Rapid Threat Modeling

Akshay Aggarwal
IOActive Inc.

Outline

• Introduction to Threat Modeling (TM 101)
  – Objectives of TM
  – Suggested process
  – Rapid TM concepts
• Basic Concept Definitions
• Utilizing Software Development Lifecycle (SDLC ) documents
• Case Study
  – DFD
  – Use Case
  – Threat Visualization
  – Attack Tree
Threat Modeling 101

• Allows systematic identification of systemic threats

• What that actually means:
  – Maps out business risks
  – Business threats are derived from business goals
  – Gives holistic view of the security of a system

Objectives of Rapid TM

• Identify architecture and design flaws
• Understand and prioritize risk
• Evaluate effects of system changes
• Mount complex, multistage attacks
• Repeatable, verifiable and consistent model
• Reutilize data generated in Software Development Lifecycle
• High-level picture of system security
• Identify conflicts in policy, requirements and trust
Process

• Suggested process
  – Initial TM during design phase
  – TM enhanced and flushed as technology decisions are taken
  – Threats, risks and mitigations reviewed before implementation
  – TM refined and verified during security review
  – Repeated for next version

The Threat Model is a living document!!

Rapid TM

• Present a quick method to derive and represent threats

• Brief glimpse into ongoing automation of threat generation

• Attack libraries: speed up the process
Definitions

• **Subject:** An actor, usually a human, interacting with the system
• **Object:** An asset that is in the business rules
• **Action:** Something done by subject on object
• **Rule:** Conditions governing a valid action
• **Threat:** Inversion of any rule

---

Definitions

• **Attack:** A process realizing a vulnerability
• **Attack Trees:** Map technical implementation and technology choices to threats
• **Mitigation:** reason why a threat is not realized
• **Weakness:** reason a specific attack succeeds
• **Vulnerability:** an unmitigated path from the leaves of an attack tree to the threat
• **Attack Library:** The IP that speeds up the process by pre-populating known threats against documented technologies/scenarios
It isn’t a threat if it doesn’t affect business

Utilizing SDLC Documentation

- Requirements:
  - Requirements documentation are essential
  - Unambiguous requirements lead to unambiguous rules
  - Examples of unambiguous rules:
    - Subject foo can create object bar
    - Subject X can update object Y if Subject X created object Y

- DFD
  - Describe data flow between processes and data stores

- State Machines:
  - Represent application state representing business logic

- Use Cases:
  - Map state machines to DFD
  - Tend to represent state in Use cases
Case Study: noFUD.org

- Creating noFUD.org ezine:

The editors of this information security magazine (noFUD.org) wish to rid the world of infosec related FUD (or at least die trying). They aim to:
- Attract top security researchers to submit articles
- Maintain tight editorial control to maintain quality
- Allow all Internet users to browse articles free of cost at all times
- Allow users to comment on articles

noFUD.org Rules

- Rules:
  - Author can create Submitted Article
  - Author can update ownSubmitted Article and old version archived
  - Editor can create/delete/update/publish any Submitted Article
  - Editor can delete any Published Article
  - Readers can read all Published Articles
  - Readers can comment on all Published Articles
Use Case for Creating Published Articles

Akshay Aggarwal - Black Hat '05

State Change for submitted article

System

Submitted Article

Published Article

State Change for submitted article

Reader

Author

Editor

Use Case for Creating Published Articles

Acknowledgement: Brenda Larcom

Action Color Key
Allow
Conditional Allow
Disallow

noFUD.org Subject Object Matrix v1.0
Threat: Reader creates/updates/deletes Published Article

Threat: Reader cannot read Published Articles
Threat: Author creates comment as other author

- Editorial note under comment contains the new comment
- Last updated by the author of the comment
- Last updated by the editor of the comment
- Content of the comment
- Last update by the editor of the comment
- Last update by the author of the comment

Updating noFUD.org

- Subscription service added for archives
- Google ads
- Change: Editor cannot publish his own article
- Consequence: Rules added/changed
- Examples:
  - Subscription user can read archived articles
  - Anonymous Reader should always view (read) ads
  - Editor creates Published article if Editor not author of Published Article
Exciting TM work

“Trike Methodology” by Paul Saitta, Brenda Larcom and Michael Eddington

– Threat Modeling methodology
– Partial automation of TM process
– Tool released soon (Hopefully!!)

Conclusion

• Understand the process of developing Rapid TMs
• Reutilize SDLC elements
• Need for unambiguous requirements
• Subject Object Matrix for developing threats
• Visualization of threats
• Speed up the TM process by automated threat generation and Attack Libraries

Contact

Akshay Aggarwal
akshay@noFUD.org