Attack Graph Modeling and Generation

Ratnesh Kumar, Professor, IEEE Fellow
Electrical and Computer Engineering, Iowa State University
PhD Students: Mariam Ibrahim
German Jordanian University
**Attack Graph: Introduction/Motivation**

- **Attack graph**: Visual Data Structure representing all attack scenarios
  - Serves as a basis for risk analysis, defense, detection, and forensics.

- Requires comprehensive overview of
  - System components and interactions with each other and environment
  - Asset Identification and Protections
  - Vulnerability and Threat/Attack identification

- Can be constructed by a state space representation and exploration:
  - Create model $M$ capturing components, behaviors & connectivity, assets/defenses/vulnerabilities/atomic threats, and security property $P$
  - Explore state-space to list paths of $M$ leading to states where $P$ is violated
  - Attack graph is union of all such paths
Formal Approach to Attack Graph: An Example

- Three hosts: 0---attacker’s location; 1, 2---targets.
  - Host 0 monitored by IDS.

- Three services:
  - `ftp` (hosts1 and 2)
  - `sshd` (host1)
  - `database` (host2)

- Four types of vulnerabilities:
  - `fshell`: an executable command shell assigned to `ftp`; exploitable by `ftp_rhosts`
  - `wdir`: a writable `ftp` home directory; exploitable by `ftp_rhosts` and `remote login`
  - `xterm`: executable `xterm` with writable `setuid root` file, exploitable by `local buffer overflow`
  - `ssb`: allows buffer overflow

- Four possible atomic attacks:
  - `ftp_rhosts (ftpr)`: Exploits `fshell` and `wdir`, creating `.rhosts` file in `ftp` home directory for trusting remote login.
  - `local buffer overflow (lbo)`: Exploits `xterm` vulnerability, allowing `root` access.
  - `sshd buffer overflow (sbo)`: Exploits `ssb` vulnerability, allowing `root` shell on victim.
  - `remote login (rlog)`: Exploits `wdir` to override trusts, allowing remote login

- Security property:
  - Attacker’s privilege level on host 2 is below root, or gets detected

1. Hosts \( H = \{0,1,2\} \); variable \( i \in \{0,1,2\} \) (static)
2. Connectivity, \( C \subseteq H \times H \); Boolean \( c_{ij} = 1 \) iff host \( i \) connected to host \( j \) (static)
3. Vulnerabilities \( V = \{\text{wdir}, \text{fshell}, \text{xterm}\} \); Boolean \( v_i = 1 \) iff vulnerability \( v \in \{\text{wdir}, \text{fshell}, \text{xterm}\} \) exists on host \( i \) (static)
4. Attack instances \( A_i \subseteq A \times H \times H \); labeled \( a_{ij} \equiv \text{attack} \) from source \( i \) to target \( j \), \( a \in \{\text{sbo}, \text{ftp}, \text{rlog}, \text{lbo}\} \) (static)
5. Intrusion detection system \( IDS: A \times H \times H \rightarrow \{0,1\} \); Boolean \( ids(a_{ij}) = 1 \) iff attack \( a \) from source \( i \) to target \( j \) is detectable (static).
6. Services \( S = \{\text{ftp}, \text{sshd}, \text{data}\} \); Boolean \( s_i = 1 \) iff service \( s \in \{\text{ftp}, \text{sshd}, \text{data}\} \) is running on host \( i \) (dynamic)
7. Trust relation \( T \subseteq H \times H \); Boolean \( t_{ij} = 1 \) iff \( i \) is trusted by \( j \) (dynamic)
8. Attacker privilege levels \( L = \{\text{none, user, root}\} \); variable \( l_i \in \{\text{none, user, root}\} \) is privilege level on host \( i \) (dynamic)
9. Global detection Boolean \( d_g \) tracks whether IDS alarm triggered for any previously executed atomic attack (dynamic)
10. Attack pre-conditions (for each of 4 attacks):
   - \( Pre(sbo_{ij}) \equiv c_{ij} = 1 \wedge (l_i \geq \text{user}) \wedge (l_j < \text{root}) \wedge (\text{ssh}_d = 1) \)
   - \( Pre(ftp_{rij}) \equiv c_{ij} = 1 \wedge (l_i \geq \text{user}) \wedge (\exists k \in H: t_{kj} = 0) \wedge (ftp_j = 1) \wedge (wdir_j = 1) \wedge (\text{fshell}_j = 1) \)
   - \( Pre(rlog_{ij}) \equiv c_{ij} = 1 \wedge (l_i \geq \text{user}) \wedge (l_j = \text{none}) \wedge (t_{ij} = 1) \)
   - \( Pre(lbo_{ij}) \equiv c_{ij} = 1 \wedge (l_j = \text{user}) \wedge (\text{xterm}_j = 1) \)
11. Attack post-conditions (for each of 4 attacks):
   - \( Post(sbo_{ij}) \equiv (l_j = \text{root}) \wedge (\text{ssh}_d = 0) \wedge ((i = d_g = 0) \Rightarrow (d_g = 0) \vee (d_g = 1)) \)
   - \( Post(ftp_{rij}) \equiv (\forall k \in H: t_{kj} = 1) \)
   - \( Post(rlog_{ij}) \equiv (l_j = \text{user}) \wedge (i = 0 \Rightarrow d_g = 1) \)
   - \( Post(lbo_{ij}) \equiv (l_j = \text{root}) \)
12. Initial States: \( l_0 = \text{root} \wedge (l_1 = l_2 = \text{none}) \wedge (\forall i j \in H \times H: t_{ij} = 0) \wedge (ftp_1 = ftp_2 = \text{ssh}_d = 1) \wedge (data_2 = 1) \wedge d_g = 0. \)
13. Security property \( P \): Attacker on host 2 has privilege level below root or gets detected. In CTL,
   \[ P \equiv AG(l_2 < \text{root} \lor d_g = 1) \equiv AG(\neg (l_2 = \text{root} \land d_g = 0)). \]

- Can be captured formally in AADL and its AGREE Annex.
Formal Model in State-Space, $M$

- $M$ tracks dynamic variables as states:
  - Privilege level
  - Trust relation
  - Services running
  - Detection log

- **Initial state:** Starting values of states

- **Transitions** on atomic attacks whenever pre-conditions hold; and then post-conditions are applied

- **Attack graph** $\equiv$ Subgraph of $M$, containing all paths violating security $P$
Attack Graph from Model $M$ and Property $P$

- Attack graph subgraph of $M$, containing all paths violating $P$
- Information about attack sequence and state evolution that lead to violation of security property
Automated Attack Graph Generation

**AGREE**, a Rockwell tool, developed over **OSATE2**—an extension of Eclipse,

- Supports AADL architectural models, with behaviors and properties captured in AGREE annex, performs translation to Luster for JKind based proving.

**AGREE work flow:**

- .aadl file
- .lus file
- .smt files
- Valid/invalid
- Sat/unsat
- Report results

**OSATE2:** *An eclipse-based tool platform to support AADL and various plug-in tools*
Automated Attack Graph Generation

\[ l_0 = \text{root}, l_1 = \text{root}, l_2 = \text{none}, t_0 = 1, t_1 = 1, t_2 = 1, \]
\[ ftp_1 = 1, sshd_1 = 0, ftp_2 = 1, data_2 = 1, \]
\[ d_g = 0 \]

- Use AADL/AGREE to generate first instance of attack scenario, i.e., counter example:
  \[ CE_1 := sbo_{01} \rightarrow ftp_{02} \rightarrow rlog_{12} \rightarrow lbo_{02} \]
  resulting in the violation of property \( P \).

- Encode this counter example \( CE_1 \) in disjunct with the property \( P \) being checked, namely, \( P \lor CE_1 \).
- Next counterexample: \( \neg(P \lor CE_1) \equiv \neg P \land \neg CE_1 \),
  i.e., a counterexample of \( P \) different from \( CE_1 \).

- This yields a new attack scenario as counterexample,
  \[ CE_2 := sbo_{01} \rightarrow ftp_{12} \rightarrow rlog_{12} \rightarrow lbo_{02} \]
- By repeating this process multiple (but finitely in number) \( CE \)'s can be found, i.e., the entire "attack graph".

• Use AADL/AGREE to generate first instance of attack scenario, i.e., counter example:
  \[ CE_1 := sbo_{01} \rightarrow ftp_{02} \rightarrow rlog_{12} \rightarrow lbo_{02} \]
  resulting in the violation of property \( P \).

- Encode this counter example \( CE_1 \) in disjunct with the property \( P \) being checked, namely, \( P \lor CE_1 \).
- Next counterexample: \( \neg(P \lor CE_1) \equiv \neg P \land \neg CE_1 \),
  i.e., a counterexample of \( P \) different from \( CE_1 \).

- This yields a new attack scenario as counterexample,
  \[ CE_2 := sbo_{01} \rightarrow ftp_{12} \rightarrow rlog_{12} \rightarrow lbo_{02} \]
- By repeating this process multiple (but finitely in number) \( CE \)'s can be found, i.e., the entire "attack graph".
- **AADL description:** 4 subcomponents and their connections
Attacker host 0, with its out ports representing threats it can initiate
“guarantee” specifies the preconditions for the attacks
Target host 1, with its out ports representing attacks it can initiate, and in ports representing attacks it can incur.

Two more out ports are added to trigger the IDS system once an attack instance occurs.
• Complete System, with all out ports representing all possible threats that can be initiated among subsystems: Host 0, Host 1, Host 2.
• Two more out ports for the l_2 attacker level and alarm dg value.
Running AGREE for Complete System

1. System implementation details for Complete_Impl
2. Debugging options in the IDE
3. Property verification results for Complete_Impl
4. Spreadsheet view for contract guarantees
Spreadsheet Counter Example (CE_1) Results

- Each Column represents components ports and local variables at a given time step.
- At each time step/column, one port/threat value is TRUE
- At time step 4, the security property is false, i.e., the dg value remains 0 (un detected), and l_2 is 2 (attacker’s level on H2 is root).

<table>
<thead>
<tr>
<th>Step</th>
<th>security property</th>
<th>TRUE</th>
<th>TRUE</th>
<th>TRUE</th>
<th>TRUE</th>
<th>FALSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>test dg.val</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>test l_2.val</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>thr ftpr 01.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>3</td>
<td>thr ftpr 02.val</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>4</td>
<td>thr ftpr 11.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr ftpr 12.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr ftpr 21.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr ftpr 22.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr lbo 01.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr lbo 02.val</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr lbo 11.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr lbo 12.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr lbo 21.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr lbo 22.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr rlog 01.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr rlog 02.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr rlog 12.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr rlog 21.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr sbo 01.val</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr sbo 02.val</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr sbo 11.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr sbo 12.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>thr sbo 21.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

Complete System ports values & and security property

<table>
<thead>
<tr>
<th>Step</th>
<th>A_sub</th>
<th>A_sub_thr ftpr 01.val</th>
<th>FALSE</th>
<th>FALSE</th>
<th>FALSE</th>
<th>FALSE</th>
<th>FALSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>A_sub_thr ftpr 02.val</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_sub_lbo 01.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_sub_lbo 02.val</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_sub_rlog 01.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_sub_rlog 02.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_sub_sbo 01.val</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_sub_sbo 02.val</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

Host 0 ports values

<table>
<thead>
<tr>
<th>Step</th>
<th>Int det dg</th>
<th>Int det dg ftpr 01.val</th>
<th>FALSE</th>
<th>FALSE</th>
<th>FALSE</th>
<th>FALSE</th>
<th>FALSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Int det dg ftpr 02.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg ftp 11.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg ftp 12.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg lbo 01.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg lbo 02.val</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg lbo 11.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg lbo 12.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg lbo 21.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg lbo 22.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg rlog 01.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg rlog 02.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg rlog 12.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg sbo 01.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg sbo 02.val</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg sbo 11.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg sbo 12.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int det dg sbo 21.val</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

ID ports and dg values

H 1 local variables & ports values

H 2 local variables & ports values
Counter Example (CE_1) Formula within Complete System AGREE Annex

- CE_1 is defined
- The security property is modified to exclude CE_1 from being a next counter example.
At time step 4, the security property is false, i.e., the dg value remains 0 (un detected), and l_2 is 2 (attacker’s level on H2 is root), and the new counterexample is different from CE_1 (FALSE).
Banking System Application Example

- System architecture, assets, controls/defenses, vulnerabilities, threats

**Assets:**
- A1 – User details
- A2 – Oracle database
- A3 – Session identity number
- A4 – User credentials/roles
- A5 – User data in memory
- A6 – DB access credentials
- A7 – Transaction logs
- A8 – Payment details
- A9 – Transaction Results

**Controls/Protections:**
- C1 – Authentication, authorization
- C2 – Websphere hardening
- C3 – TLS 1.2
- C4 – Role-permission based authorization
- C5 – DB security controls
- C6 – Other application security controls (secure code checklist)
- C7 – One Time Password (OTP) for fund transfer

**Vulnerabilities:**
- Using non-validated inputs to generate SQL queries
- Failure to validate input from all sources such as cookies, query string parameters, HTTP headers, databases, or network resources
- Relying on client-side validation
- Failing to audit failed logins
- Failing to secure audit files
- Failing to audit across application tiers
- Revealing too much information to the client
- Using inadequate separation of privileges
- Failing to limit database access to specified stored procedures

**Security property:** Credit card data must not be compromised
Concluding Remarks

- Model-based approach for automated generation of attack scenarios
  - Needs a comprehensive system description (hosts, connectivity, services, vulnerabilities, detection systems, atomic attacks), and security property of interest

- Identify dynamic variables that undergo evolution under attack actions
  - Perform state exploration under sequence of attacking actions to identify all paths of the model that violate the specified security property.

- Use AADL/AGREE tool to demonstrate the approach

- New Directions:
  - Automate generation of ALL counterexample
  - More general model-checker to reason over properties of history of states
  - Case of partial system description
  - Test cases…