

Security

# Security

Fault taxonomy: attacks identified as *malicious faults*

*Malicious faults can be executed with success only if there is a vulnerability in the system*

Dependability in the face of system's vulnerability and attacks

Causes of security violations are different from the causes of failures in hw or sw

Attackers learn over time  
Attackers build a strategy over time

# Security

**Coupling of vulnerability and security exploitation makes security failures different from traditional failures**

- **Vulnerability:**

a computer system vulnerability is a flaw or weakness in a system or network that could be exploited to cause damage, or allow an attacker to manipulate the system in some way

Causes of vulnerabilities may be system components or basic flaws in an individual program or interactions of software programs, ....

- **Exploit**

Exploiting is the means through which a vulnerability can be leveraged for malicious activity (piece of software; sequence of commands, open-source exploit kits, ....)

# Security

Security is defined as resilience to malicious attacks

This can be viewed as  
computer systems failures due to intentional  
attacks

Attackers learn over time  
Attackers build a strategy over  
time

## **Survivability:**

Capability of a system to fulfill its mission in a timely manner, in presence of attacks, failures or accidents

Survivability is related to the ability of the system to perform the intended function

## Reliability in the face of system's vulnerability and malicious attacks

Development of stochastic descriptions of events that may occur during a cyber attack

Probabilities in modelling cyber attacks

Stochastic models for computing measures

# Security

## Availability in the face of system's vulnerability and malicious attacks

Depends on

- attack's own impact on the system
  - effort to diagnose the attack
  - restore system service following the attack
- how long a system remains following a successful attack

## Safety under malicious attacks

- safety depends on the effects of a system failure other than on the causes of failures
- quantification of safety in the context of cyber attacks

Example: Denial of service cyber attack

- Impact of that type of attack on system safety
- The system's attempts to cope with it
  - > we can evaluate the time spent in states that reflect the attack



# Security

Models for security analysis must describe

1. How and when security attacks occur
2. Impact of an attack on the system when it is executed successfully
3. Mechanisms, effects and costs of system recovery, system maintenance and defenses

There are differences with classical dependability

- In the nature and details of security models

Asset: information or resources that could be subject to attack

# Security modelling

## Security threats and vulnerability

- STRIDE Threat Modeling tool
- PLOVER : Preliminary List Of Vulnerability Examples for Researchers

## Quantitative evaluation of security

- Attack trees
- ADversary Vlew Security Evaluation (ADVISE)

# Microsoft Security Development Lifecycle (SDL)

## Threat Modeling tool

The STRIDE threat model provides a way to methodically review system designs and highlight security threats (<https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling>)

STRIDE uses six security threat categories to review system design (developed at Microsoft):

Threat	Desired property
Spoofing	Authenticity
Tampering	Integrity
Repudiation	Non-repudiability
Information disclosure	Confidentiality
Denial of Service	Availability
Elevation of Privilege	Authorization

an adversary exploiting confusion about who is talking

an adversary modifying data

an adversary denying that something happened

disclosure of information to someone not authorized to see it

deny or degrade service to users

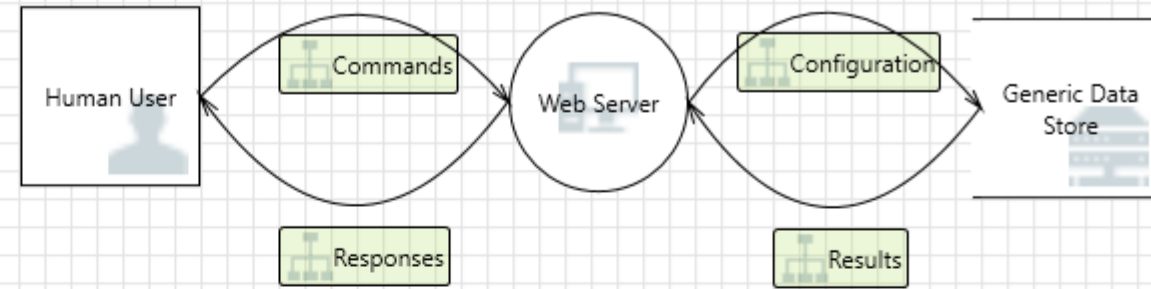
gain capabilities without proper authorization

*Shostack (2014). Threat Modeling: Designing for Security. Wiley.*

# Microsoft Security Development Lifecycle (SDL) Threat Modeling tool

Taken from <https://docs.microsoft.com/it-it/azure/security/develop/threat-modeling-tool-getting-started>

ThreatModelingTool2016



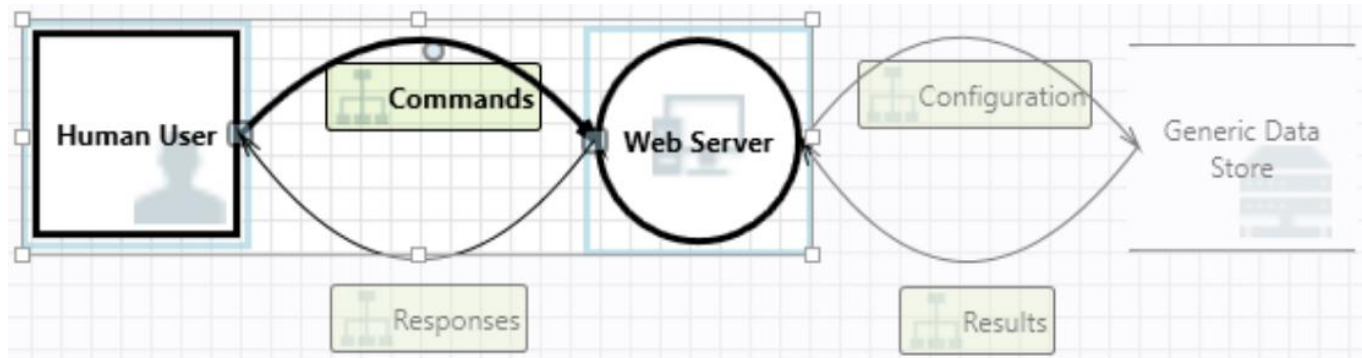
“what  
can go  
wrong in  
this  
system  
we're  
working  
on?”

Diagram	Changed By	Last Modified	State	Title	Category	Description	Justification	Interaction	Priority
Diagram 1		Generated	Not Started	Spoofing the...	Spoofing	Human User...		Commands	High
Diagram 1		Generated	Not Started	Cross Site Scr...	Tampering	The web serv...		Commands	High
Diagram 1		Generated	Not Started	Elevation Usi...	Elevation Of...	Web Server...		Commands	High
Diagram 1		Generated	Not Started	Spoofing of D...	Spoofing	Generic Data...		Configuration	High
Diagram 1		Generated	Not Started	Potential Exc...	Denial Of Ser...	Does Web Se...		Configuration	High
Diagram 1		Generated	Not Started	Spoofing of S...	Spoofing	Generic Data...		Results	High
Diagram 1		Generated	Not Started	Cross Site Scr...	Tampering	The web serv...		Results	High
Diagram 1		Generated	Not Started	Persistent Cr...	Tampering	The web serv...		Results	High
Diagram 1		Generated	Not Started	Weak Access...	Information...	Improper dat...		Results	High

SDL report

# Microsoft Security Development Lifecycle (SDL) Threat Modeling tool

Taken from <https://docs.microsoft.com/it-it/azure/security/develop/threat-modeling-tool-getting-started>



Threat Properties	
ID: 0	Diagram: Diagram 1
Status:	Not Started
Title:	Spoofing the Human User External Entity
Category:	Spoofing
Description:	Human User may be spoofed by an attacker and this may lead to unauthorized access to Web Server. Consider using a standard authentication mechanism to identify the external entity.
Justification:	
Interaction:	Commands
Priority:	High

the tool highlight the importance of adding an authentication mechanism

➤ Diagram - > Identify - > Mitigate - > Validate ->

# Vulnerability classification

Many other approaches classify vulnerabilities and threats that may appear in general in a computer system

CWE: Common Weakness Enumeration <https://cwe.mitre.org/index.html>

*“a community-developed list of software and hardware weakness types. It serves as a common language, a measuring stick for security tools, and as a baseline for weakness identification, mitigation, and prevention efforts”*

PLOVER : Preliminary List Of Vulnerability Examples for Researchers

<https://cwe.mitre.org/documents/sources/PLOVER.pdf>

identifies 28 specific Weaknesses, Idiosyncrasies, Faults and Flaws (WIFFs)

Working document which lists over 1400 real examples of vulnerability (2006)

# PLOVER : Preliminary List Of Vulnerability Examples for Researchers

SECTION.9.10.	[RACE] Race Conditions
SECTION.9.11.	[PPA] Permissions, Privileges, and ACLs
SECTION.9.12.	[HAND] Handler Errors
SECTION.9.13.	[UI] User Interface Errors
SECTION.9.14.	[INT] Interaction Errors
SECTION.9.15.	[INIT] Initialization and Cleanup Errors
SECTION.9.16.	[RES] Resource Management Errors
SECTION.9.17.	[NUM] Numeric Errors
SECTION.9.18.	[AUTHENT] Authentication Error
SECTION.9.19.	[CRYPTO] Cryptographic errors
SECTION.9.20.	[RAND] Randomness and Predictability
SECTION.9.21.	[CODE] Code Evaluation and Injection
SECTION.9.22.	[ERS] Error Conditions, Return Values, Status Codes
SECTION.9.23.	[VER] Insufficient Verification of Data
SECTION.9.24.	[MAID] Modification of Assumed-Immutable Data
SECTION.9.25.	[MAL] Product-Embedded Malicious Code
SECTION.9.26.	[ATTMIT] Common Attack Mitigation Failures
SECTION.9.27.	[CONT] Containment errors (container errors)
SECTION.9.28.	[MISC] Miscellaneous WIFFs

# Preliminary List Of Vulnerability Examples for Researchers

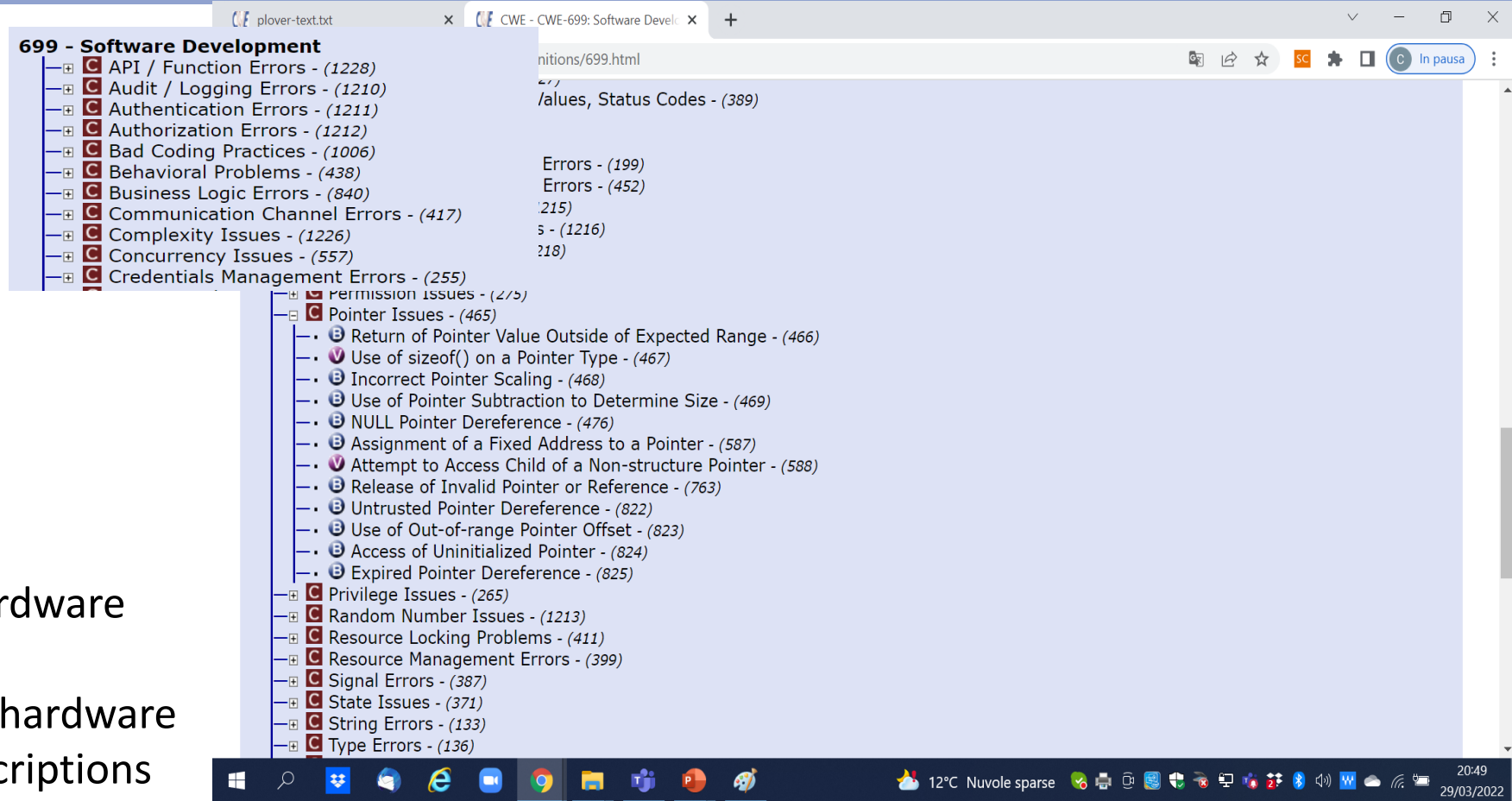
<https://cwe.mitre.org/index.html>

## View CWE

- By software development
- By Hardware design
- By Reserach concepts
- By other criteria

## 2021 CWE Most Important Hardware Weaknesses

“community-developed list of hardware weaknesses with detailed descriptions and authoritative guidance for mitigating and avoiding them”





# Quantitative evaluation of Security

## Combinatorial models

All basic events must be statistically independent

Do not model state - they model operational dependency of the system on the components

Reliability block diagrams: not used in security

Attack trees (similar to Fault Trees)

- Consider a security breach as a system failure
- An attack tree models all possible attacks against the system

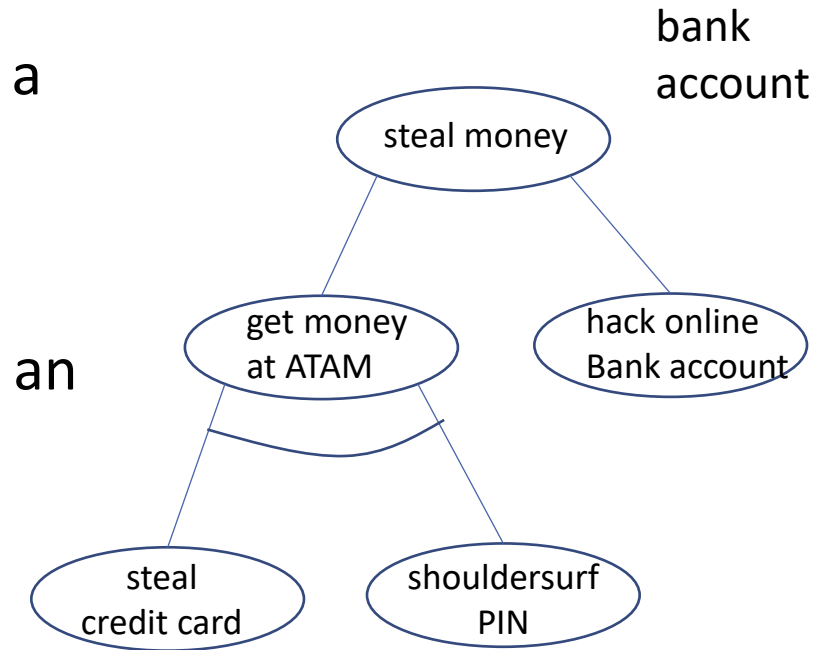
# Attack Trees

The tree describes sets of events that can lead to the goal in a combinatorial way

Security of the system:

set of attack trees, where the root of each tree is the goal of an attacker that can damage the system operation

1. Root = goal of an attacker
2. Leaf nodes = different basic ways to achieve that goal (atomic attacks)
3. OR nodes = a node of which only one of its child nodes needs to be successful
4. AND nodes = a node of which all of its child nodes need to be successful



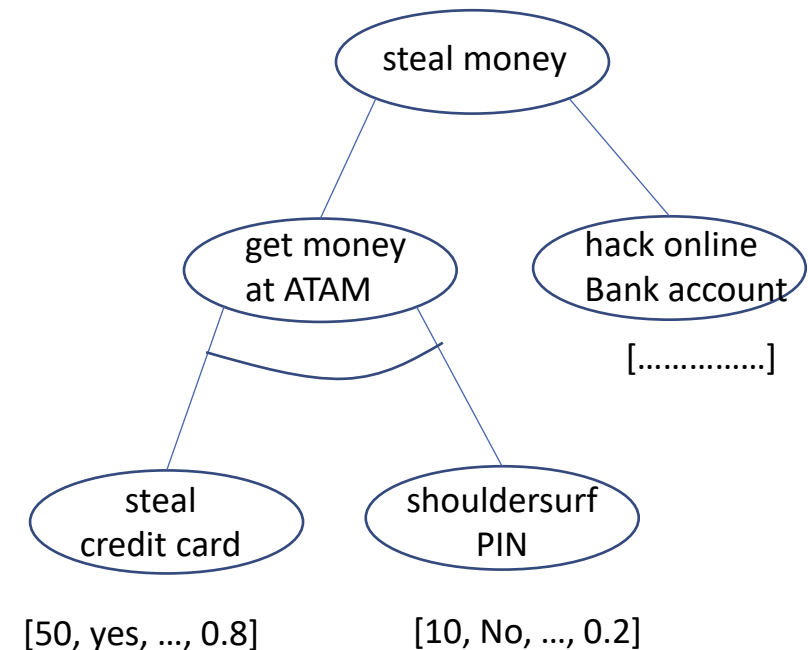
Sjouke Mauw and Martijn Oostdijk,  
Foundations of Attack Trees,  
Information Security and Cryptology,  
ICISC 2005, Lecture Notes in  
Computer Science, vol 3935, 2006

# Attack Trees

- leaf nodes represent atomic attacks
- atomic attacks are assigned attributes

In the example: [Cost, Special Equipment, ..., Probability]

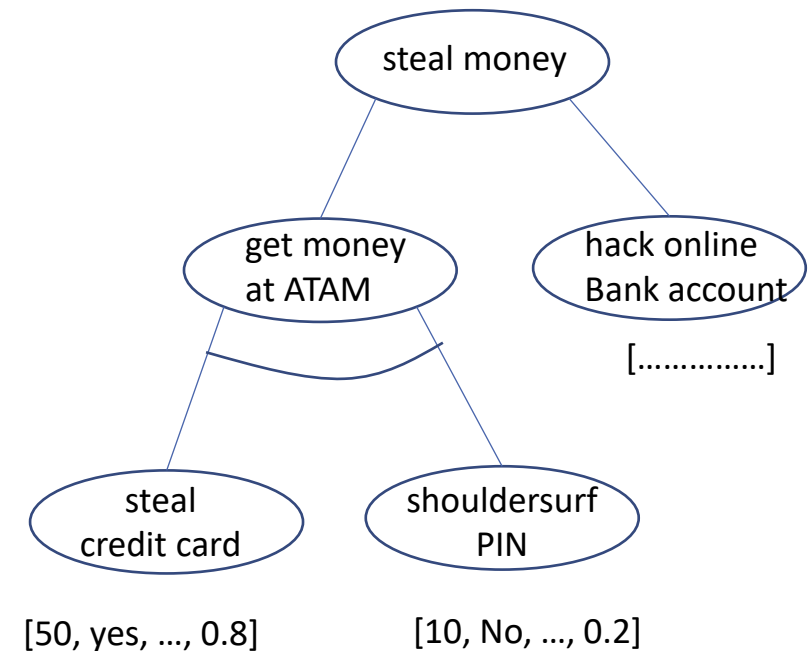
- The result of an analysis can be the value of an attribute in the root node (for example the cost of the cheapest attack)
- Values of different attributes can be defined
- Those attacks costing less than 100K Euro and that do not require use of special equipment.



# Attack Trees: system vulnerability analysis

The attack tree can be used to combine the values of these attributes and help users to learn more about a system's vulnerability

assign values to leaf nodes and propagate the node value up to the root



# Evaluation of Security

Minimum cut-set -> set of atomic attacks that achieve a goal

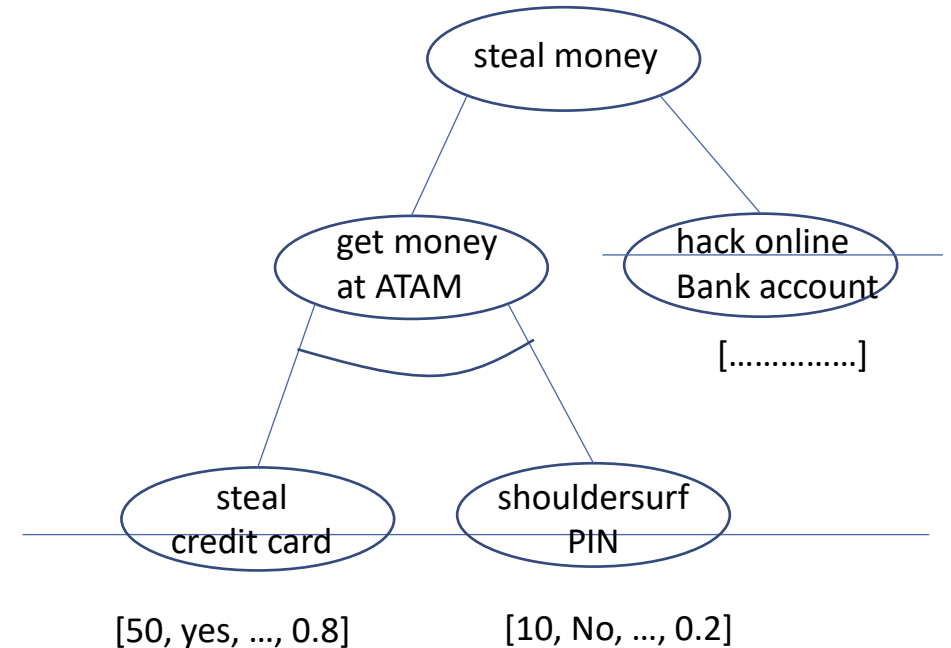
$S = \{\{\text{Steal credit card, Shouldersurf PIN}\}$   
 $\{\text{Hack online Bank account}\}\}$

$\{\text{Hack online Bank account}\}$  is sufficient for the success of the attack

Impact of certain atomic attacks on the overall system security

Attack Trees: systematic ways to describe system vulnerability , making possible to assess risks and making security decisions

Attack trees: reusable as part of a larger attack tree for a system



# Models for security analysis

Previous models:

- do not capture the dependence of security vulnerability and attacks as well as sequences of attacks steps

Stochastic assumptions are needed to describe systems that have yet to be built and for systems whose set of vulnerability is unknown.



state-based stochastic methods application in security context

# Models for security analysis

These models must describe

1. *How and when security attacks occur*
2. *Impact of an attack on the system when it is executed successfully*
3. *Mechanisms, effects and costs of system recovery, system maintenance and defenses*

There are differences with classical dependability

- In the nature and details of security models

*Asset*: information or resources that could be subject to attack

# ADversary View Security Evaluation - ADVISE

These set of slides are based on the paper:

E. LeMay, M. D. Ford, K. Keefe, W. H. Sanders and C. Muehrcke, "Model-based Security Metrics Using ADversary Vlew Security Evaluation (ADVISE)," *2011 Eighth International Conference on Quantitative Evaluation of SysTems*, Aachen, 2011, pp. 191-200.



# ADversary View Security Evaluation - ADVISE

Main objective:

- Compare security strenght of different system architectures
- Analyse threats by different adversaries



Executable state-based security model system

1. A system
2. An adversary view (how the adversary can attack the system)
3. Security metrics

An *attack* is specified in terms of many small attack steps.

Specification of an *Attack Execution Graph* (AEG)

*Attack decision function*  
how the adversary selects the most attractive next attack step

# Attack Execution Graph - AEG

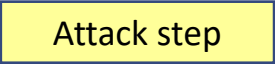
Attack execution graph (AEG)

$\langle A, R, K, S, G \rangle$

Mobius tool

<https://www.mobius.illinois.edu/>

A: set of attack steps




Attack step

R: set of access domains in the system



Access

K: set of knowledge items relevant to attack the system



Know  
ledge

S: set of the adversary attack skills



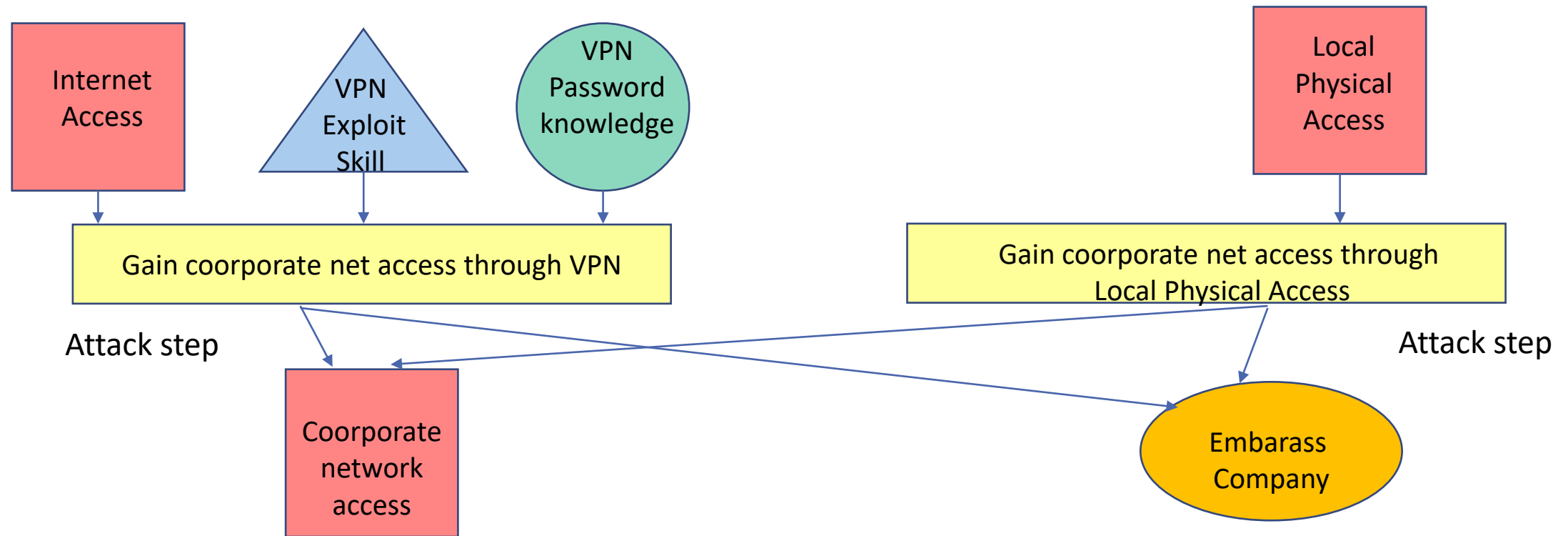
Skill

G: set of adversary attack goals relevant to the system



Goal

# ADVISE: AEG



Example of AEG taken from paper

E. LeMay, M. D. Ford, K. Keefe, W. H. Sanders and C. Muehrcke, "Model-based Security Metrics Using ADversary Vlew Security Evaluation (ADVISE)," *2011 Eighth International Conference on Quantitative Evaluation of SysTems*, Aachen, 2011, pp. 191-200.

# Attack Step

$$a_i = \langle B_i, T_i, C_i, O_i, Pr_i, D_i, E_i \rangle$$

X is defined as the set of all reachable model states

$$X = \{s_1, \dots, s_n\}$$

$$B_i: X \rightarrow \{\text{true}, \text{false}\}$$

precondition to check if the attack is enabled

The adversary has the access, the knowledge, and/or skill needed for the attack and the adversary does not have what can be gained when the attack is executed with success

$$T_i: X \times \mathbb{R}^+ \rightarrow [0, 1]$$

time required to execute the attack.

$T_i(s)$  is a random variable defined over a prob. distribution function

$$C_i: X \rightarrow \mathbb{R}^{\geq 0}$$

cost of attempting the attack

$$O_i:$$

finite set of outcomes (e.g., success and failure)

$$Pr_i: X \times O \rightarrow [0, 1]$$

prob. of outcome o after the attack

$$(\sum_o Pr(s, o) = 1)$$

$$D_i: X \times O \rightarrow [0, 1]$$

probability that the attack is detected when outcome o occurs

$$E_i: X \times O \rightarrow X$$

next state when the outcome o occurs

# Attack Step do-nothing

$a_{DN} = \text{do-nothing}$

$B_{DN}$   
precondition is always true

$T_{DN}$   
time between two occurrences  
of do nothing

$C_{DN}$   
cost is zero

$D_{DN}$   
detectability is zero

$E_{DN}(s, o) = s$   
the next state is the same of the current state

$\Pr_{DN}(s, o) = 1$   
there is only one outcome, with probability 1

Every AEG contains the  $a_{DN}$  attack step



there is always at least one attack step in the AEG whose precondition is satisfied

# Model state $s$

A state  $s$  in  $X$  reflects the progress of the adversary in attacking the system

$$s = \langle R_s, K_s, G_s \rangle$$

$R_s$  : set of domains that the adversary can access

$K_s$  : set of knowledge of the adversary

$G_s$  : set of attack goals achieved by the adversary

# Adversary Profile definition

Adversary Profile =  $\langle s_0, L, V, w_C, w_P, w_D, U_C, U_P, U_D, N \rangle$

$s_0$ : initial state of the model

$L$ : attack skill level function

$V$ : attack goal value function

$w_C, w_P, w_D$  : weights for preferences: weight for cost, payoff, detection probability

$U_C, U_P, U_D$ : utility functions for cost, payoff, detection probability

$N$ : planning horizon

# Adversary Profile definition

Adversary Profile =  $\langle s_0, L, V, w_C, w_P, w_D, U_C, U_P, U_D, N \rangle$

$s_0$ : starting point of the adversary attack

different for insider (more access and knowledge) and outsider adversary

$L$  is the attack skill level function

$L : S \rightarrow [0, 1]$  maps each attack skill to a value in  $[0, 1]$  (proficiency of the adversary)

$V$  is the attack goal value function

$V: G \rightarrow \mathbb{R}^{>=0}$ , monetary value of each attack goal in the AEG from the adversary viewpoint, more valuable  $\rightarrow$  larger value

Payoff value  $P(s)$  of a state  $s$  is a function of the value of all goals  $V(g)$  achieved in the model state  $P(s) = f(V(g))$



# Adversary Profile definition

Adversary Profile =  $\langle s_0, L, V, w_C, w_P, w_D, U_C, U_P, U_D, N \rangle$

Attack preference weight: attractiveness in each of the three criteria when deciding an attack. They are a value in  $[0,1]$

$w_C$ : relative attractiveness of decreasing the cost in attempting the attack step

$w_P$ : relative attractiveness of increasing the payoff for successfully executing the attack step

$w_D$ : relative attractiveness of decreasing the probability of being detected during or after the attack

# Adversary Profile definition

Adversary Profile =  $\langle s_0, L, V, w_C, w_P, w_D, U_C, U_P, U_D, N \rangle$

Utility functions: map the native value of each attractiveness criterion to a  $[0, 1]$  utility scale (higher utility values represent more desirable values)

$U_C: R^{\geq 0} \rightarrow [0, 1]$  map the monetary value of the attack step cost to a  $[0, 1]$   
lower cost - higher utility value

$U_P: R^{\geq 0} \rightarrow [0, 1]$  map the monetary value of the attack step payoff to a  $[0, 1]$   
higher payoff - higher utility value

$U_D: [0, 1] \rightarrow [0, 1]$  map the probability of attack step detection to a  $[0, 1]$   
lower detection probability - higher utility value

# ADVISE model: execution

$A_s$  is the set of available attack steps  $a_i$  in state  $s$ :  
the attack steps whose precondition is satisfied ( $B_i(s)=\text{True}$ )

The attractiveness of the all available attack steps is evaluated from the viewpoint of the adversary with the criteria

- Cost
- Detectability
- Expected payoff in the next state

The *attack decision function* chooses the next attack step

The attack step outcome determines the next state (the outcome is stochastic)

The process is repeated

# ADVISE model: attack decision function

Short sighted adversary attack decision function

$$\text{attr}(a_i, s) = w_C C_i(s) + w_P P_i(s) + w_D D_i(s)$$

linear combination of adversary preferences weights with the data about attack step

$$P_i(s) = \sum_o (P(E_i(s,o)) \cdot \text{Pr}_i(s, o))$$

↓  
expected payoff

↓ Payoff in the next state  
reached by outcome o ( $E_i(s,o)$ )

$$D_i(s) = \sum_o (D_i(s,o) \cdot \text{Pr}_i(s, o))$$

$\beta(s)$  best next attack step

$$\{a^* \text{ in } A_s \mid \text{attr}(a^*, s) = \max \{ \text{attr}(a_i, s) \text{ for all } a_i \text{ in } A_s \} \}$$

one of the maximally attractive attack steps is chosen uniformly

# ADVISE model: execution

Utility function  $U_C$   $U_P$   $U_D$  are not shown in  $\text{attr}(a_i, s)$  for simplicity  
They should be applied to move towards a common unit of utility.

$C_i(s) \text{ ---- } U_C(C_i(s))$

$P_i(s) \text{ ---- } U_P(P_i(s))$

$D_i(s) \text{ ---- } U_D(D_i(s))$

$C_i(s) = 2.01 \text{ million}$

$C_i(s') = 2.05 \text{ million}$

Better mapped  $\rightarrow$  same  
utility value

$C_i(s) = 10.000$

$C_i(s') = 50.000$

better mapped  $\rightarrow$  two  
distinct utility values

An attack step outcome is randomly generated using the probabilities distributions

The attack step outcomes determine the sequence of state transitions

# ADVISE execution algorithm

## ADVISE model execution algorithm

---

```
Time <- 0
State <-  $s_0$ 
while Time <  $\tau$  do
    Attacki <-  $\beta(\text{State})$ 
    Outcome <- o, ----- o, Probi(State)
    Time <- Time + t, ----- t, Ti(State)
    State <- Ei (State, Outcome) ----- Ei, next state
function
end while
```

---

# ADVISE metrics specification

*State metrics*  $\langle \tau, \lambda, \sigma \rangle$

$\tau$  is the end time  $[0, \tau]$

$\lambda$  is the type of state metrics :

**EndProb**: probability of being in state  $s$  at time  $\tau$  with  $\sigma(s)=\text{True}$

**AvgTime** : average amount of time spent in state  $s$  such that  $\sigma(s)=\text{True}$   
in the interval  $[0, \tau]$

$\sigma$  is the state indicator function:  $s = \langle R, K, G \rangle$

$\sigma(s)$  returns True, for states of interest

e.g.,  $\sigma(s) = \text{true}$  if goal  $g1$  has been achieved

# ADVISE metrics specification

*Event metrics*  $\langle \tau, \delta, \varepsilon \rangle$

$\tau$  is the end time  $[0, \tau]$

$\delta$  is the type of event metrics : let  $\varepsilon$  a set of events

**Freq**: number of occurrences of events in  $\varepsilon$  in the interval  $[0, \tau]$

**ProbOcc** : prob. that all the events in  $\varepsilon$  occur at least once in the interval  $[0, \tau]$

$\varepsilon$  is a set of events in the model

(attack steps, attack step outcomes, access domains, knowledge or goals)

Example

Frequency of attack step  $a_i$  in the interval  $[0, \tau]$

$\varepsilon$  is equal to  $\{a_i\}$



# ADVISE model

In the paper:

- more sophisticated adversary decision with a long range planning attack decision function are shown (State Look-Ahead Tree)
- A case study on a SCADA (Supervisory Control and Data Acquisition) architecture is analysed: 2 variants of the architecture and 4 different profiles of adversaries.

# Attacks to a back-end server for autonomous vehicles

**UN Regulation No. 155 - Addendum 154**  
**Uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system**

**Annex 5** List of threats and corresponding mitigations

**Part A. Vulnerability or attack method related to the threats**

**Table A1 List of vulnerability or attack method related to the threats**

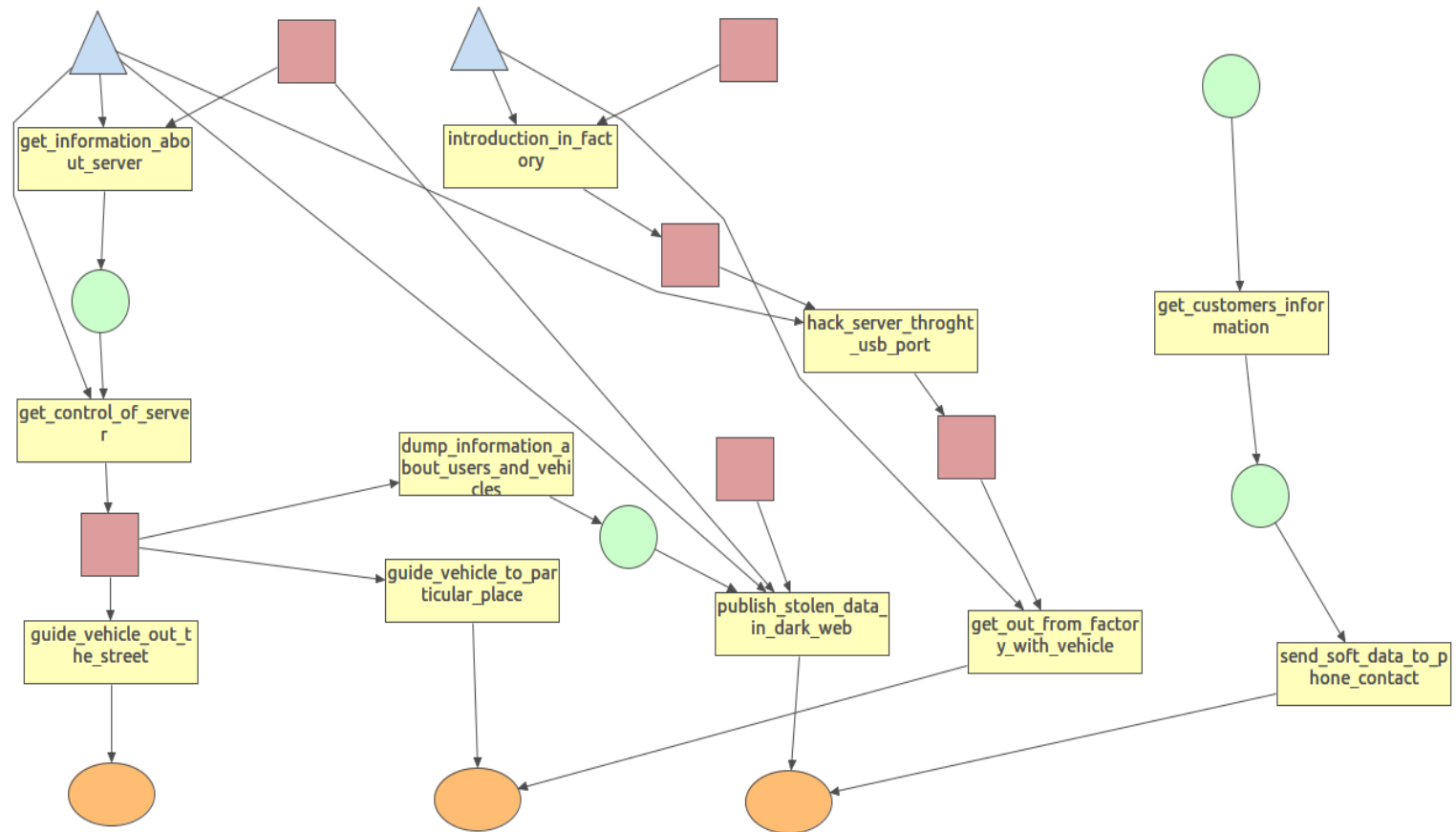
<i>High level and sub-level descriptions of vulnerability/ threat</i>			<i>Example of vulnerability or attack method</i>	
4.3.1 Threats regarding back-end servers related to vehicles in the field	1	Back-end servers used as a means to attack a vehicle or extract data	1.1	Abuse of privileges by staff (insider attack)
			1.2	Unauthorized internet access to the server (enabled for example by backdoors, unpatched system software vulnerabilities, SQL attacks or other means)
			1.3	Unauthorized physical access to the server (conducted by for example USB sticks or other media connecting to the server)
	2	Services from back-end server being disrupted, affecting the operation of a vehicle	2.1	Attack on back-end server stops it functioning, for example it prevents it from interacting with vehicles and providing services they rely on
	3	Vehicle related data held on back-end servers being lost or compromised ("data breach")	3.1	Abuse of privileges by staff (insider attack)
			3.2	Loss of information in the cloud. Sensitive data may be lost due to attacks or accidents when data is stored by third-party cloud service providers
			3.3	Unauthorized internet access to the server (enabled for example by backdoors, unpatched system software vulnerabilities, SQL attacks or other means)
			3.4	Unauthorized physical access to the server (conducted for example by USB sticks or other media connecting to the server)
			3.5	Information breach by unintended sharing of data (e.g. admin errors)

# Exercise

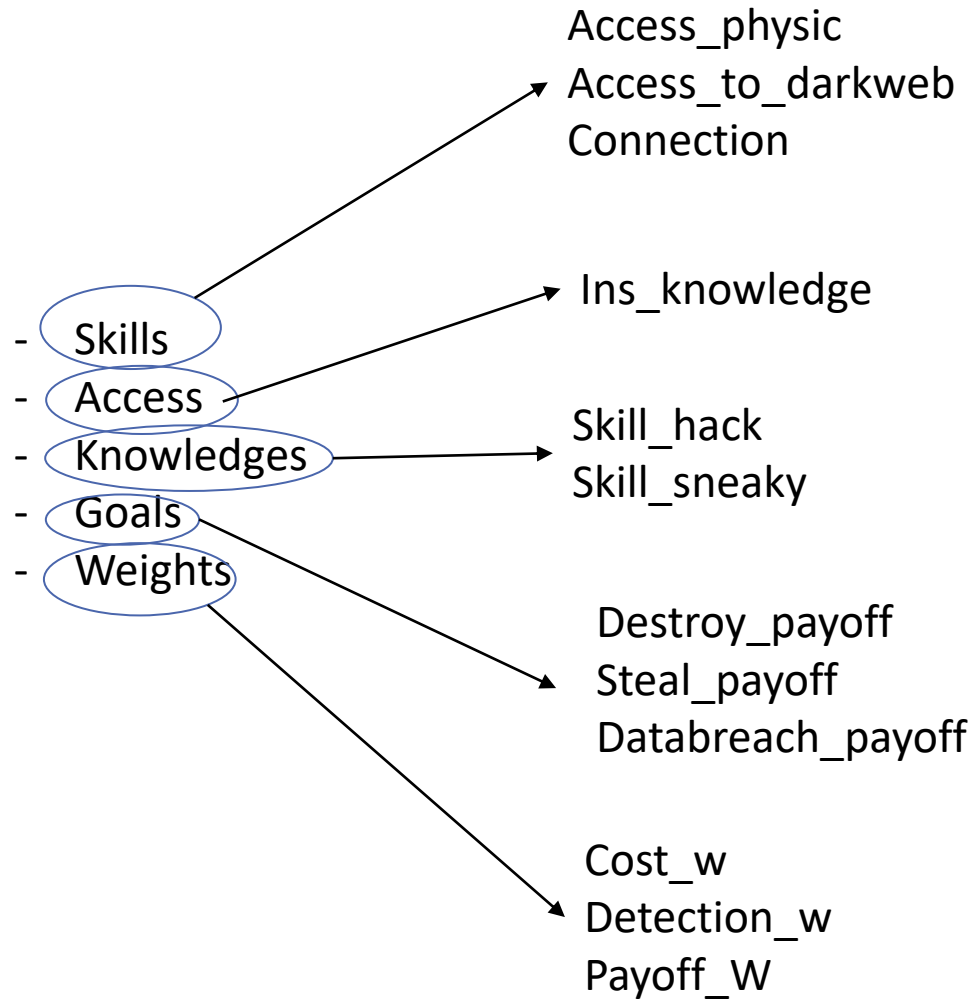
Create an ADVISE model for different Adversaries trying to attack the back-end servers related to autonomous vehicles, including the threats suggested in the next table.

- 1. Create an attack tree with roughly 20 elements ( knowledge, access, skill, attack step and goal).
- 2. Create 3 different adversary profiles 1. Insider 2. Hacker 3. Physical Intruder
- 3. Evaluate the probability of each adversary to achieve the goals.
- 4. Analyze the results.

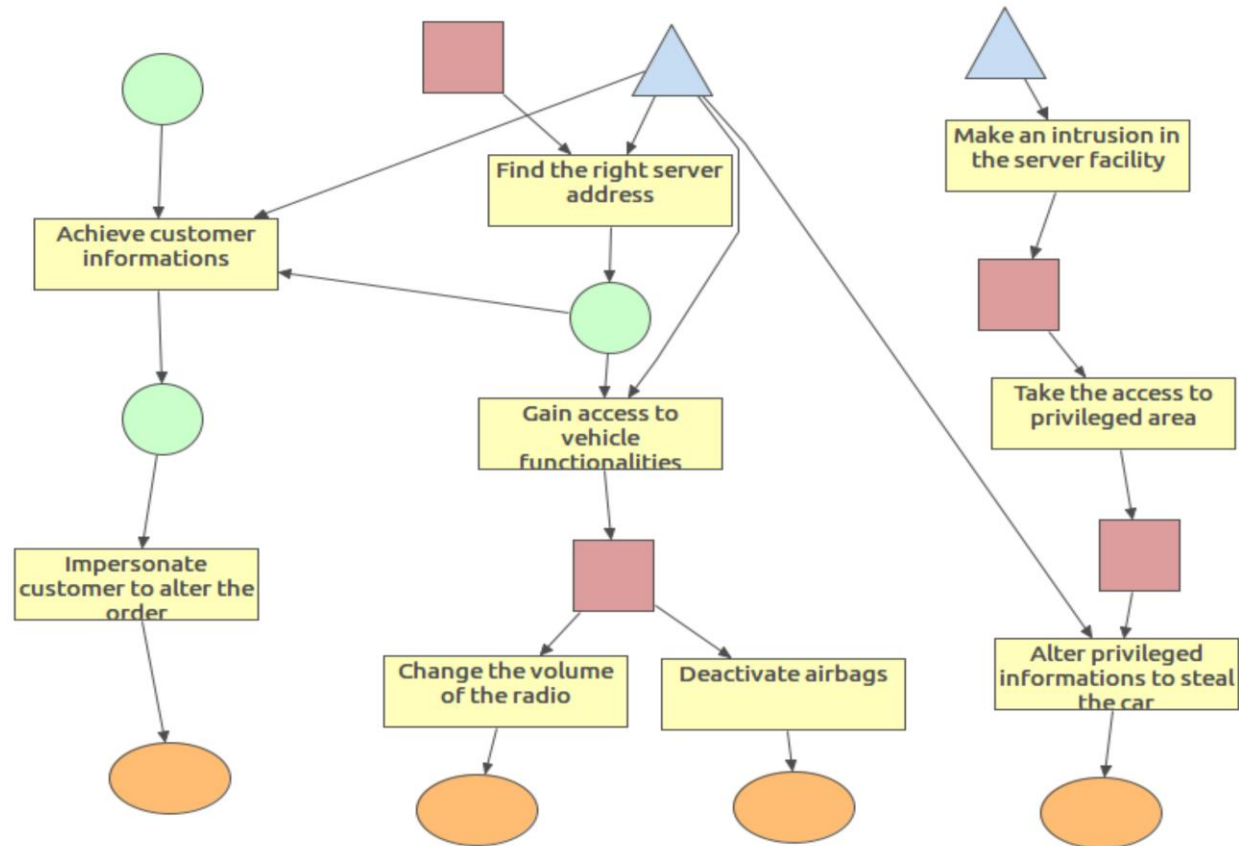
# An ADVISE model



# An ADVISE model



# Another ADVISE model



# Another ADVISE model

## Goals:

- Alter the order
- Annoy the user
- Harm the user
- Steal the car