基于大数据的软件安全

Big Data Analysis for Software Security

授课教师：　陈　恺

助　　教：　赵　月

2018-2019学年秋季学期

第1次课

一. 课程介绍

二. 第一章

授课教师：陈恺

授课时间：2018-9-11

授课地点：国科大雁栖湖校区教1-223

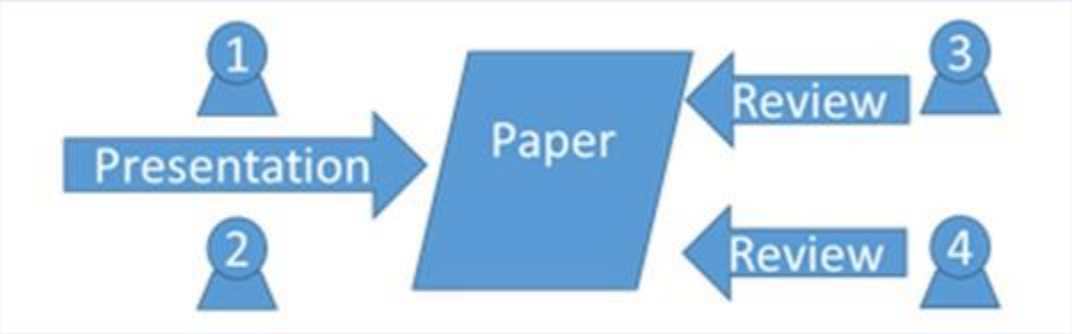
预期收获

- 能学到什么?
  - 软件安全基本问题和解决办法
  - 大数据分析思想和应用
  - 了解国际前沿的研究并站在前沿
  - 如何做好高水平的科研工作

课程介绍

课程组织形式

- 课堂分组（共9组）：2位同学为一个组，每个小组将分别对2篇论文进行报告和评审
- 规则如下：



- 每周由一个小组进行报告，第二个小组进行评审，其他每个分组讨论并提出问题，不同于其他分组提出的问题
- 当周报告/评论的小组需要在当周周一晚上11:59前将各自的review发至助教邮箱(zhaoyue@iie.ac.cn)

如何汇报与评审

- 汇报：
  - 要把思想讲清楚，尤其是为什么作者要这么去做这么一件事情；回答评审及其他人提到的问题
  - 不要简单罗列论文中的步骤
  - 不要过于复杂，例如讲一系列公式等
  - 不要用网上的教学素材（如播放视频）
- 评审：
  - 优点在哪里?
  - 缺点在哪儿？不一定是作者在论文discussion里面写到的不足。每个reviewer提出4个以上“批判性”问题

课程介绍

课程组织形式

- Presentation （35%）
  - 每位同学参与1次报告 + 回答问题
- Review （35%）
  - 每位同学提交1次评审
- Class Participant （20%）
  - 每位同学的课堂参与度（提问+讨论）
- Attendance （10%）
  - 缺一次课扣1分，所有课缺勤扣10分

基于大数据的软件安全
Big Data Analysis for Software Security

1. 什么是软件安全?
2. 和大数据有什么关系?
3. 《基于大数据的软件安全》讲什么?

基于大数据的软件安全

Big Data Analysis for Software Security

第一章 大数据与软件安全概述

第一章大数据与软件安全概述

本章内容

1.1 大数据与软件安全简介

- 1.1.1 软件安全简介
- 1.1.2 大数据简介
- 1.1.3 软件安全与大数据的关系

1.2 软件分析中的大数据问题及解决思路简介

软件安全简介

有哪些你碰到过的软件安全案例?

软件安全简介

软件安全漏洞

- 伊朗核设施，近20%的离心机因此报废（离心机在失控情况下不断加速而最终损毁）
- 手机被悄悄的Root、支付等
- Wannacry（魔窟）

软件安全简介

恶意代码

- 个人电脑：远程控制、窃取信息
- 手机：金融支付、偷拍照片、录像/语音、跟踪定位等
- 智能家居：控制家里的防盗装置、摄像头等

软件安全简介

- 软件安全分析并不容易
  - 程序分析过程包含大量路径
  - Fuzz方法，大量的待测试输入
  - ……

软件安全简介

- 软件安全分析并不容易
  - 程序分析过程包含大量路径
  - Fuzz方法，大量的待测试输入
  - ……

能否以大数据的思路进行分析、解决问题?

第一章大数据与软件安全概述

本章内容

1.1 大数据与软件安全简介

- 1.1.1 软件安全简介
- 1.1.2 大数据简介
- 1.1.3 软件安全与大数据的关系

1.2 软件分析中的大数据问题及解决思路简介

大数据的应用

- 大数据的应用
  - 大数据的应用在包括大科学、RFID、感测设备网络、天文学、大气学、交通运输、基因组学、生物学、大社会数据分析、互联网文件处理、制作互联网搜索引擎索引、通信记录明细、军事侦查、社交网络、通勤时间预测、医疗记录、照片图像和视频封存、大规模的电子商务等。

图 大数据应用于运动界

1.1.2 大数据与软件安全简介

大数据的应用

- 苏轼诞生980周年时，清华附小开展致苏轼的活动
- 临摹苏轼的字画等活动

1.1.2 大数据与软件安全简介

大数据的应用

- 苏轼诞生980周年时，清华附小开展致苏轼的活动
- 临摹苏轼的字画等活动
- 大数据分析苏轼

大数据分析帮你进一步认识苏轼

官天泽、徐子昂、王储玉、马梓铭、葛宇轩

上学期我用大数据的方法写了一首《如梦令》，这次我们小组研究苏轼，我们也想再用大数据的方法对苏轼的诗词进行进一步的分析。

一、数据证明苏轼是名高产作家

首先我和徐子昂把苏轼的 3458 首诗词都找了出来，大概有 25 万字。我们发现唐宋诗词由 9552 位作者创作了 276545 首诗词，平均下来每位作者要完成 28-29 首诗词的创作，而苏轼一个人就相当于 120 位诗人，占了整个唐宋诗词量的 1.25%。苏轼一共活了 66 岁（其实按照今天计算方法来算，他只活了 64 年，古代出生的时候就算 1 岁，他是元丰五年十二月十九日生，过了年又算 1 岁，所以在他出生半个月的时候就 2 岁了），我们按照他实际年龄来计算，他每年需要写 54 首诗词，这样下来平均每周至少写一首诗词。这些仅仅是他的诗词，不包括散文、札记、书信等等。

二、通过数据看苏轼的人生经历

1、我们的方法

我和爸爸通过一段程序把苏轼的 3458 首诗词进行了分词研究，找出了这些诗词中的高频词。

排名前 50 的高频词表如下：

子由	归来	使君	不见	故人	平生	人间	何处	无人	万里
229	157	152	148	135	130	123	122	119	109
东坡	何时	明月	归去	西湖	白发	青山	江南	草木	惟有
108	101	100	92	92	90	85	84	83	83
山中	风流	东风	不须	江湖	春风	可怜	明年	新诗	梅花
80	78	75	73	73	72	70	70	68	66
风雨	当时	当年	佳人	闻道	清风	仙都	道人	南山	太守
66	66	63	62	61	61	60	60	59	57
饮酒	秋风	去年	黄州	公子	少年	回首	诗云	归路	何曾
57	56	56	54	54	54	53	52	52	51

注：每一个词下面的数字代表在它在苏轼诗词里面出现的次数

由于汉语里有很多一个字的词，这些词也需要考虑，于是我们把所有的高频字也做了分析。

排名前 50 的高频字表如下：

不	人	一	有	山	无	我	子	来	君
3410	2891	2183	2057	2041	1842	1732	1593	1571	1568
风	何	中	生	如	此	为	诗	自	日
1417	1428	1384	1379	1358	1346	1331	1261	1261	1251
见	云	与	老	知	天	归	月	二	时
1207	1206	1193	1175	1120	1094	1063	1059	1056	1046
首	水	花	上	未	之	相	已	长	春
1031	1020	1006	963	955	937	932	905	888	879
清	得	江	二	谁	可	空	千	十	南
879	874	866	846	826	820	817	802	798	792

注：每一个字下面的数字代表在它在苏轼诗词里面出现的次数

2、我们的问题

这些高频词和字分析出来之后，我们产生了很多疑惑。比如：

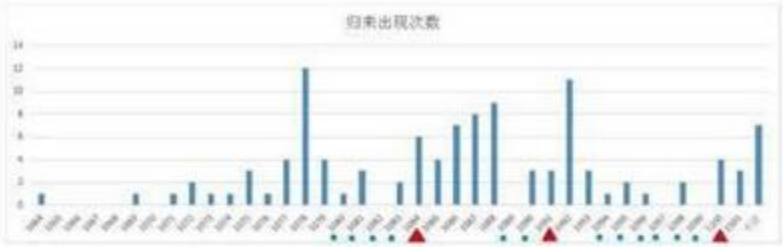
——“归来”这个词竟然出现了 157 次，是苏轼诗词里面用得最多的一个词（注：第一次分析高频词时，还没有搜索“子由”，因此，排在第一位的词汇是“归来”），“归去”出现 92 次，苏轼是在到处云游吗？

——苏轼经常提到“故人”，出现了 135 次，还有“道人”60 次，这些人都是指得谁呀？他是不是有很多和尚、道士朋友呀？

——苏轼诗词里面提到“西湖”92 次，“江南”84 次，这些诗词是否都是他

代。但由于时间的原因，我们没有来得及对苏轼所有包含“归来”词做查找。因此，下面的分析是基于 111 首包含“归来”诗的统计结果。

这是苏轼一生中“归来”在诗中出现的次数分布图：



我们查找了苏轼三次被谪的经历，即第一次（1080-1084），因为“乌台诗案”遭到新党诬陷，被谪黄州。第二次（1089-1091），因为不同意司马光尽废新法，被谪杭州、颍州。第三次（1094-1101），因为与章惇政见不合，被谪惠州、儋州。把这些时间节点都标注到图中，其中蓝色点的区域是被谪的时间，红色的三角是每次被谪结束的年份。

我们发现，每次被谪结束之后，苏轼诗中的“归来”出现的次数都会有所增加，苏轼这些“归来”诗，与他跌宕起伏的一生似乎存在着联系，他一直满怀忧国之情，总能将这些归去归来的经历，化作美好的文学意境。

数据分析的结果印证了我们的猜想，让我们从一个新的角度认识这位文学巨匠。苏轼一生忧患重重，多次被贬，正是这些苦难的经历和丰富的阅历，使苏轼更关心民间疾苦，更亲近大自然，使他的作品成为传世的杰作。时至今日，我们读苏轼的诗词，仍然能感到无限的哀怨和悲凉，更能体味到中国文化的深厚底蕴和幽香。

三、额外的发现

在研究过程中，我们还发现“子由”出现在很多诗词中，“子由”是苏轼弟弟的字，这使我们想到应该检索一下“子由”在苏轼作品中出现的次数。于是，我们重新检索了一下，发现“子由”在《苏轼诗词全集》居然出现了 229 次，它才是苏轼高频词里面的王者！为此，我们更新了高频词表。

我问爸爸为什么第一次做的高频词表中没有搜到“子由”，爸爸给我们解释

229	157	152	148	135	130	123	122	119	109
东坡	何时	明月	归去	西湖	白发	青山	江南	草木	惟有
108	101	100	92	92	90	85	84	83	83
山中	风流	东风	不须	江湖	春风	可怜	明年	新诗	梅花
80	78	75	73	73	72	70	70	68	66
风雨	当时	当年	佳人	闻道	清风	倚栏	道人	南山	太守
66	66	63	62	61	61	60	60	59	57

说，“子由”在汉语中不是一个词汇，因此电脑软件第一次在做分词的时候，并没有对它进行检索。分词是一门很深的学问，每一部著作都有自己的特点，对每部著作的分析是一个不断发现的过程。今天看着正确的分析结论，可能随着研究的深入就不一定正确了，比如这次。

这次对高频词表的更新，让我们在研究主题之外，有了额外的收获：就是我们发现了原来苏轼和他的弟弟子由之间手足情深。通过上网进一步查资料，我们了解到苏轼几乎每到一个任所就给弟弟子由寄信赠诗，晚年被贬谪时更是如此。苏家兄弟情谊之深厚是文学史上的佳话。他们是兄弟、是师生、是诗词唱和的良友、是政治上荣辱与共的伙伴、是精神上相互勉励安慰的知己。我们的高频词表也进一步印证了他俩之间的情谊，我们可以得出结论，研究表明苏轼还是一个好哥哥。

这些高频词和字分析出来之后，我们产生了很多疑惑。比如：

——“归来”这个词竟然出现了 157 次，是苏轼诗词里面用得最多的一个词（注：第一次分析高频词时，还没有搜索“子由”，因此，排在第一位的词汇是“归来”），“归去”出现 92 次，苏轼是在到处云游吗？

——苏轼经常提到“故人”，出现了 135 次，还有“道人”60 次，这些人都是指得谁呀？他是不是有很多和尚、道士朋友呀？

——苏轼诗词里面提到“西湖”92 次，“江南”84 次，这些诗词是否都是他

大数据简介

哪些技术可能与我们相关?

第一章大数据与软件安全概述

本章内容

1.1 大数据与软件安全简介

- 1.1.1 软件安全简介
- 1.1.2 大数据简介
- 1.1.3 软件安全与大数据的关系

1.2 软件分析中的大数据问题及解决思路简介

1.1.3 软件安全与大数据的关系

本节内容

- 传统的检测与防御机制力不从心
  - 在大数据时代，面对不断变化的海量软件的安全需求，传统的检测与防御机制已经显得力不从心，尤其在数量上和时间上已经不能满足现在的需求。如，Google Play上有200万以上软件，如何对其安全性进行测评?

1.1.3软件安全与大数据的关系

本节内容

- 结合软件安全与大数据的需求迫切
  - 在大数据时代，若想更好的服务于软件安全，必然要结合或者利用大数据技术，做到软件的海量处理、海量检测、快速应答等。才能更好的应对大数据带来的安全挑战。

本章内容

1.1 大数据与软件安全简介

- 1.1.1 软件安全简介
- 1.1.2 大数据简介
- 1.1.3 软件安全与大数据的关系

1.2 软件分析中的大数据问题及解决思路简介

1.2 软件分析中的大数据问题及解决思路

本节内容

- 软件重打包/剽窃检测
- 恶意代码检测
- 高效漏洞检测
- 跨平台软件分析
- 其他问题（软件混淆与反混淆、软件反汇编/反编译, ……）

1.2 软件分析中的大数据问题及解决思路

软件重打包/剽窃检测

- 什么是软件重打包/剽窃?
  - 攻击者通过在重打包软件中植入恶意代码并进行传播。例如，Android开发中流传着所谓的"打包党"，即将其他开发者开发的畅销应用解开后修改部分内容（比如广告ID等），然后重新进行打包上传到商店以谋求暴利。
  - 重打包与剽窃是典型的软件安全问题。对其检测不仅涉及百万数量软件的分析，且需对万亿量级的代码进行相互比较。

1. 完整拷贝
2. 更改注释
3. 更改空格，重排版
4. 标识符重命名
5. 代码段更换顺序
6. 改变代码段中语句顺序
7. 改变表达式中的操作符顺序
8. 改变数据类型
9. 增加冗余语句和变量
10. 替换控制结构为等价的控制结构

剽窃变换列表

基于文本相似性的源代码级同源性度量技术

1.2 软件分析中的大数据问题及解决思路

软件重打包/剽窃检测

- 解决思路
  - 签名检查
  - 同源性分析
  - 源代码级的同源性分析
    - 基于文本相似性
    - 基于编程风格分析

1.2 软件分析中的大数据问题及解决思路

恶意代码检测

- 恶意代码
  - 在大数据时代，随着用户数量剧增，各种各样的软件也越来越多，充斥在各大市场的各类软件的质量也良莠不齐，攻击者趁机作乱，在软件中加入各种恶意代码后将软件大量传播，给用户带来巨大损失。
  - 恶意代码的庞大数量和复杂性，也给恶意代码的检测带来巨大的挑战。

1.2 软件分析中的大数据问题及解决思路

恶意代码检测

- 解决思路
  - 基于行为特征的分析
  - 基于代码特征的分析
  - 基于代码比对的分析
  - 基于机器学习的分析

1.2 软件分析中的大数据问题及解决思路

漏洞检测

- 软件漏洞
  - 软件漏洞是软件生命周期中涉及安全的设计错误、编码缺陷和运行故障。
  - 将对软件进行已公开漏洞的发现和未知漏洞的发掘的技术称为软件漏洞检测技术。
  - 将分析漏洞形成原因及利用价值的技术称为漏洞分析技术。
  - 将利用漏洞实施攻击的技术称为漏洞利用技术。

1.2 软件分析中的大数据问题及解决思路

漏洞检测

- 软件漏洞检测难点
  - 在大数据浪潮下，移动终端应用软件漏洞数量和类别飞速增长。如何快速且高效的检测大量软件中的漏洞成为效率难题。
  - 各类系统及应用软件的漏洞种类也在不断增多，各种各样的不同种类漏洞给软件漏洞的甄别及检测工作带来麻烦，如何提高漏洞的可识别种类也成为难题。

1.2 软件分析中的大数据问题及解决思路

漏洞检测

- 软件漏洞检测思路
  - 淘金式——“挖洞”
    - 白盒测试
    - 黑盒测试
  - “举一反三”式——找相似漏洞
    - 补丁比对
    - 同源性分析

1.2 软件分析中的大数据问题及解决思路

跨平台软件分析

- 什么是跨平台软件分析
  - 从PC时代到移动时代，跨平台软件迅速增加。往往同一款软件同时具有电脑客户端、Android客户端和IOS客户端等不同平台的客户端，于是，同一软件在其中某一平台出现的安全问题在其他平台客户端也往往会存在。于是对各种跨平台软件进行统一分析也成了软件分析过程中的必要措施。

1.2 软件分析中的大数据问题及解决思路

跨平台软件分析

- 当跨平台软件分析遇上大数据
  - 当跨平台软件分析遇上大数据，难度也在成倍增加。在大数据时代，对跨平台软件分析这类安全问题的检测也就变成需要同时处理海量数据与跨平台两种特性。

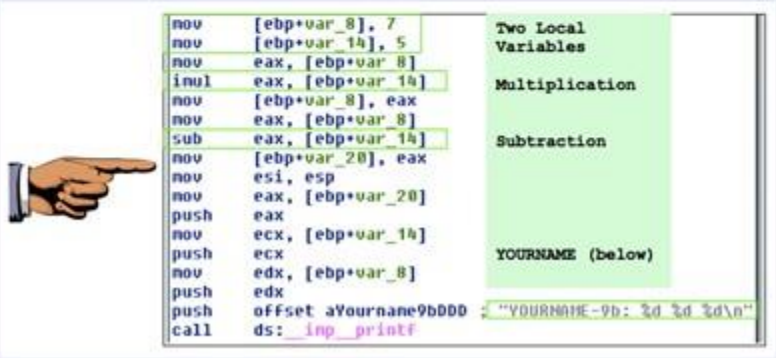
1.2 软件分析中的大数据问题及解决思路

软件混淆与反混淆

- 代码混淆（Obfuscated code）
  - 将计算机程序的代码，转换成一种功能上等价，但是难于阅读和理解的形式的行为。
  - 代码混淆可以用于程序源代码，也可以用于程序编译而成的中间代码。执行代码混淆的程序被称作代码混淆器。
  - 目前已经存在许多种功能各异的代码混淆器。
- 反混淆
  - 相对于代码混淆，同时也有代码的反混淆，使用反混淆工具，将混淆后的代码还原成可供阅读的代码。

反汇编

- 反汇编(Disassembly)
  - 把目标代码转为汇编代码的过程（也可以说是把机器语言转换为汇编语言代码、低级转高级）
  - 常用于软件破解（例如找到它是如何注册的，从而解出它的注册码或者编写注册机）、外挂技术、病毒分析、逆向工程、软件汉化等领域。



1.2 软件分析中的大数据问题及解决思路

反编译

- 反编译
  - 反编译的功能与编译相反。可以进行反编译的工具叫做反编译器，顾名思义，就是将已编译好的编程语言还原到未编译的状态，也就是找出程序语言的源代码。
  - 一种反编译器通常只能反编译1~2种编程语言，反编译器的功能只局限在某些语言上，如Java，像C/C++便没有适合的反编译器可使用。

1.2 软件分析中的大数据问题及解决思路

研究与讨论

一. 大数据与软件安全概述/1.2 软件分析中的大数据问题及解决思路简介

Q & A

Paper List (1)

- Group 1
  1. Neural Nets Can Learn Function Type Signatures From Binaries (CCS'17)
  2. Recognizing Functions in Binaries with Neural Networks (USENIX Security'15)
- Group 2
  1. Transcend: Detecting Concept Drift in Malware Classification Models (USENIX Security'17)
  2. Catching Worms, Trojan Horses and PUPs: Unsupervised Detection of Silent Delivery Campaigns (NDSS'17)
- Group 3
  1. HinDroid: An Intelligent Android Malware Detection System Based on Structured Heterogeneous Information Network (KDD'17)
  2. Gotcha - Sly Malware!: Scorpion A Metagraph2vec Based Malware Detection System. (KDD'18)
- Group 4
  1. SemFuzz: Semantics-based Automatic Generation of Proof-of-Concept Exploits (CCS'17)
  2. APISAN-Sanitizing API Usages through Semantic Cross-checking (USENIX Security'16)
- Group 5
  1. Skyfire: Data-Driven Seed Generation for Fuzzing (S&P'17)
  2. VulDeePecker: A Deep Learning-Based System for Vulnerability Detection (NDSS'18)

Paper List (2)

- Group 6
  1. Predicting the Resilience of Obfuscated Code Against Symbolic Execution Attacks via Machine Learning. (USENIX Security'17)
  2. Syntia: Synthesizing the Semantics of Obfuscated Code (USENIS Security'17)
- Group 7
  1. FeatureSmith: Automatically Engineering Features for Malware Detection by Mining the Security Literature (CCS'16)
  2. MaMaDroid: Detecting Android Malware by Building Markov Chains of Behavioral Models (NDSS'17)
- Group 8
  1. Malware Detection in Adversarial Settings: Exploiting Feature Evolutions and Confusions in Android Apps (ACSAC'18)
  2. SecureDroid: Enhancing Security of Machine Learning-based Detection against Adversarial Android Malware Attacks (ACSAC'18)
- Group 9
  1. Scalable Graph-based Bug Search for Firmware Images (CCS'16)
  2. Neural Network-based Graph Embedding for Cross-Platform Binary Code Similarity Detection (CCS'17)