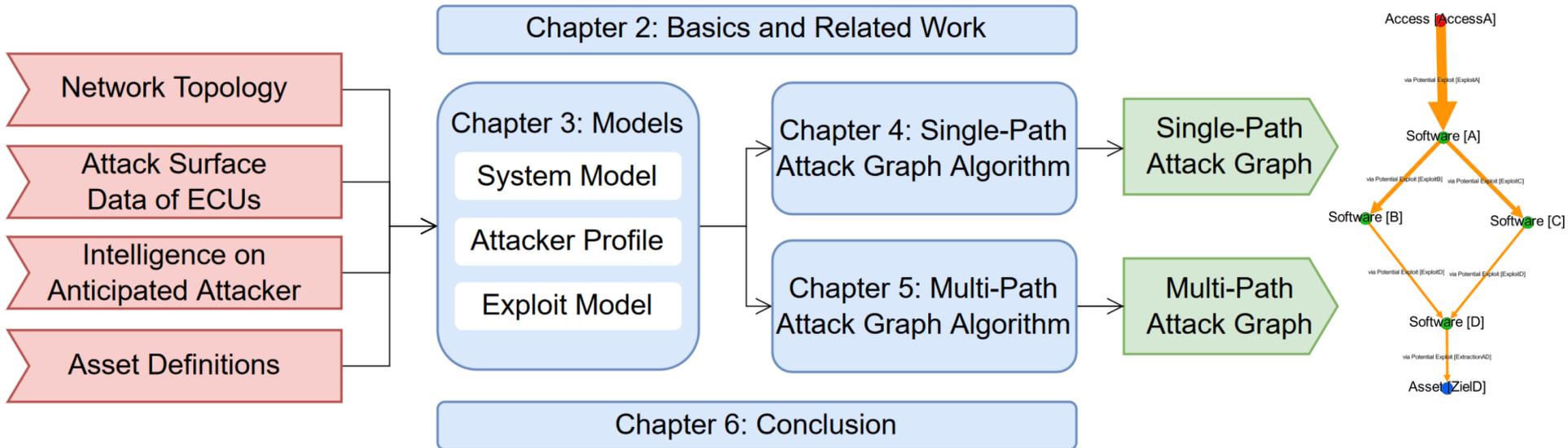


Automotive Security Analyzer for Exploitability Risks

An Automated and Attack Graph-Based Evaluation of On-Board Networks



Questions and Disclaimer

Feel free to ask questions at any time, even during the talk. I will repeat them for the recording.

- Short questions during the talk.
- Normal questions during the Q&A, or afterwards.

Disclaimer

- Personal opinions only - not employers' views.
- Automotive Security Analyzer for Exploitability Risks Proof of Concept approved, but surrounding tools/processes NOT.
- Informational purposes only - no liability.
- Third-party IP remains with owners.

> whoami

Academia:

- TUM: PhD in “**Automotive Security Analyzer for Exploitability Risks: An Automated and Attack Graph-Based Evaluation of On-Board Networks**”.
- National Institute of Informatics in Tokio, Japan on content security.

Industry:

- Research and Development for **On-Board Networks 2008-2015**.
- In IT Security full time for 14 years.

Study:

- TUM/LMU/UniA: M. Sc. with honours in Software Engineering.
- HM/KPU: B.Sc. in Computer Science.

Military: Electronic Warfare Battalion 932 in Frankenberg/Eder.



Agenda

1. Motivation

- Problem: Larger Attack Surface and More Attacks
- Vision: Attack Graphs with Risk Annotations
- C1 Survey

2. Data and Models

- C2 System Model, Attacker Profile, a. Exploit M.
- C3 Attack Surface Exploitability Quantification
- C4 Implementation and Evaluation of Models

3. Practical Demo

4. Algorithms for Attack Graphs

- C5 Single-Path Attack Graph Algorithm (PI+PII)
- C6 Implementation and Evaluation of PI+PII

5. Algorithms for Total Risk

- C7 Probabilistic Model
- C8 Multi-Path Attack Graph Algorithm (P3Salfer)
- C9 Bayes Network Unsuitability Finding
- C10 Design and Implementation of an Alternative Algorithm with Bayesian Networks (P3Bayes)
- C11 Implementation and Evaluation of the Multi-Path Attack Graph Algorithm (P3Salfer)

6. Future Work

7. Further Material Overview

Larger Attack Surface and Assets Raise the Protection Level

Vehicles connect to the whole world.

Vehicles carry our credentials and our lives.



C1: Survey: Tools and Standards Boost Hacking



Some hacking examples:

- **Chevrolet Impalla:** Koscher et al. @ S&P 2010 and Checkoway et al. @ Usenix Security 2011;
- **Keyless Entry:** Rouf et al. @ Usenix Sec. 2010 and Verdult/Garcia/Balsch @ Usenix Sec. 2012;
- **Jeep:** Miller/Valasek @ BlackHat 2015.
- **Tesla Model S:** Rogers/Mahaffey @ Def Con 23 and Keen Labs in 2016 and @ Blackhat '17;
- **Public Charging:** see Dalheimer @ 34C3.
- **Tesla's Passive Entry system:** See Herfurt @ Troopers22

Stakeholder to Support on Potential Security Risks

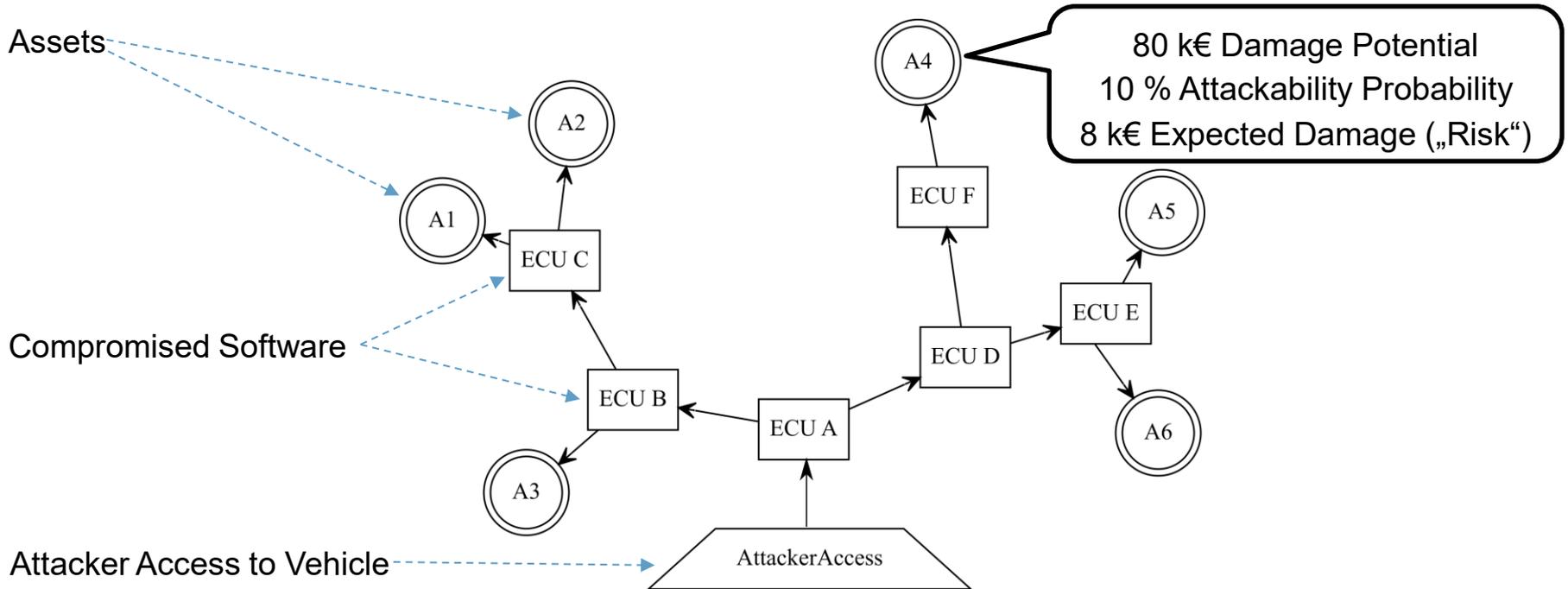
Architects: „How bad would this affect security?“



Penetration Tester: “Where to look at first?“



Vision: Attack Graphs with Risk Annotations



C1: Survey: From Safety Analysis to Security Automation

1980s: Safety Precursors

FTA (Fault Tree Analysis), FMEA (Failure Modes and Effects Analysis).

1990s: Attack Trees

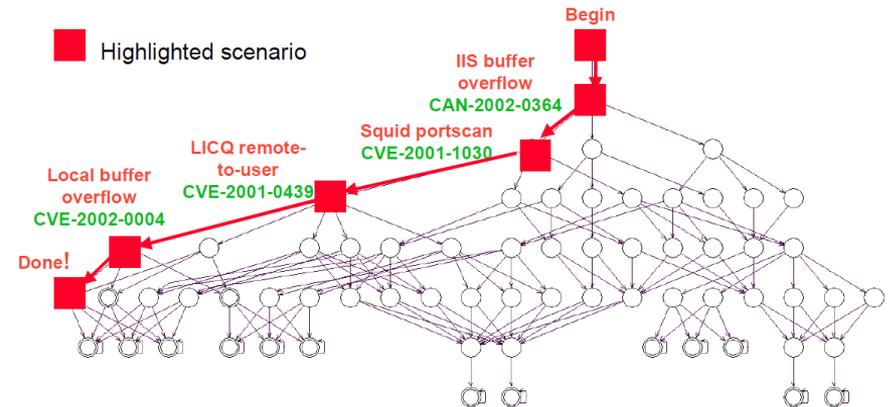
E. Amoroso(Bell Labs), Salter (NSA), Schneier.

2000s: Automated Construction

Sheyner et al. und Ou et al.

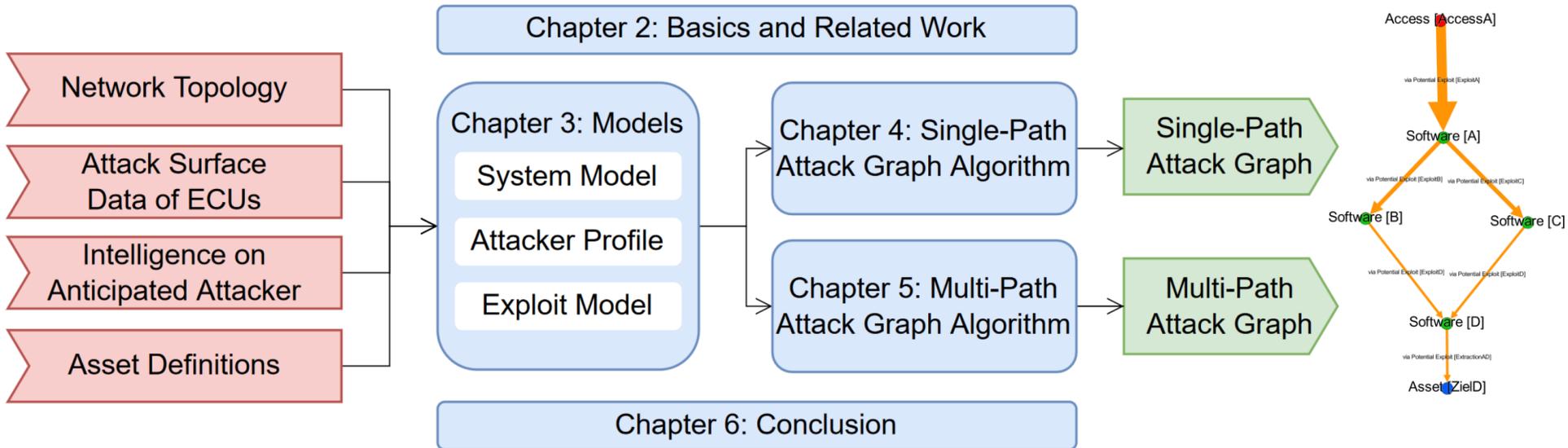
2010s: Automated Sourcing and Effects

Roschke et al. (HPI-VDB and IDS).

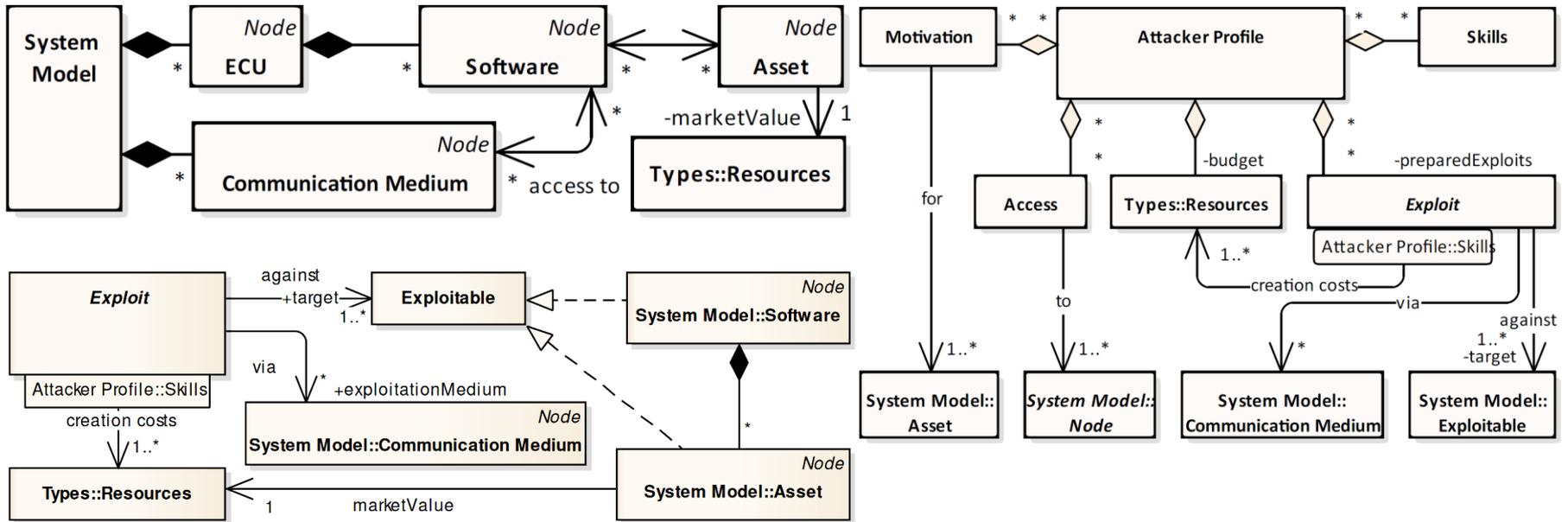


Automotive Security Analyzer for Exploitability Risks (AutoSAIfER)

An Automated and Attack Graph-Based Evaluation of On-Board Networks



C2: System Model, Attacker Profile, and Exploit Model



C3: Attack Surface Exploitability Quantification

Possibility for a vulnerability:

$$P_B(X > 0) = 1 - P_B(X = 0) = 1 - q^i = 1 - (1 - v)^i$$

Total effort estimation:

$$\bar{\mathcal{C}}_E = s_{sf0} + s_{sfi} \left(s_{sv} \sum_{i=1}^{i=k} \left(i (1 - s_{sv})^{i-1} \right) + k (1 - s_{sv})^k \right) + s_{sa} + s_{sb} + s_{sc}$$

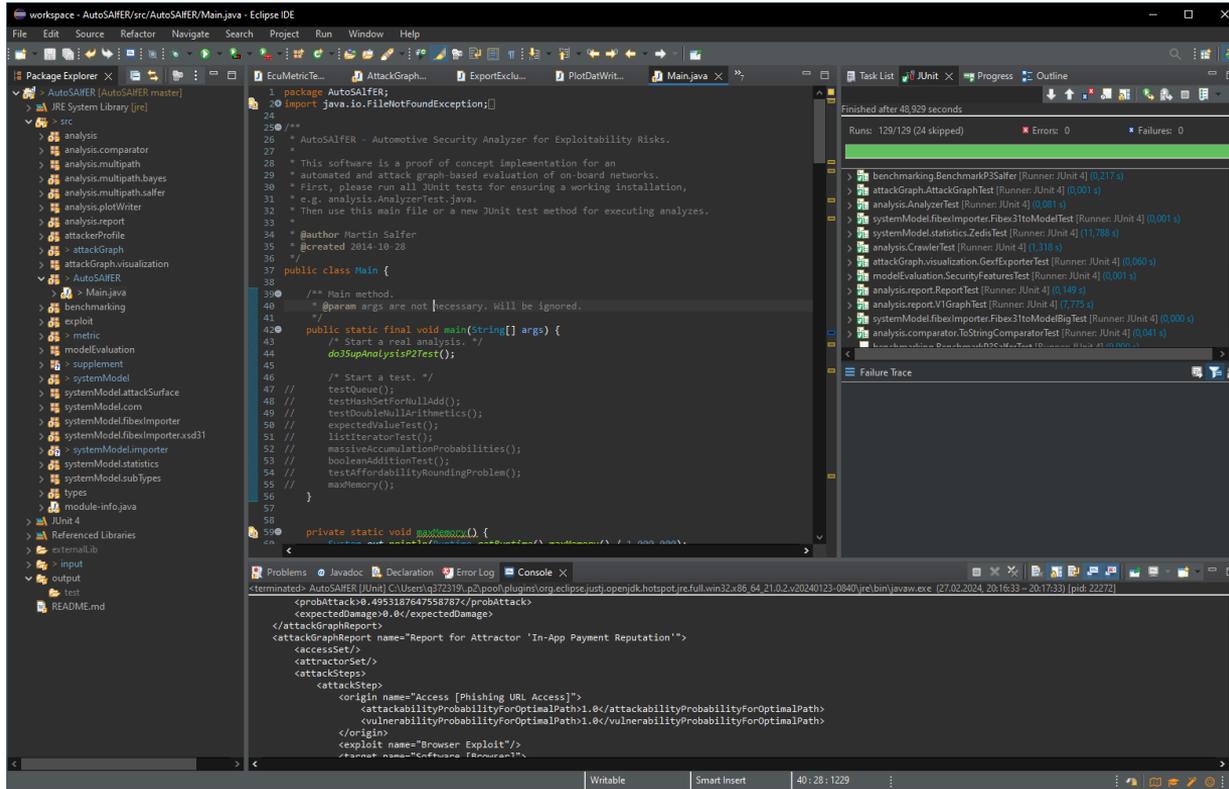
... see the paper on SECRIPT 2015

C4: Implementation and Evaluation of the Models and the Attack Surface Exploitability Quantification

```
REF="_93bda6eb477b49a48417cb897d075c41_25_64ca21c7bf884ed1948ed94cadb4ce33_13"/>
</fx: SIGNAL-TRIGGER>
</fx: SEND-CONDITION>
<ho:DESC xml:lang="de" TYPE="Standard">Die Nachricht wird - ab Klemme 15 EIN - bei jeder Änderung eines
versandt.</ho:DESC>
- <fx:MANUFACTURER-EXTENSION xsi:type="MANUFACTURER-EVENT-CONTROLLED-TIMING-EXTENSION">
- <
  N-SEND-CONDITION>
- <
  SIGNAL-TRIGGER-ON-VALUE-CHANGED>
  <
    SIGNAL-INSTANCE-REF ID-
    REF="_93bda6eb477b49a48417cb897d075c41_25_65578f269cce4492bdf2a550e3eecd13_5">
  <
    SIGNAL-INSTANCE-REF ID-
    REF="_93bda6eb477b49a48417cb897d075c41_25_64ca21c7bf884ed1948ed94cadb4ce33_13
  <
    SIGNAL-TRIGGER-ON-VALUE-CHANGED>
  <
    N-SEND-CONDITION>
</fx:MANUFACTURER-EXTENSION>
</fx:EVENT-CONTROLLED-TIMING>
</fx:TIMINGS>
<fx:PDU-REF ID-REF="_93bda6eb477b49a48417cb897d075c41_25"/>
```

- Tech Stack: Java with xmlbeans, poi, poi-ooxml, gexf4j, StAX, and JUnit
- Ingest File Types:
 - FIBEX,
 - OfficeOpenXML, and
 - XLS
- Output File types:
 - GraphViz DOT
 - GEXF

Demo main



```
workspace - AutoSAIFer/src/AutoSAIFer/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer
AutoSAIFer [AutoSAIFer master]
  JRE System Library [jre]
  src
    analysis
    analysis.comparator
    analysis.multipath
    analysis.multipath.bayes
    analysis.multipath.safer
    analysis.plotWriter
    analysis.report
    attackerProfile
    attackGraph
    attackGraph.visualization
  AutoSAIFer
    Main.java
    benchmarking
    exploit
    metric
    modelEvaluation
    supplement
    systemModel
    systemModel.attackSurface
    systemModel.com
    systemModel.fibexImporter
    systemModel.fibexImporter.xsd31
    systemModel.importer
    systemModel.statistics
    systemModel.subTypes
    types
    module-info.java
  JUnit 4
  Referenced Libraries
  externalLib
  input
  output
  test
  README.md

EcuMetricTe... AttackGraph... ExportExclu... PlotDrawWit... Main.java x

1 package AutoSAIFer;
2 import java.io.FileNotFoundException;
24
25 /**
26  * AutoSAIFer - Automotive Security Analyzer for Exploitability Risks.
27  *
28  * This software is a proof of concept implementation for an
29  * automated and attack graph-based evaluation of on-board networks.
30  * First, please run all JUnit tests for ensuring a working installation,
31  * e.g. analysis.AnalyzerTest.java.
32  * Then use this main file or a new JUnit test method for executing analyzes.
33  */
34 * @author Martin Salfer
35 * @created 2014-10-26
36 */
37 public class Main {
38
39     /** Main method.
40      * @param args are not necessary. Will be ignored.
41      */
42     public static final void main(String[] args) {
43         // Starts a real analysis.
44         do3SupAnalysisP2Test();
45
46         // Start a test.
47         testQueue();
48         testHashSetForNullAdd();
49         testDoubleNullArithmetics();
50         expecteValueTest();
51         listIteratorTest();
52         massiveAccumulationProbabilities();
53         booleanAdditionTest();
54         testAffordabilityRoundingProblem();
55         makeMemory();
56     }
57
58     private static void @SuppressWarnings("unchecked") {
59         // ...
60     }
61 }

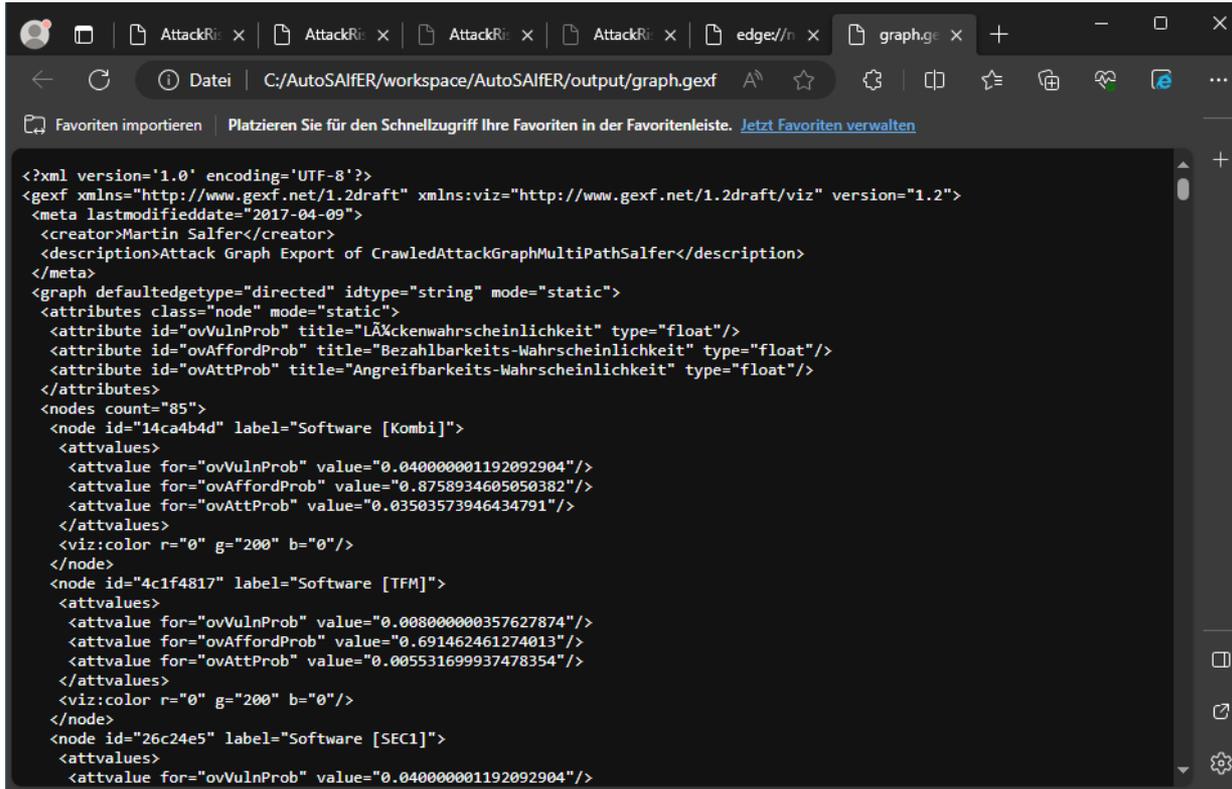
Finished after 48,929 seconds
Runs: 129/129 (24 skipped) Errors: 0 Failures: 0
benchmarking.BenchmarkP2Safer [Runner: JUnit 4] (0,217 s)
attackGraph.AttackGraphTest [Runner: JUnit 4] (0,001 s)
analysis.AnalyzerTest [Runner: JUnit 4] (0,081 s)
systemModel.fibexImporter.Fibex31toModelTest [Runner: JUnit 4] (0,001 s)
systemModel.statistics.ZedisTest [Runner: JUnit 4] (11,788 s)
analysis.CrawlerTest [Runner: JUnit 4] (0,318 s)
attackGraph.visualization.GoxExporterTest [Runner: JUnit 4] (0,060 s)
modelEvaluation.SecurityFeaturesTest [Runner: JUnit 4] (0,001 s)
analysis.report.ReportTest [Runner: JUnit 4] (0,149 s)
analysis.report.VIGraphTest [Runner: JUnit 4] (7,775 s)
systemModel.fibexImporter.Fibex31toModelBigTest [Runner: JUnit 4] (0,000 s)
analysis.comparator.ToStringComparatorTest [Runner: JUnit 4] (0,041 s)
BenchmarkingBenchmarkP2SaferTest [Runner: JUnit 4] (0,000 s)

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<probAttack@b.4953187647558787c/probAttack>
<expectedDamage@b.0/><expectedDamage>
</attackGraphReport>
<attackGraphReport name="Report for Attractor 'In-App Payment Reputation'">
<accessSet>
<attackerSet/>
<attackSteps>
<attackStep>
<origin name="Access [Phishing URL Access]">
<attackabilityProbabilityForOptimalPath>1.0</attackabilityProbabilityForOptimalPath>
<vulnerabilityProbabilityForOptimalPath>1.0</vulnerabilityProbabilityForOptimalPath>
</origin>
<origin name="Browser Exploit/">
<attackabilityProbabilityForOptimalPath>1.0</attackabilityProbabilityForOptimalPath>
<vulnerabilityProbabilityForOptimalPath>1.0</vulnerabilityProbabilityForOptimalPath>
</origin>
</attackSteps>
</attackGraphReport>
</accessSet>
</attackerSet>
</probAttack>

[terminated] AutoSAIFer [JUnit] C:\Users\g372319\p2\pool\plugins\org.eclipse.jst.j2ee.ui.hotspot.jre.full\win32_x86_64_21.0.2_20240123-0840\jre\bin\javaw.exe (27.02.2024, 20:16:33 - 20:17:33) [pid: 22272]
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<probAttack@b.4953187647558787c/probAttack>
<expectedDamage@b.0/><expectedDamage>
</attackGraphReport>
<attackGraphReport name="Report for Attractor 'In-App Payment Reputation'">
<accessSet>
<attackerSet/>
<attackSteps>
<attackStep>
<origin name="Access [Phishing URL Access]">
<attackabilityProbabilityForOptimalPath>1.0</attackabilityProbabilityForOptimalPath>
<vulnerabilityProbabilityForOptimalPath>1.0</vulnerabilityProbabilityForOptimalPath>
</origin>
<origin name="Browser Exploit/">
<attackabilityProbabilityForOptimalPath>1.0</attackabilityProbabilityForOptimalPath>
<vulnerabilityProbabilityForOptimalPath>1.0</vulnerabilityProbabilityForOptimalPath>
</origin>
</attackSteps>
</attackGraphReport>
</accessSet>
</attackerSet>
</probAttack>

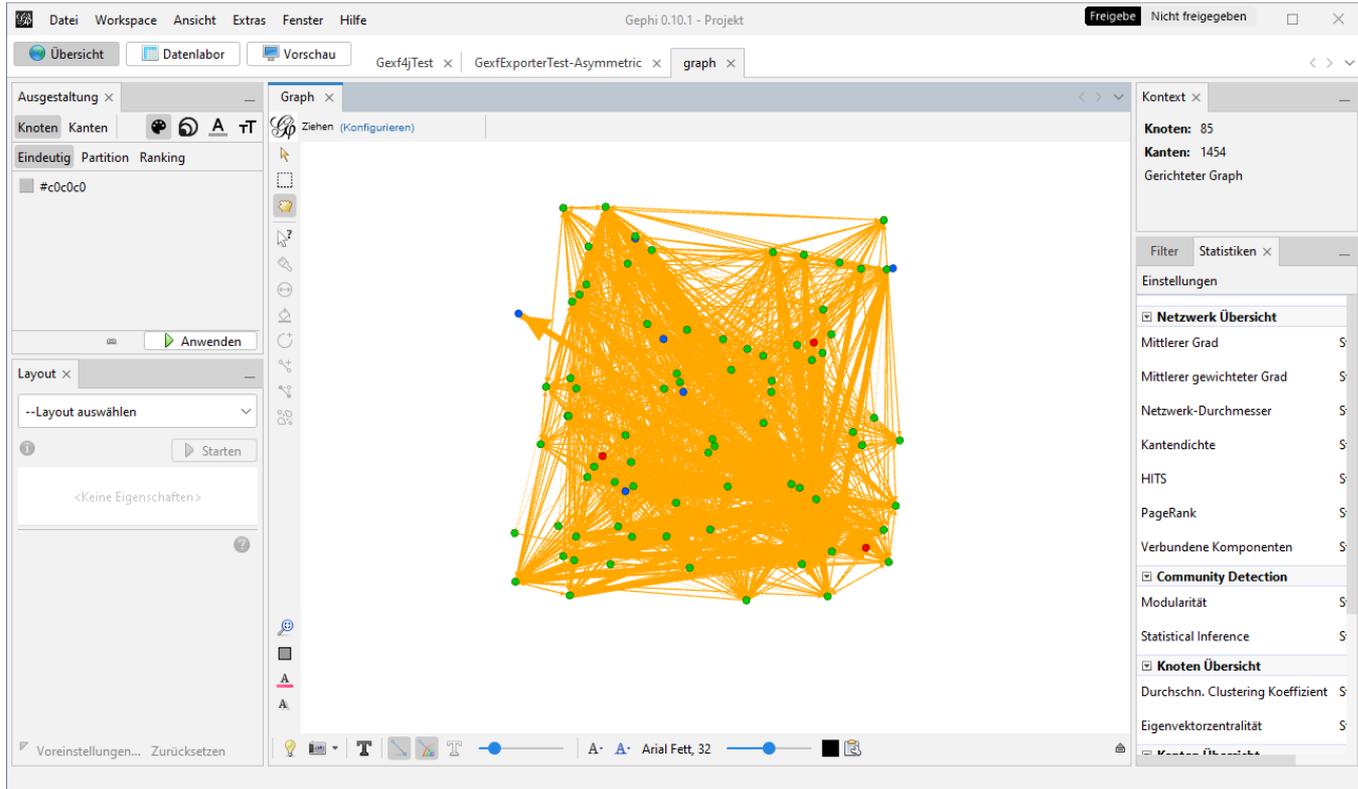
Writeable Smart Insert 40:28 | 1229
```


Demo Output Graph (1/3)

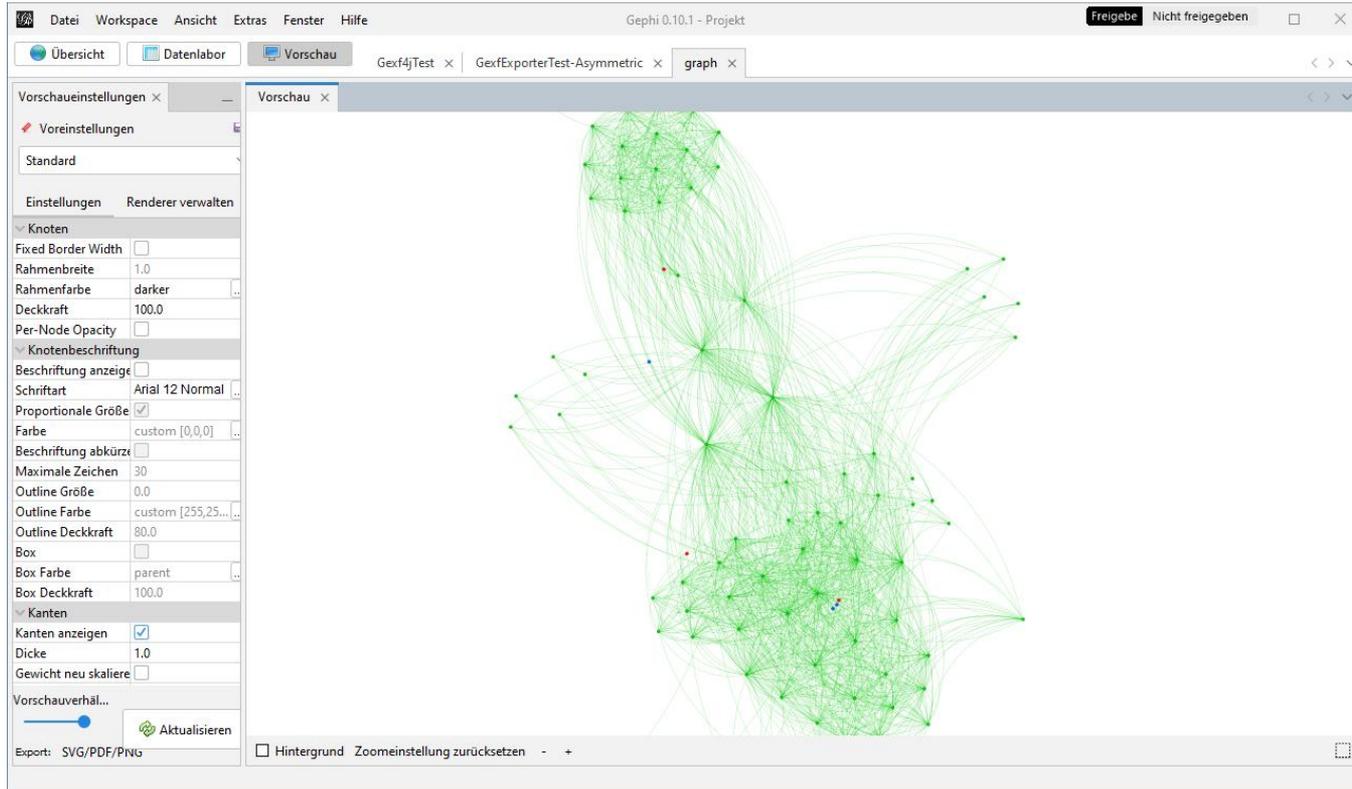


```
<?xml version='1.0' encoding='UTF-8'?>
<gexf xmlns="http://www.gexf.net/1.2draft" xmlns:viz="http://www.gexf.net/1.2draft/viz" version="1.2">
  <meta lastmodifieddate="2017-04-09">
    <creator>Martin Salfer</creator>
    <description>Attack Graph Export of CrawledAttackGraphMultiPathSalfer</description>
  </meta>
  <graph defaultedgetype="directed" idtype="string" mode="static">
    <attributes class="node" mode="static">
      <attribute id="ovVulnProb" title="LÄXckenwahrscheinlichkeit" type="float"/>
      <attribute id="ovAffordProb" title="Bezahlbarkeits-Wahrscheinlichkeit" type="float"/>
      <attribute id="ovAttProb" title="Angreifbarkeits-Wahrscheinlichkeit" type="float"/>
    </attributes>
    <nodes count="85">
      <node id="14ca4b4d" label="Software [Kombi]">
        <attvalues>
          <attvalue for="ovVulnProb" value="0.040000001192092904"/>
          <attvalue for="ovAffordProb" value="0.8758934605050382"/>
          <attvalue for="ovAttProb" value="0.03503573946434791"/>
        </attvalues>
        <viz:color r="0" g="200" b="0"/>
      </node>
      <node id="4c1f4817" label="Software [TFM]">
        <attvalues>
          <attvalue for="ovVulnProb" value="0.008000000357627874"/>
          <attvalue for="ovAffordProb" value="0.691462461274013"/>
          <attvalue for="ovAttProb" value="0.005531699937478354"/>
        </attvalues>
        <viz:color r="0" g="200" b="0"/>
      </node>
      <node id="26c24e5" label="Software [SEC1]">
        <attvalues>
          <attvalue for="ovVulnProb" value="0.040000001192092904"/>
        </attvalues>
      </node>
    </nodes>
  </graph>
</gexf>
```

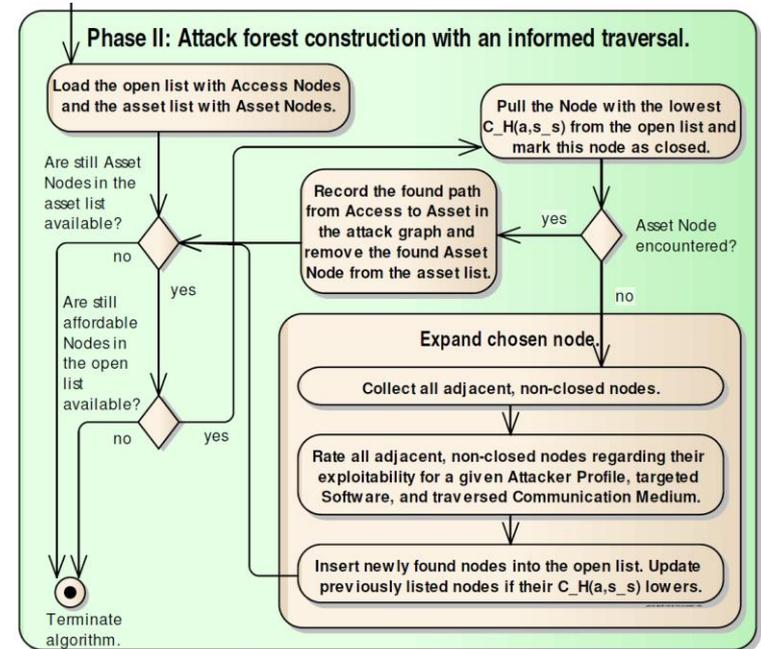
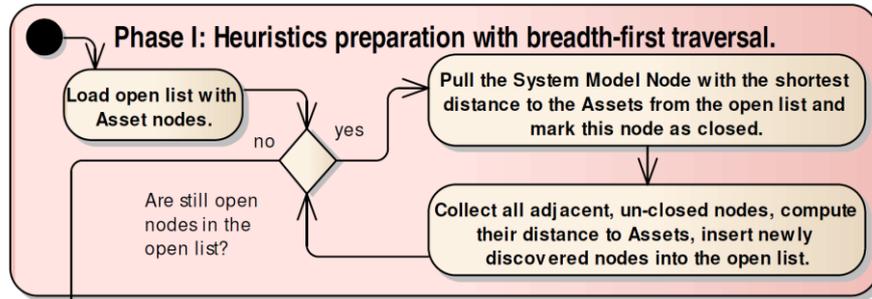
Demo Output Graph (2/3)



Demo Output Graph (3/3)



C5: Design of the Single-Path Attack Graph Algorithm (1/2)



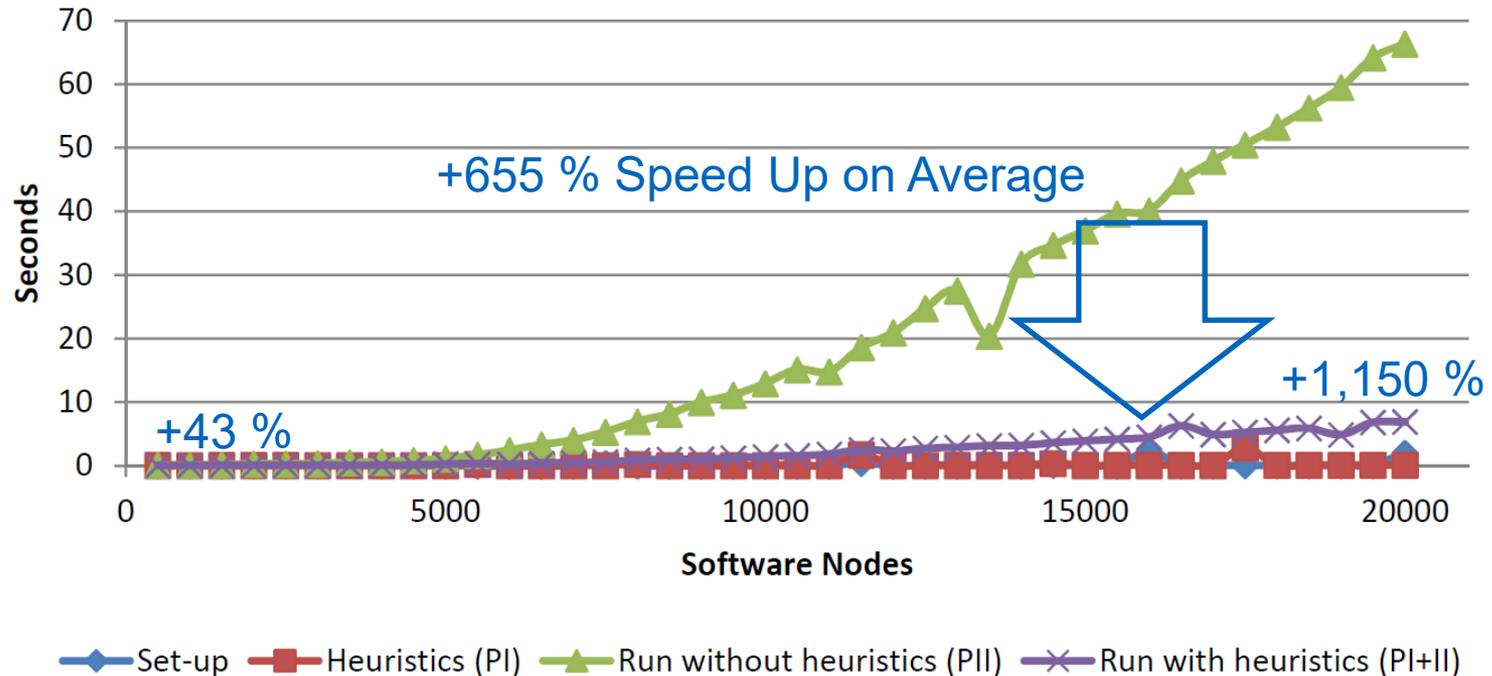
C5: Design of the Single-Path Attack Graph Algorithm (2/2)

- Sheyner (2004): $O(c^n)$:
with NuSMV and Model Checking takes
 - 5 minutes for a network of 2 hosts and
 - 30 minutes for a network of 4 hosts.
- Ou et al. (2006): $O(n^2)$: with Prolog and Logic Programming
- Salfer et al. (2014): $O(n \cdot \log(n))$ and $\Omega(n+m)$ with Java and Imperative Programming with n hosts and m exploits

```
1  /**
2  * Run the second phase of the system model crawl.
3  * This phase finds the shortest path and acts depth-first.
4  * The heuristic must be consistent, as nodes are expanded
5  * only once.
6  * Complexity:  $O(S \cdot (\log(S) + E))$ .
7  */
8  private void runPhase2Crawl() {
9      clearLists(); // Complexity:  $O(1)$ 
10     loadAccessNodesIntoOpenList(); // Complexity  $O(1)$  as of
        assumption 5.
11     loadOpenAttractors(); // Complexity  $O(1)$  as of assumption 5.
12     while (openAttractors.size() > 0 && openList.size() > 0) {
        // Complexity:  $O(S)$  as a software node is closed each
        round.
13         Crawlable c = openList.poll(); // Complexity:  $O(\log(S))$ .
14         c.close(); // Complexity:  $O(1)$ .
15         expand(c); // Complexity:  $O(\log(S) + E)$  as of *
16     }
17     clearLists(); // Complexity:  $O(1)$ 
18 }
```

... see paper at ISC 2014

C6: Implem. and Eval. of the Single-Path Attack Graph Algor.



... see paper at ISC 2014

3. What Other Attack Paths Exist and What is the Total Risk for an Asset?

C7: Probabilistic Model

C8: Multi-Path Attack Graph Algorithm (P3Salfer)

C9: Bayes Network Unsuitability Finding

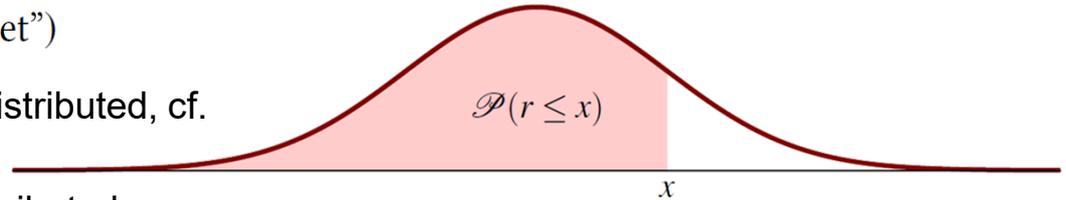
C10: Design and Implementation of an Alternative Algorithm with Bayesian Networks (P3Bayes)

C11: Implementation and Evaluation of the Multi-Path Attack Graph Algorithm (P3Salfer)

C7: Probabilistic Model for Overall Risk Computation (1/2)

Budget $\mathcal{P}_B(a, p) = \Phi(\text{"cost} \leq \text{budget"})$

- Cost Value:** Approximated Gaussian distributed, cf. Central Limit Theorem.
- Budget Value:** Assumed Gaussian distributed



The accumulated probability $\mathcal{P}(r \leq x) = \Phi\left(\frac{x - \mu_r}{\sigma_r}\right)$

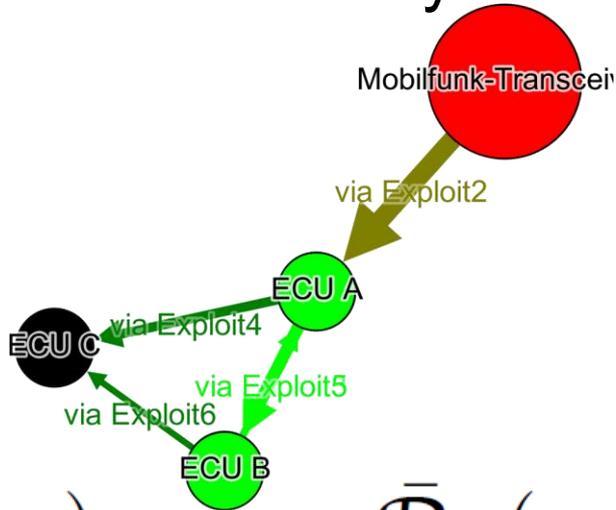


Carl F. Gauß 1777-1855

$$= \Phi\left(\frac{\overset{\text{Budget}}{\downarrow} a_b \mu - \overset{\text{Cost}}{\downarrow} C(a, p) \mu}{\sqrt{a_b^2 \sigma + C(a, p)^2 \sigma}}\right)$$

C7: Probabilistic Model for Overall Risk Computation (2/2)

Vuln. and Attackability



Vulnerability Probability

$$\mathcal{P}_V(p) := \prod_{i=1}^n \mathcal{P}_V(s_{si})$$

Attackability Probability

$$\mathcal{P}_A(p) = \mathcal{P}_V(p) * \mathcal{P}_B(p)$$

Risk

$$\mathcal{R}(a, s_a) := s_{av} \bar{\mathcal{P}}_A(a, s_a) = s_{av} \frac{\sum_{i=1}^n \mathcal{P}_A(a, p_1)}{n}$$

... see paper at ARES 2018

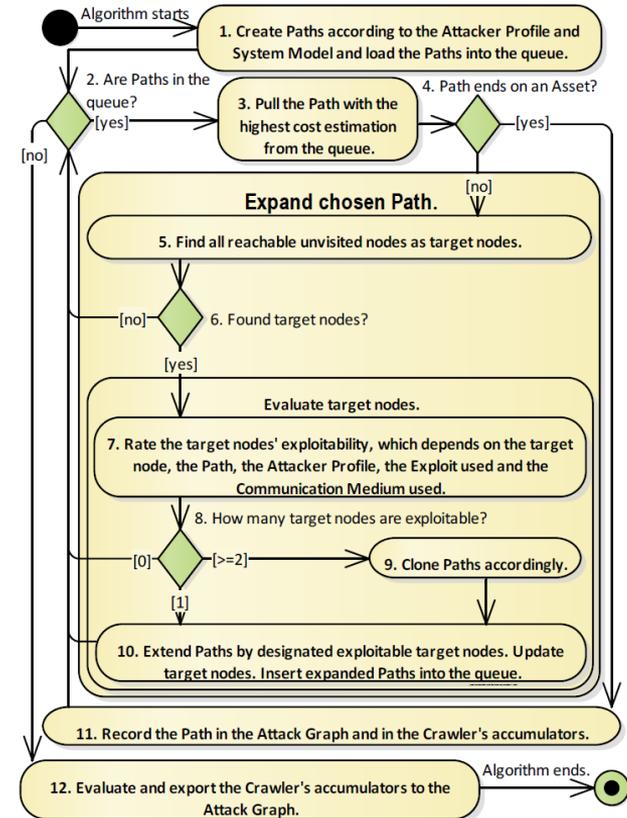
C8: Design of the Multi-Path Attack Graph Algorithm (P3Salfer)

Performance Considerations:

- Early Path Drops, still >95% coverage @ 2σ .
- Fast Gauß Computation Instead of Sampling
- “MapReduce-able”: > 94% computational time is independent.

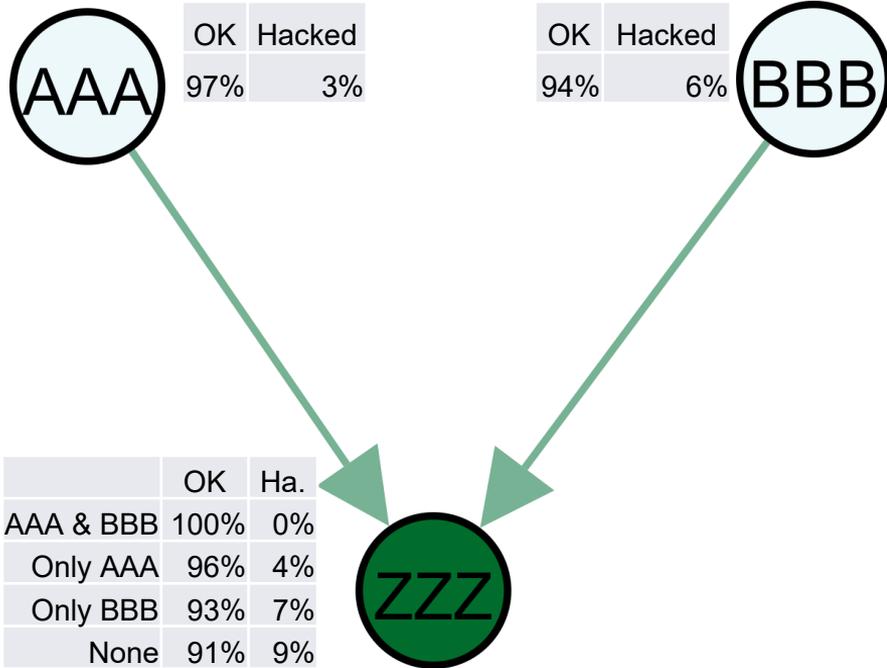
Termination Guarantee

- Budget Overrun
- Probability Insignificance
- Reaching all Assets
- Expansion Exhaust



... see paper at ARES 2018

C9: Bayes Network Unsuitability Finding



Stochastic, graphical models



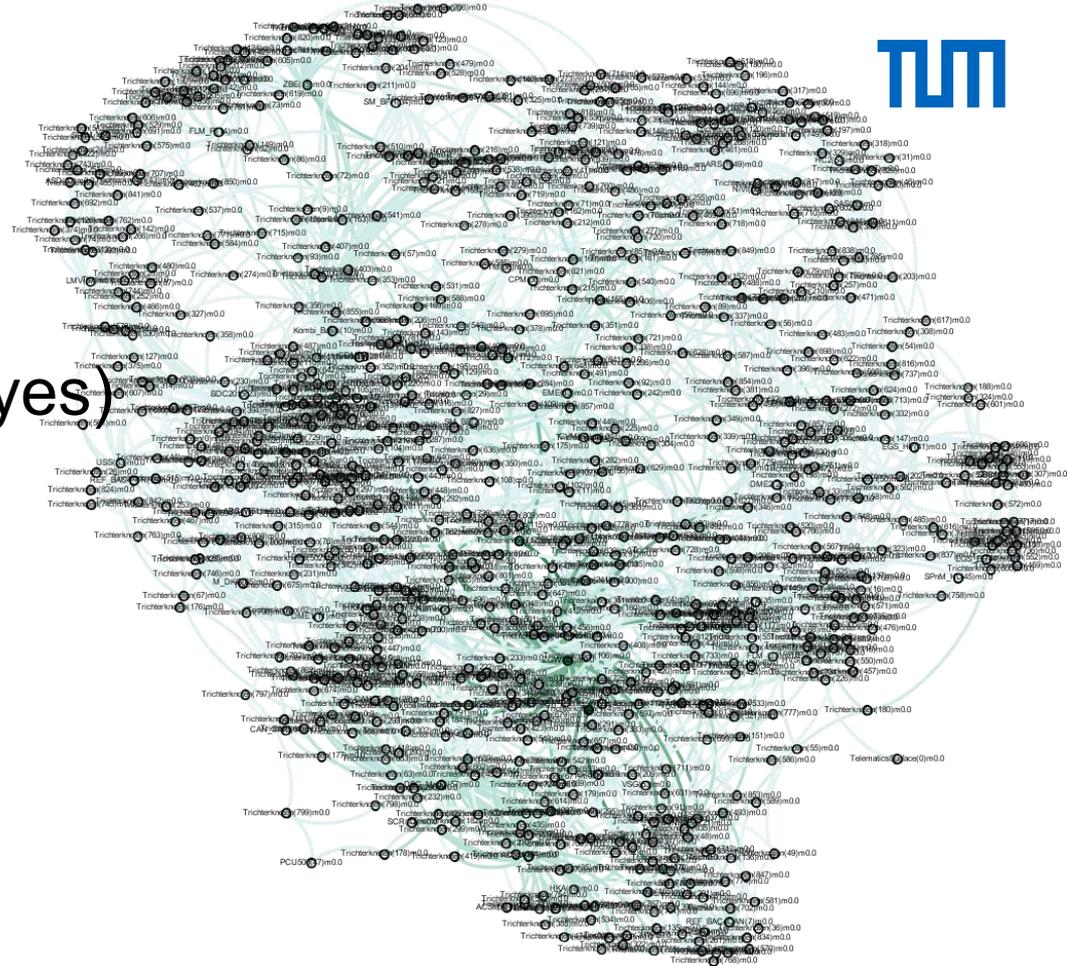
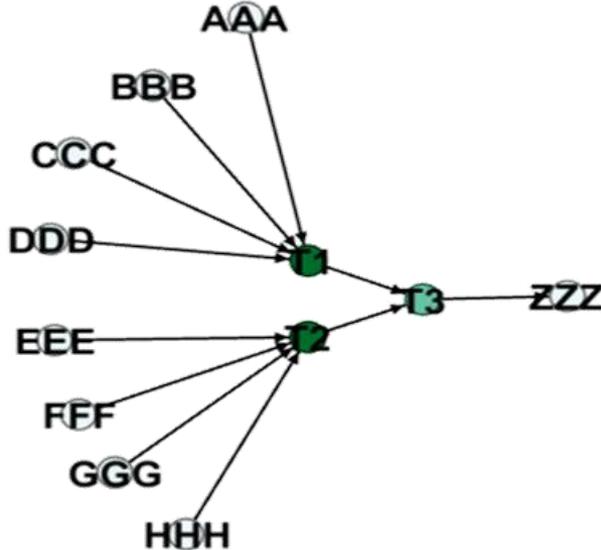
Thomas Bayes 1701-1761

But:

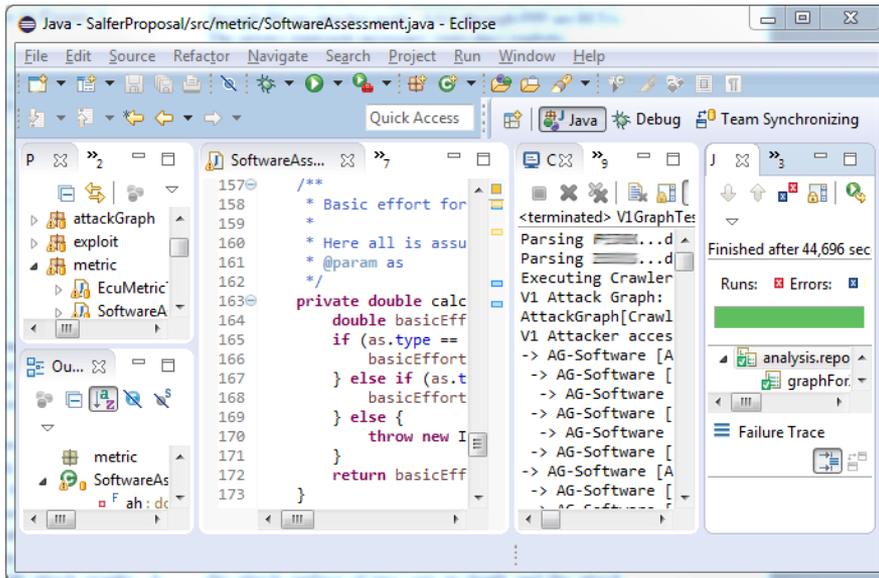
- Distribution Table Explosion (or node growth)**,
Space complexity: $2^{(n+4)}B$; $n=30$ inputs \rightarrow ~17 GB.
- Cycle Inability**
- Probabilities are static / history-agnostic**

... see paper at ARES 2018

C10: Design and Implem. of an Altern. Algorithm with Bayesian Networks (P3Bayes)

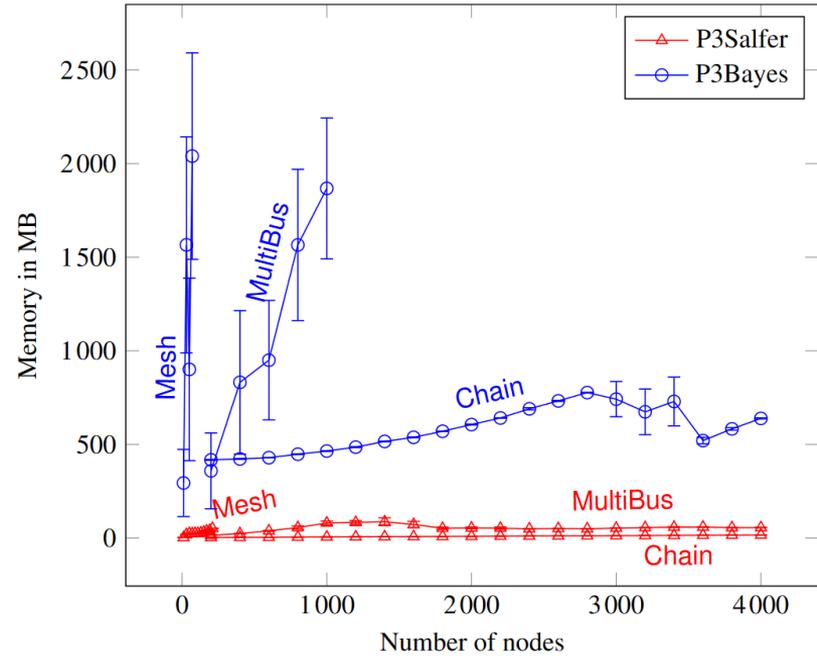
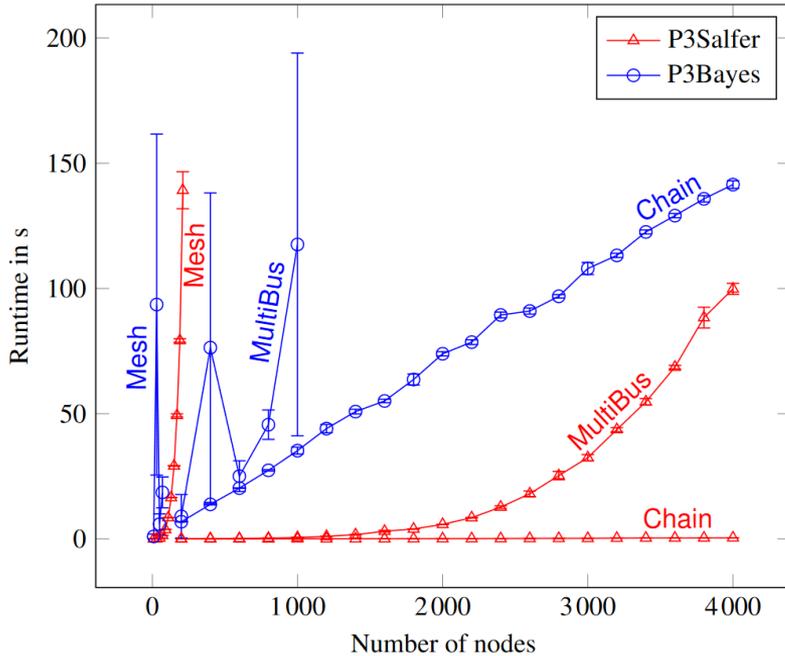


C11: Implementation and Evaluation of the Multi-Path Attack Graph Algorithm (P3Salfer): Feature Comparison



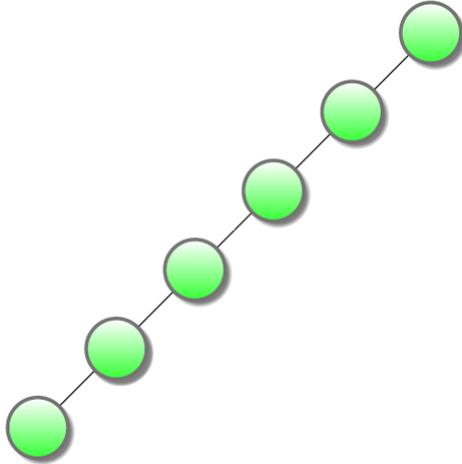
	P3Salfer	P3Bayes
Model	See above	Bayes Network
Inference	See above	Junction Tree
Technology	Standard JDK (Collections, JAXB)	Standard JDK, Unbbayes.jar
Cycles	OK	Not OK
Probabilities	Dynamic	Static
In Parallel	Single-Thread	Multi-Threaded

C11: Benchmark Result: 200-5000x faster, 40-200x smaller

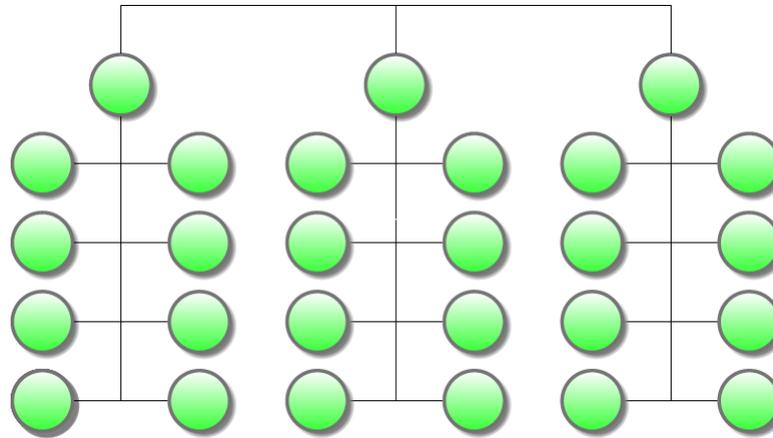


C11: Benchmarking with Scalable Synthetic Models

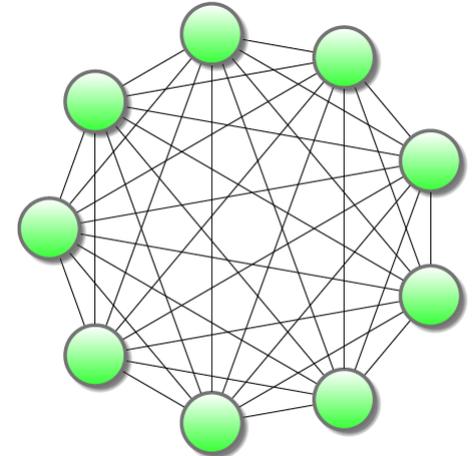
Chain



Multibus

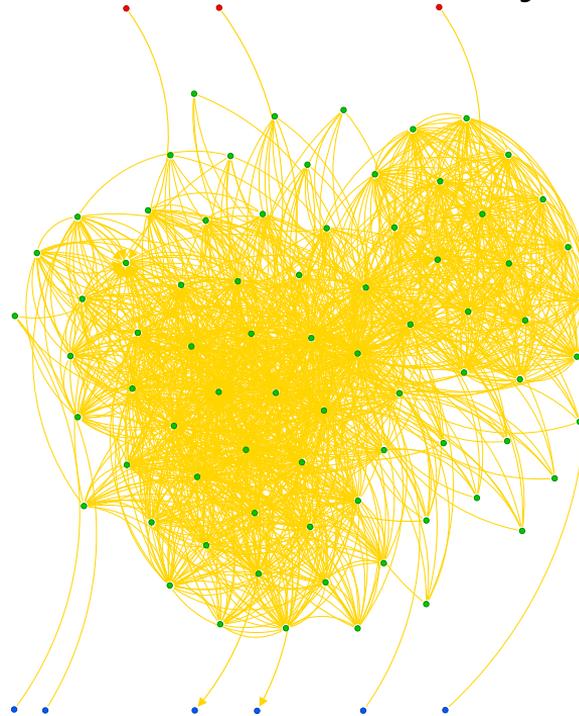


Mesh



harder

C11: Application Test with Realistically Sized Graphs



Future Work

- Graphical user interface
- More input for higher assurance and higher accuracy
 - more development data
 - penetration test tools input for undocumented attack surfaces
- Networks beyond automotive
 - avionics/railway
 - operational technology
 - poorly segmented networks
- Maven packaging
- Documentation

```
attractorWeb = new Attractor("web2.0", swBrowser, new Resources(0.045, 0.02d));
attractorPay1 = new Attractor("In-App Payment Financial", swPayment);
attractorPay2 = new Attractor("In-App Payment Reputation", swPayment);
Attractor vCarAccess = new Attractor("Car life module access", swCas, new Resources(50_000d, 10_000d));
Attractor vCarSafety1 = new Attractor("Car DSC Safety", swDsc, new Resources(1_000_000, 1_000_000d));
Attractor vCarSafety2 = new Attractor("Car EHF Safety", swEhf, new Resources(1_000_000, 1_000_000d));

attractors.add(attractorWeb);
attractors.add(attractorPay1);
attractors.add(attractorPay2);
attractors.add(vCarAccess);
attractors.add(vCarSafety1);
attractors.add(vCarSafety2);
for (Attractor a : attractors) {
    sysModel.addNode(a);
}
attractorsPayment.add(attractorPay1);
attractorsPayment.add(attractorPay2);

/* Attacker Profile. */
final AttackerProfile attackerProfile = new AttackerProfile("Internet Harvest Group", 5, new Resources(300_000, 100_000));
attackerProfile.addAccess(new Access("Phishing URL Access", swBrowser, 10_000, new Resources(0.1, 0.1d)));
attackerProfile.desires(attractorWeb);
attackerProfile.desires(attractorPay1, new Resources(1_000, 1_000d));
attackerProfile.desires(attractorPay2, new Resources(10_000, 10_000d));
attackerProfile.desires(vCarAccess);
attackerProfile.desires(vCarSafety1, new Resources(0, 1_000d));
attackerProfile.desires(vCarSafety2, new Resources(0, 1_000d));

Skill skillBrowser = new Skill("Browser");
Skill skillDbus = new Skill("dbus");
attackerProfile.addSkill(skillBrowser, 0.8f);
attackerProfile.addSkill(skillDbus, 0.2f);

/* Exploit database. */
final HashSet<Exploit> exploits = new HashSet<Exploit>();
final PotentialExploit exploit1 = new PotentialExploit("Browser Exploit", swBrowser, new Resources(), new Resources(5));
final PotentialExploit exploit2 = new PotentialExploit("Browser Extraction", attractorWeb, new Resources(), new Resou);
final PotentialExploit exploit3 = new PotentialExploit("Payment Exploit", swPayment, new Resources(), new Resources(1));
final PotentialExploit exploit4 = new PotentialExploit("Payment Extraction", attractorPay1, new Resources(), new Resou);
final PotentialExploit exploit5 = new PotentialExploit("Payment Reputation", attractorPay2, new Resources(), new Resou);
final PotentialExploit exploit6 = new PotentialExploit("Life Module Access", vCarAccess, new Resources(), new Resou);
final PotentialExploit exploit7 = new PotentialExploit("DSC Safety Control", vCarSafety1, new Resources(), new Resou);
final PotentialExploit exploit8 = new PotentialExploit("EHF Safety Control", vCarSafety2, new Resources(), new Resou);
final Set<PotentialExploit> exploitsGenerally = new HashSet<PotentialExploit>();
for (Set<Software> set : swSet) {
    for (Software sw : set) {
        final PotentialExploit potentialExploit = new PotentialExploit("General Potential Exploit for " + sw.toString()
            + " on " + sw.getPlatform(), sw, new Resources(), new Resources(1));
        exploitsGenerally.add(potentialExploit);
    }
}
```

```
<terminated> ReportTest [JUnit C:\Program Files\Java\jre7\bin\javaw.exe (23.04.2014 11:06:03)
AttackGraph[CrawledAttackGraph]:
Internet Harvest Group access by Access [Phishing URL Access] with Potr
-> AG-Software [Browser] exploits further with Potential Exploit [Brow
-> AG-Software [Browser] exploits further with Potential Exploit [Paym
-> AG-Software [Payment Service] exploits further with Potential Explo
-> AG-Software [Payment Service] exploits further with Potential Explo
-> AG-Software [Browser] exploits further with Potential Exploit [Gene
-> AG-Software [CIC] exploits further with Potential Exploit [Genera
-> AG-Software [Z00] exploits further with Potential Exploit [Genera
-> AG-Software [DSC] exploits further with Potential Exploit [DSC Sa
-> AG-Software [Z00] exploits further with Potential Exploit [Genera
-> AG-Software [CAS] exploits further with Potential Exploit [Life I
-> AG-Software [Z00] exploits further with Potential Exploit [Genera
-> AG-Software [EHF] exploits further with Potential Exploit [EHF Sa

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AttackRiskReport date="Wed Apr 23 11:06:09 CEST 2014" title="Generated
  <attackerProfile name="Internet Harvest Group" expectedNumber="5">
    <budget>
      <moneyExpectedValue>300000.0</moneyExpectedValue>
      <moneyStandardDeviation>100000.0</moneyStandardDeviation>
    </budget>
    <accessSet>
      <access name="Phishing URL Access" numberOfTargetableVehic
        <accessCostPerVehicle>
          <moneyExpectedValue>0.1</moneyExpectedValue>
          <moneyStandardDeviation>0.1</moneyStandardDeviation>
        </accessCostPerVehicle>
      </accessSet>
    </AttackProfile>
  </attackerProfile>
  <attackGraphReport name="Report for Attractor 'web2.0'">
    <attackerSet/>
    <attackerSteps>
      <attackStep>
        <origin name="Access [Phishing URL Access]"/>
        <exploit name="Browser Exploit"/>
        <target name="Software [Browser]"/>
      </attackStep>
      <attackStep>
        <origin name="Software [Browser]"/>
        <exploit name="Browser Extraction"/>
        <target name="Attractor [web2.0]"/>
      </attackStep>
    </attackerSteps>
    <cost>
      <moneyExpectedValue>8500.0</moneyExpectedValue>
      <moneyStandardDeviation>4500.959188320257</moneyStandardDe
    </cost>
  </attackGraphReport>
</AttackRiskReport>
```

Further Material Overview (1/6)

EU Patents

1. M. Salfer, **System and Method for Simulating and Foiling Attacks on a Vehicle On-Board Network**, EP3490223B1, 24.11.2017, <https://patents.google.com/patent/EP3490223B1>.
2. M. Salfer, Secure and User-Specific Data Use in Motor Vehicles, 15801365.6, EP3235212, 2015-11-18, <https://patents.google.com/patent/EP3235212A1/en>
3. M. Salfer, Methods and devices for transmitting and identifying radio identifiers, EP3766268, 2022-09-21, <https://patents.google.com/patent/EP3766268A1>

US-Patent 11,716,165: M. Salfer, Methods and Devices for the Concealment of Radio Identifiers and Transmitter Positions, application: 2019-04-18, patent grant: 2023-08-01, patent expiration: 2040-01-30.

China Patent 111788569: M. Salfer, Method and apparatus for hiding radio identifiers and transmitter locations,, application: 2019-04-18, patent grant: 2023-09-12, patent expiration: 2039-04-18.

Further Material Overview (2/6)

German Patent Filings

1. M. Salfer, Verfahren und Vorrichtungen zur Verschleierung von Funkkennungen und Senderpositionen, [102018206476.8](#), 26.04.2018,.
2. M. Salfer, Verfahren und Vorrichtungen zum Senden und Identifizieren von Funkkennungen, [102018203949.6](#), 15.03.2018,.
3. M. Salfer, **Verfahren zur Ermittlung eines Angriffswegs in einem Systemmodell und Computerlesbares Speichermedium**, [102014212419.0](#) , 27.06.2014
4. M. Salfer, D. Burgkhardt und S. Zimmermann, System und Verfahren für einen beschränkten Zugang zu einem Fahrzeug, [102014219502.0](#), 26.09.2014
5. M. Salfer, Sichere Datennutzung in Kraftfahrzeugen, [102014226219.4](#), 17.12.2014
6. M. Salfer, Pseudozufällige Funkkennungen für mobile Funkvorrichtungen, [102015204210.3](#), 10.03.2015
7. M. Salfer und Z. Ren, Notbelieferung eines Fahrzeugs mit Kraftstoff, [102014213023.9](#), 04.07.2014
8. M. Salfer, Beschichtung für ein Objekt sowie Fahrzeug umfassend eine Beschichtung., [102014208531.4](#), 07.05.2014
9. M. Salfer, C. Lottermann und P. Hoffmann, Beobachtung einer Umgebung eines Fahrzeugs, [102014206928.9](#), 10.04.2014

Further Material Overview (3/6)

Conference Papers

1. M. Salfer und C. Eckert, "**Attack Graph Automation for Assessing Security Risks of Automotive On-Board Networks**" (P3Salfer/P3Bayes), 13th International Conference on Availability, Reliability and Security (ARES 2018), Hamburg, Germany.
2. M. Salfer und C. Eckert, "**Attack Surface and Vulnerability Assessment of Automotive Electronic Control Units**" (Probability Calculations), 12th International Conference on Security and Cryptography (SECRYPT 2015), Colmar, France.
3. M. Salfer, H. Schweppe, und C. Eckert, „**Efficient Attack Forest Construction for Automotive On-board Networks**“ (Model + PI/PII), Information Security – 17th Int. Conference (ISC 2014), Hong Kong.
4. N. Broy, S. Goebel, M. Hauder, T. Kothmayr, M. Kugler, F. Reinhart, M. Salfer, K. Schlieper, und E. André, „A cooperative in-car game for heterogeneous players“, in (AutomotiveUI 2011), Salzburg, Austria.
5. M. Salfer, S. Wohlgemuth, S. Schrittwieser, B. Bauer, und I. Echizen, „Data Provenance with Watermarks for Usage Control Monitors at Disaster Recovery“, in (IEEE iThings/CPSCoM 2011), China.

Further Material Overview (4/6)

Journal Article

1. Hans-Ulrich Michel, Dirk Kaule, und Martin Salfer, „Virtualisierung: Vision einer intelligenten Vernetzung.“, *Elektronik automotive*, Nr. 04/2012, S. 28–32, Apr. 2012.

Poster

1. M. Salfer, "IT-Sicherheit im Auto - Graphen-basierte Angriffssicherheitsevaluation von automobilen Bordnetzen.", Poster + Proceedings, BMW ProMotion Dialogtag 2014
2. M. Salfer, "Sicherheitsarchitektur - Quantitative Bewertung von Sicherheitsarchitekturen virtualisierter Mehrkern-Steuergeräte.", Poster + Proceedings, BMW ProMotion Dialogtag 2013
3. M. Salfer, "E/E-Sicherheitsarchitektur - IT-Sicherheit virtualisierter Mehrkern-Systeme im Fahrzeug.", Poster + Proceedings, BMW ProMotion Dialogtag 2012

Further Material Overview (5/6)

The software for the contributions

- C4 “Implementation and Evaluation of the Models and the Attack Surface Exploitability Quantification”,
- C6 “Implementation and Evaluation of the Single-Path Attack Graph Algorithm”,
- C10 “Design and Implementation of an Alternative Algorithm with Bayesian Networks (P3Bayes)”, and
- C11 “Implementation and Evaluation of the Multi-Path Attack Graph Algorithm (P3Salfer)”

Is published (with the main algorithms in `/src/analysis/Crawler.java`) as

“Automotive Security Analyzer for Exploitability Risks” (AutoSAIfER) on

<https://github.com/MarSalfer/AutoSAIfER/>

Head start with libraries in <https://syncandshare.lrz.de/dl/fiCZTBTseBwuqoJiwqDUvD/externalLib.zip>

Further Material Overview (6/6)

M. Salfer, **Automotive Security Analyzer for Exploitability Risks**: An Automated and Attack Graph-Based Evaluation of On-Board Networks, Springer Vieweg, 2024-03-15, ISBN 978-3-658-43505-9, <https://link.springer.com/book/9783658435059>

“Smart” features are “not so smart” if these introduce further attack vulnerabilities.

Someone can

- **pay now by investing in proper security** or
- **pay later** by losing lives, assets, or revenue.

