Program Analysis for System Security and Reliability

Martin Vechev
Spring 2018
Professor of Computer Science at ETH since January 2012

Sofia, Bulgaria

SFU, Canada, B.Sc.

Cambridge, England, PhD

Today

DEEP CODE
AI for Code, Adviser

CHAINSECURITY
Blockchain Security, Adviser

Professor at ETH Zurich,
Lead SRL: http://www.srl.inf.ethz.ch

Researcher @ IBM T.J. Watson Research Center,
New York, USA
Today

Overview of Material

Course organization
The DAO Attacked: Code Issue Leads to $60 Million Ether Theft

The DAO, the distributed autonomous organization that had collected over $150m worth of the cryptocurrency ether, has reportedly been hacked, sparking a broad market sell-off.

A leaderless organization comprised of a series of smart contracts written on the Ethereum codebase, the DAO has lost 3.6m ether, which is currently sitting in a separate wallet after being split off into a separate group of accounts called "child DAOs".

CloudFlare apologizes for Telia screwing you over

Unhappy about massive outage

21 Jun 2016 at 20:34, Kieren McCarthy

Tesla and GM self-drive cars involved in road collisions

24 January 2018
WHAT IN TARNATIONS
IS GOING ON HERE

CloudFlare apologizes for Telia screwing you over
Unhappy about massive outage

Tesla and GM self-drive cars involved in road collisions

The DAO Attacked: Code Issue Leads to $60 Million Ether Theft

The DAO, the distributed autonomous organization that had collected over $150m worth of the cryptocurrency ether, has reportedly been hacked, sparking a broad market sell-off.

A leaderless organization comprised of a series of smart contracts written on the ethereum codebase, the DAO has lost 3.6m ether, which is currently sitting in a separate wallet after being split off into a separate grouping dubbed a "child DAO".
THE FORCE AWAKENS

Year is 2018...Software has penetrated all areas of life...making decisions instead of humans...threatening to seize all control with its massive size...it seems all hope to make it behave as intended is lost...only a small band of resistance fighters stand against the rising tyranny...hoping to restore a spark of hope to the fight...
Philosophy upon which the course is built...

Problems now too complex, so a single method tends to not work too well. Thus:

1. Adopt an inter-disciplinary approach: study methods from different areas.
2. Focus on an application domain.
3. Phrase the problem formally.
4. Produce a mathematically clean solution. Aim for guarantees.
5. Build a real system and run it to test your solution.
6. Aim to generalize solution and identify core reusable concept.

“...I regard the success of this approach, which has been the basis for much future work, as a triumph of the power of a simple mathematical idea over ad hoc hacking. Yes, the real world is messy and complicated, but one should try to base algorithms on clean, comprehensible mathematical ideas and only complicate them when absolutely necessary…”

“Register Allocation and Spilling via Graph Coloring”
Gregory Chaitin, Retrospective
Course Breakdown: by applications

- Programmable Networks
- Blockchain Security
- Attacks and Defenses of Deep Learning
- Probabilistic Security
## Course Overview:
Core Methods, Systems, Applications

### Core Methods
- Probabilistic Graphical Models
- Datalog
- Abstract Interpretation
- Synthesis
- Gradient-Based Search
- SMT
- Probabilistic Programming

### Systems
- synet.ethz.ch
- Bayonet
- apk-deguard.com
- psisolver.org
- Batfish
- securify.ch
- Spire
- jsnice.org
- Debin
- ai2.ethz.ch

### Applications
- Programmable Networks
- Blockchain Security
- Attacks and Defenses on Deep Learning
- Probabilistic Security

We will see some of the methods used in very different systems and applications
Course Breakdown

- Programmable Networks
- Blockchain Security
- Attacks and Defenses of Deep Learning
- Probabilistic Security
Motivation

Amazon server outage affects millions of companies and causes online chaos

Amazon's massive AWS outage was caused by human error

CloudFlare apologizes for Telia screwing you over

Unhappy about massive outage

Level3 switch config blunder blamed for US-wide VoIP blackout

Network dropped calls because it was told to

The summer of network misconfigurations


Amazon today blamed human error for the the big AWS outage that took down a bunch of large internet sites for several hours on Tuesday afternoon.
Motivation

Network Configuration is Hard and Misconfigurations are Common
Why is Configuration Hard?

High-level, global routing requirements

OSPF + Static routes

Low-level, local router configurations

Multiple interacting routing protocols (OSPF, BGP, ..)

Network N1

Network N2

Network N3

R1: Packets from N1 to N2 must follow the path A → D

R2: Packets from N1 to N3 must follow the path A → B → C → D
R1: Packets from N1 to N2 must follow the path A → D
R2: Packets from N1 to N3 must follow the path A → B → C → D
R1: Packets from N1 to N2 must follow the path A → D 
R2: Packets from N1 to N3 must follow the path A → B → C → D
R1: Packets from N1 to N2 must follow the path A → D

R2: Packets from N1 to N3 must follow the path A → B → C → D

Add a static route to A configuration

Configure A to prefer static routes over OSPF

Static routes table

<table>
<thead>
<tr>
<th>Network</th>
<th>NextHop</th>
</tr>
</thead>
<tbody>
<tr>
<td>N3</td>
<td>B</td>
</tr>
</tbody>
</table>

OSPF link cost

Network N1

Network N2

Network N3
R1: Packets from N1 to N2 must follow the path A → D

R2: Packets from N1 to N3 must follow the path A → B → C → D
Router configurations must be such that:

1. A prefers **static routes** over **OSPF**
2. A has a static route to B for N3
3. A → D → N2 must be lowest cost from A to N2
4. B → C → D → N3 must be lowest cost from B to N3

---

R1: Packets from **N1** to **N2** must follow the path A → D

R2: Packets from **N1** to **N3** must follow the path A → B → C → D
Current Practice

Problems and Challenges

- Diversity in protocol expressiveness
- Protocol dependencies
- No correctness guarantees

Initially not configured

All routers are configured
What are we going to do about it?

In lecture 2, we will study **automated analysis of configurations** and specifically the Batfish system. With this approach, we can automatically check whether a specific configuration is correct.

In lecture 3, we will study **automated synthesis of configurations** and specifically the SyNET system. In this way we can automatically generate the desired configuration!

In lecture 4, we will study **probabilistic analysis of configurations**, showing how to use probabilistic programming to prove key properties under uncertainty (e.g., network link failures). In particular we will cover the Bayonet system.

These lectures explain the formal techniques (e.g., Datalog, SMT, Probabilistic Inference) and how to apply these to networks. They also outline ongoing research in the area (e.g., network chatbots, network obfuscation, etc).
## What will we learn in this part?

### Core Methods
- Probabilistic Graphical Models
- Datalog
- Abstract Interpretation
- Synthesis
- SMT
- Probabilistic Programming

### Systems
- **Bayonet**: synet.ethz.ch, apk-deguard.com
- **Batfish**: psisolver.org, securify.ch
- **Spire**: jsnice.org, Debin
- **Spire**: ai2.ethz.ch

### Applications
- **Programmable Networks**: Attacks and Defenses on Deep Learning
- **Blockchain Security**: Probabilistic Security
Sample efforts in this space

Academia

Cornell-Princeton Center for Network Programming
http://network-programming.org/

Ice center@ETH
http://ice.ethz.ch

Start-ups

Veriflow
https://www.veriflow.net/

Forward Networks
https://www.forwardnetworks.com/

Intentionet
https://www.intentionet.com/

Barefoot Networks
https://barefootnetworks.com/

Overall theme: Languages/Analysis/Synthesis for Networks
Course Breakdown

Programmable Networks

Blockchain Security

Attacks and Defenses of Deep Learning

Probabilistic Security
Growth of Ethereum

**Billions** of USD invested in Ethereum smart contracts

- **Jan 2016**: $100M
- **Jan 2017**: $1B
- **Jan 2018**: $100B

Decentralized businesses are built on top of smart contracts.
Smart Contract Security Bugs in the News

The DAO Attacked: Code Issue Leads to $60 Million Ether Theft

The DAO, the distributed autonomous organization that had collected over 14 million ether, has reportedly been hacked, sparking a broad market sell-off.

A leaderless organization comprised of a series of smart contracts written to manage funds for a group of investors, it was thought to be unhackable. However, the DAO has lost 3.6 million ether, which is currently sitting in a separate wallet account.

The DAO Falls Victim to Cyber Attack Leading Ethereum to Crash Over 20%

The event is still ongoing as hackers have already stolen over 3.5 million ETH from the DAO's coffers.

Wallet bug freezes more than $150 million worth of Ethereum

Over $30 million worth of cryptocurrency have been stolen in another hacker attack targeting a blockchain startup.

Security

Theft due to security issue with Parity's wallet software

A bug in Parity, a popular wallet for Ethereum, may have resulted in近35 million at current prices. The amount of the stolen ether has been confirmed by Etherscan.io.
What is a Smart Contract?

- Small programs that **handle money** (ETH)
- Executed on the Ethereum blockchain
- Written in high-level languages (*e.g.*, Solidity)
- **No patching** after release

```
mapping(address => uint) balances;

function withdraw() {
  uint amount = balances[msg.sender];
  msg.sender.transfer(amount);
  balances[msg.sender] = 0;
}
```

What can go wrong when programs handle billions of USD?
July 2017: Parity Multisig Bug 1
Parity Multisig **Bug 1**: Unprivileged Write to Storage

A user may change the wallet's owner

[the slide is very simplified, in practice, you need to find a way to change 'owner']

An attacker used a similar bug to **steal $30M** in July 2017
A user *froze $170M* by “accidentally” deleting the wallet library.
More Security Bugs...

- Unexpected ether flows
- Insecure coding, such as unprivileged writes (*e.g.*, *Multisig Parity bug*)
- Use of unsafe inputs (*e.g.*, reflection, hashing, ...)
- Reentrant method calls (*e.g.*, *DAO bug*)
- Manipulating ether flows via transaction reordering
What are we going to do about it?

In lecture 5, we will provide an overview of different blockchains (e.g., Ethereum, Bitcoin, Hyperledger, etc), their programming language, their operation and the kind of security and reliability issues that can arise in each.

In lecture 6, we will study automated analysis of smart contracts via Securify (http://securify.ch), the first automated formal audit platform for Ethereum Smart Contracts, commercialized by ChainSecurity (http://chainsecurity.com).

These lectures will show how to use same method (e.g., Datalog) but used in a very different context than with networks. It will also introduce Abstract Interpretation. They will also outline some of the ongoing research in the area, specifically, the different kinds of automated audit platforms being built today.
Released in Fall 2017, so far:

- 95% positive feedback
- > 15K uploaded smart contracts
- > 1000 users signed up for updates

Interesting discussions on Reddit

[-] mcgravier 22 points 12 days ago
Seems almost too good to be true :) What are the limitations and how exactly does it work under the hood?

[-] AlexanderSupersloth 12 points 12 days ago
Please, someone, humour a layman: how can a Turing complete language be formally verified?

I thought formally verifiable languages were necessarily not Turing complete, and we can therefore not formally verify Solidity.

[-] pirapira Ethereum - Yoichi Hirai 3 points 11 days ago
It's great that the authors of the tool are aware they are approximating the set of behaviors in the growing direction. That's the way to go if they seek safety properties without false-negatives. I'm interested how they compare their EVM semantics against other EVM implementations in the wild.
What will we learn in this part?

**Core Methods**
- Probabilistic Graphical Models
- Synthesis
- Gradient-Based Search
- SMT
- Probabilistic Programming

**Datalog**
- Abstract Interpretation

**Systems**
- synet.ethz.ch
- Bayonet
- apk-deguard.com
- psisolver.org
- Batfish
- Spire
- securify.ch
- jsnice.org
- Debin
- ai2.ethz.ch

**Applications**
- Programmable Networks
- Blockchain Security
- Attacks and Defenses on Deep Learning
- Probabilistic Security
Course Breakdown

- Programmable Networks
- Blockchain Security
- Attacks and Defenses of Deep Learning
- Probabilistic Security
Attacks on Machine Learning...

Slight Street Sign Modifications Can Completely Fool Machine Learning Algorithms

By Evan Ackerman
Posted 4 Aug 2017 | 18:00 GMT

How well can we get along with machines that are unpredictable and inscrutable?

The Dark Secret at the Heart of AI

No one really knows how the most advanced algorithms do what they do. That could be a problem.

AI programs exhibit racial and gender biases, research reveals

Machine learning algorithms are picking up deeply ingrained race and gender prejudices encoded within the patterns of language use, scientists say.

European Union regulations on algorithmic decision-making and a "right to explanation"

Bryce Goodman, Seth Flaxman
(Submitted on 26 Jun 2018 (v1), last revised 31 Aug 2018 (this version, v2))

We summarize the potential impact that the European Union's new General Data Protection Regulation will have on the use of machine learning algorithms, and the effort to link them to specific users. This law will also effectively create a "right to explanation," whereby a user can ask for an explanation of an algorithmic decision that was made that while this law will pose large challenges for industry, it highlights opportunities for computer scientists to take the lead in designing algorithms and ensuring that they are fair, transparent and accountable.

Comments: presented at 2018 ICML Workshop on Human Interpretability in Machine Learning (HIL 2018), New York, NY
Subjects: Machine Learning (stat.ML); Computers and Society (cs.CY); Learning (cs.LG)
Cite as: arXiv:1806.03412 [stat.ML]
Fooling Self-Driving Cars

Deep Neural networks are *not* robust to input perturbations (e.g., image rotation / change of lighting)

Misclassifications in neural networks deployed in self-driving cars [1]
In each picture one of the 3 networks makes a mistake...

Deep Learning is not Robust to Noise

Source: Explaining and Harnessing Adversarial Examples. Goodfellow, Shlens, Szegedy, ICLR ’15
What are we going to do about it?

In lecture 7, we will survey the necessarily basics of deep learning and show how and why adversarial examples arise.

In lecture 8, we will study different ways for attacking and defending deep learning models, in particular, differentiable search and symbolic methods for finding adversarial examples (e.g., FGSM and Madry’s defense).

In lecture 9, we will study approaches which prove the network is free of adversarial examples in various regions of the input! We will cover the AI2 system (http://ai.ethz.ch) (AI for AI) and methods that use symbolic methods during training (e.g., further defending the network).

These lectures explain the formal reasoning techniques (e.g., Gradient-based search, Abstract Interpretation) and outline some of the most exciting ongoing research in the area, in particular the combination of continuous optimization and symbolic reasoning, often referred to as the 3rd wave of AI.
What will we learn in this part?

Core Methods
- Probabilistic Graphical Models
- Datalog
- Abstract Interpretation
- Synthesis
- SMT
- Gradient-Based Search
- Probabilistic Programming

Systems
- synet.ethz.ch
- Bayonet
- apk-deguard.com
- psisolver.org
- Batfish
- securify.ch
- Spire
- Debin
- jsnice.org
- ai2.ethz.ch

Applications
- Programmable Networks
- Attacks and Defenses on Deep Learning
- Blockchain Security
- Probabilistic Security
Course Breakdown

- Programmable Networks
- Blockchain Security
- Attacks and Defenses of Deep Learning
- Probabilistic Security
Public output reveals information about confidential input. We want to restrict the amount of revealed information in some way.
Example: Genomic Data

Each patient has a pair of a red or green gene
Carol is a child of Alice and Bob
Example: Genomic Data

Each patient has a pair of a red or green gene. Carol is a child of Alice and Bob.
Example: Genomic Data

Each patient has a pair of a red or green gene.
Carol is a child of Alice and Bob.

Eve (medical researcher)
Example: Censor Bob’s data

Input (confidential)

Query (processes data)

Output (public)
Example: Censor Bob’s data

Can Eve learn anything about Bob’s genes?
Example: Censor Bob’s data
Example: Censor Bob’s data

Carol inherits her genes from Alice and Bob.
Example: Censor Bob’s data

Carol inherits her genes from Alice and Bob.
Input $c$ (confidential)

Query (processes data)

Output $o$ (public)

Bayesian Inference using $P(I = i)$ and the query

Attacker prior $P(I = i)$

Revised Attacker Belief $P(I = i | O = o)$
What are we going to do about it?

In lecture 10, we will show how to perform probabilistic privacy enforcement. The idea will be to leverage powerful probabilistic solvers (e.g., PSI) that can perform exact Bayesian inference and to then define synthesis mechanisms that add noise to the query such that we can guarantee a bound on the amount of information leakage. If time permits, we will also see how to leverage these solvers combined with gradient-based methods to perform analysis and synthesis for Differential Privacy.

In the process we will again see probabilistic programming (as we did for networks) but in a rather different context. We will also see how synthesis and gradient-based methods interact with probabilistic programming.

The lecture touches on probabilistic reasoning for security and outlines some of the open problems in the area.
Let’s look at another domain where security and machine learning interact.
What does this code do?

Minified JavaScript

source: bbc.com

Most JS code on the Web has undergone ‘layout obfuscation’, making it hard to read, understand and identify malicious activity.
function FZ(e, t) {
    var n = [];
    var r = e.length;
    var i = 0;
    for (; i < r; i += t)
        if (i + t < r)
            n.push(e.substring(i, i + t));
        else
            n.push(e.substring(i, r));
    return n;
}
What are we going to do about it?

In lecture 12, we will study some of the latest techniques on using advanced machine learning models (e.g., probabilistic graphical models) to reverse the process of layout de-obfuscation. We will show how these methods apply across a number of different languages (e.g., JavaScript, Android).

Concretely, we will illustrate the operation of JSNice.org and APK-DeGuard.com and the Nice2Predict.org systems and frameworks. These systems have > 500,000 users currently and are popular in the developer communities.

In the process we will outline some of the open problems in using machine learning for security (e.g., de-obfuscation, learning security rules) as well as the general area of learning probabilistic models from code (also, explored by startups such as http://deepcode.ai)
function FZ(e, t) {
    var n = [];
    var r = e.length;  var i = 0;
    for (; i < r; i += t) if (i + t < r) n.push(e.substring(i, i + t)); else n.push(e.substring(i, r));
    return n;
}

function chunkData(str, step) {
    var colNames = [];
    var len = str.length;
    var i = 0;
    for (; i < len; i += step)
        if (i + step < len)
            colNames.push(str.substring(i, i + step));
        else
            colNames.push(str.substring(i, len));
    return colNames;
}
JSNice.org

- Every country
- 200,000 users
- Top ranked tool

This Page Amsterdam @thispage_ams - Jul 16
Do you write ugly JavaScript code? Not to worry. JSNice will make it look like you are a superstar coder. Yai! - buff.ly/1HR4JL7

Ingvar Stepanyan @RReverser - Aug 6
JSNice.org became my must-have tool for code de-obfuscation.

Brevity @seekbrevity - Jul 28
JSNice is an amazing tool for de-minifying javascript files. JSNice.org, it's great for learning and reverse engineering.

Alvaro Sanchez @alvasvi - Jun 10
This is gold. Statistical renaming, Type inference and Obfuscation.

Alex Vanston @mvdot - Jun 7
I've been looking for this for years: JS NICE buff.ly/1pQ5qfr javascript #unminify #deobfuscate #makeItReadable

Kamil Tomšík @cztomsik - Jun 6
tell me how this works!
de-minify #jquery #javascript incl. args, vars & #jsdoc impressive! jsnice.org
package a.b.c
class a extends SQLiteHelper {
    SQLiteDatabase b;  public
    a(Context ctx) {   b =
        getWritableDatabase();  }
    Cursor c(String str) {
        return b.rawQuery(str); }}

dultudeb
package com.example.dbhelper
class DBHelper extends
SQLiteHelper {
    SQLiteDatabase db;
    public DBHelper(Context ctx) {
        db = getWritableDatabase();
    }
    Cursor execSQL(String str) {
        return db.rawQuery(str); }
    }

Funny Reddit post/comment

[-] Tycon712 • 3 points 2 days ago
Can someone tell me what the point of using Proguard is if there are tools out there like this?
permalink  embed  pocket

[-] theheartbreakpug • 6 points 2 days ago
As far as I know, this is brand new. I asked the creator of ProGuard a week ago how hard it is to unobfuscate code after it's run through proguard. He said it strips all the names out of the code so it's essentially impossible. I'm super impressed by what they've done here.
## What will we learn in this part?

### Core Methods
- **Probabilistic Graphical Models**
  - Datalog
  - Abstract Interpretation
- **Synthesis**
  - SMT
- **Gradient-Based Search**
  - Probabilistic Programming

### Systems
- **synet.ethz.ch**
- **psisolver.org**
- **jsnice.org**
- **Bayonet**
- **Batfish**
- **Spire**
- **apk-deguard.com**
- **securify.ch**
- **Debin**
- **ai2.ethz.ch**

### Applications
- **Programmable Networks**
- **Blockchain Security**
- **Attacks and Defenses on Deep Learning**
- **Probabilistic Security**
Today

Overview of Material

Course organization
Learning Objectives

Understand core, widely applicable methods in depth

Understand how to apply these by building practical systems.

Understand the open research problems in the area.

Key objective: 686176652066756e21 😊
Should I take this course?

You will enjoy this course if you:
– agree with the objectives
– enjoy learning about core concepts through application
– like the intersection of theory and systems
– like to learn about open research problems

Do not take this course if you:
– want to focus on pure theory only or only pure hacking
– do not agree with some of the objectives
– do not want to work during the semester (e.g., homework, project)
More lectures from:

Dr. Petar Tsankov
Dr. Dana Drachsler-Cohen
Dr. Veselin Raychev (CTO, deepcode.ai)
honorable mention winner
Timon Gehr
Main author of PSI
(http://psisolver.org/)

Course web site: http://www.srl.inf.ethz.ch/pass.php

All information posted there: lectures notes, exercises, etc.
This Course

Prerequisites

- Course is self-contained
- 5 credits: 2h lecture + 1h exercises.

Grading

- 50% final exam (make sure you do the homework)
- 50% project: ideally done in pairs.
- If you have an idea for a different project, let us know.
Related Courses

Seminar: Blockchain Security

Seminar: Deep Learning for Big Code
(http://www.srl.inf.ethz.ch/bigcode18.php)

Programmable Networks (at ITET – relates to Part I of this course)
(https://adv-net.ethz.ch/)

Research: Research in Computer Science (5 credits)
(for those interested in the topics and want to explore.
contact me if interested)
Exercises

Come with questions/solutions
   The TA will go over the homework

Exercises are not graded, but if you do them, you will have an advantage on the final exam
Next Lecture:

Introduction to Datalog and SMT Application to Network analysis

Dr. Petar Tsankov