C</td>
<td>Enabled and Disabled</td>
<td>Yes</td>
<td>Identical</td>
</tr>
<tr>
<td>D</td>
<td>Enabled and Disabled</td>
<td>Yes</td>
<td>Addition</td>
</tr>
<tr>
<td>E</td>
<td>Enabled</td>
<td>No</td>
<td>Addition</td>
</tr>
</tbody>
</table>
Scenario choice is a Security-Usability Trade-off

• While shifting the attack surface, which features to disable and which features to enable?
  – More features => more usable system
  – More features => larger attack surface
A Game Theoretic Approach to Moving Target Defense

• Prior work: static software development process
  – No assumptions about the attacker

• Moving target defense is a dynamic scenario
  – Interaction between a defender and an attacker is a game
  – Explicit attacker model
Two-Player Stochastic Game Model [LW02]

• Game = $<S, A^d, A^a, T, Rd, Ra, \beta>$

• $S$: set of states
• $A^*$: action sets
• $T$: $S \times A^d \times A^a \times S \rightarrow [0,1]$: transition function
• $R^*$: $S \times A^d \times A^a \rightarrow \mathbb{R}$: reward functions
• $\beta$: discount factor
Goal: maximize discounted reward.
States, Actions, and Transitions

• State: Feature $\rightarrow$ Configuration

• Action: Feature $\rightarrow$ FeatureAction

• Transition: Specific to a system and its environment
Reward Functions

• $\Delta F$: change in features
• $\Delta AS$: shift in the AS
• $\Delta ASM$: change in the ASM

$R^d: B_1^d (\Delta F) + B_2^d (\Delta AS) - C^d (\Delta ASM)$

$R^a: B^a (\Delta ASM) - C^a (\Delta AS)$
Optimal Defense Strategies

• Model the interaction as an extensive game
  – Complete and perfect information
  – General sum game

• Solution: Equilibrium
Stationary and Dynamic Strategies

• Stationary strategy
  – Independent of history
  – Nash equilibrium
  – Non-linear program for stochastic games [FV96]

• Dynamic strategy
  – Optimal action after every game history
  – Subgame perfect Nash equilibrium
  – Dynamic programming approach [MG07]
Future Work: Instantiate the Model

Challenges in applying the model to real-world systems

• State space explosion
  – Focus on an important set of features

• Transition probabilities

• Reward functions
  – Cost and Benefit functions
Future Work: Model Efficacy

• How much **effort** is necessary to **instantiate** the model?
  – Is the model’s **benefit** worth the effort?

• How does one **compare** alternative game models?

• **Alternative approaches** to achieve moving target defense?
Future Work: Software Development Lifecycle

- Which features to **add** and which to **remove**?
- Prior work: Use the feature’s **contribution to ASM**
A Game Theoretic Approach

• Consider a feature’s “reward” value
  – $B_1^d (\Delta F) + B_2^d (\Delta AS) - C^d (\Delta ASM)$
  – Add features in decreasing order of reward
  – Remove features in increasing order of reward

The simplistic approach ignores feature interaction.
Shapley Value [S53]

- **Coalitional game** \((N, v)\)
  - \(N\): a set of players
  - \(v\): \(2^N \rightarrow \mathbb{R}\) : characteristic function

\[
\Phi_i (v) = \sum_{C \subseteq N \setminus i} \frac{|C|!(|N| - |C| - 1)!}{|N|!} \{v(C \cup \{i\}) - v(C)\}
\]
Choose Features According to their Shapley Value

- **Features** are players in a coalitional game

- Characteristic function: **Reward function**

- Shapley value: A feature’s contribution to security and usability
Related Work

• Moving target defense

• Game theory and security
Summary

• **Formalized** the notion of shifting the attack surface
  – Introduced *game theoretic approaches* to shift and reduce the attack surface

• **A first step** in moving target defense
  – Understanding over time will lead to better approaches
Backup


