wasp_SE_course/tree/main/assignments/2025/Lu_SiKai/Assignment2.md

Introduction

My research lies at the intersection of programming analysis and security. We utilize static analyzer uncover vulnerabilities that arise due to language design choices. Currently, i am studying the provenence of prototype pollution in the client side environment, mainly in the browser context.

Prototype pollution is a security vulnerability rooted in the fundamental design flaw of JavaScript. JavaScript supports object oriented programming. Modern object oriented programming languages support inheritence. Inheritence allows a child class object accesses value and functions defined in the parent class. In Javascript, either bracket notation or dot dot enable an object access its properties. However, if the requested property is undefined directly on the object, the javascript runtime will check if such field is defined in its parent classes, if exists then return the value. In Javascript, every class is a child class for the Prototype class, every object have access to the Prototype's field. This permits that an object can access properties defined in the prototype object. If the field doesn't exist in the class, the javascript runtime will go through the inheritence hierarchy and search the field until the prototype is reached. This design creates opportunities for exploitation. If an attacker modifies a propert in the prototype, any object that does not explicitly define this field will inherit the malicious value. As a result, prototype pollution can lead to unexpected and potentially dangerous program behavior.

Prototype pollution alone is not sufficient for a full exploitation; rather, it creates the conditions for potential attacks. Successful exploitation typically requires two additional components: (1) a property reference requests the value of an undefined property of an object, and (2) that value subsequently propagates into a malicious sink. Each of these components is worth studying in its own right. Prior research has approached these challenges through static analysis, dynamic analysis, the development of templates based on industrial and practical experience, and formal problem formulations, among other methods. My work focuses on applying static analysis and examining how its findings can enhance security studies.

This represents one perspective on my research.

Lecture Principles

The lecture mentioned verification and validation, and used both terms in an unfamiliar way. Verification is the process of checking of whether we are building the project right and validation is the process of checking whether we are building the right project. The definition of verification most familiar to me comes from program correctness: if a function is called when its preconditions hold, then after execution the program state should satisfy the corresponding postconditions. This idea is central to design by contract and program verification. I would say that the former definition of verification is a general abstraction of the latter. Program correctness is about how to build the project right.

Validation, in this context, is about checking whether we are building the right project. From a programmer's perspective, that responsibility is usually not ours. We are often the executors of someone else's idea, translating requirements into executable code. The burden of deciding what the "right project" is falls on the shoulders of the business. From a researcher's perspective, validation can be rephrased as checking whether we are studying a question worth solving. This merits some considerations. I would that questions i am studying have practical utilities and solving these questions helping me gain insights in programming language theory.

I would also like to emphasize the importance of static analysis. This is not just because it's my area of work, but because static analysis can detect software deficiencies before deployment. It is often considered a lightweight form of verification, and it falls under the principle of building the project right. I believe that static analyzers can go beyond simple code improvements or linting for ML code. I'm not a machine learning researcher, so I may be phrasing this imperfectly, but consider an example from image processing: a kernel is applied by sliding it across the input image. If a program mistakenly provides a kernel of the wrong size to a layer expecting a 3×3 kernel, the operation is undefined and results in a dimension mismatch error. Because Python is dynamically typed, such errors can only be caught at runtime. This is precisely where static analysis could add value: a type system could track kernel sizes and reject programs that contain mismatched kernels at compile time.

Guest Lecture Principles

In the guest lecture, Goal Modeling was introduced. In the context of

security research, this idea closely relates to the concept of a threat model. We often define it when writing a paper. A threat model specifies the type of vulnerability under consideration, the preconditions or environment that allow an attacker to exploit it, and the definition of potential sinks. Importantly, even for the same vulnerability, different environments/preconditions form different threat models. In a research paper, it is important to clearly specify the threat model, since an exploitation that is valid under one threat model may not be applicable under another, or it may result in less severe consequences.

I find it difficult to see a strong connection between levels of abstraction and my research topic. The one link I can identify is that, in security analysis, the implementation details of a static analyzer are often irrelevant. As long as the analyzer supports taint flow analysis, it is sufficient for most security research—the internal mechanics are abstracted away. However, I often feel the need to break this abstraction. The analyzer's capabilities sometimes fall short of my expectations. Issues, that are obvious to the human inspecation, may not be detected by the underlying mechanism. This could stem from performance trade-offs, usability constraints, or other limitations. In such cases, I end up delving into the source code and even rebuilding parts of the analyzer myself. I suppose this break through of the abstraction barrier is a part of research.

Data Scientists verus Software Engineers

Do you agree on the essential difference between them?

I agree that there are essential differences between data scientists and software engineers as described in the book. My background is in computer science, from Bachelor's through Master's. From my observations, securing a PhD position in AI often requires publishing conference papers even before starting the doctoral program. It seems highly unlikely for someone trained primarily in software engineering to transition into data science. While it is possible, such individuals are rare and could be considered "unicorns," as described in the book.

The portrayal of a data scientist in that sense is quite accurate. They may use version control, but often in a limited way. Their scripts are not always reproducible, as they do not often know the version of the

packages their model depends on. Their approaches can be prohibitively expensive, since models require enormous computational resources to run.

Ultimately, the two fields require very different forms of training and expertise.

Do you think there roles will evolve and specialise further or that “both sides” will need to learn many of the skills of the other side?

From the Chapter Two reading, I would say that while the roles of data scientists and software engineers may overlap slightly, they do not fully coincide. Each inevitably acquires some skills from the other, but their primary responsibilities remain distinct. This division of roles creates the need for MLOps. Similar to DevOps, MLOps integrates machine learning components into a CI/CD pipeline, greatly reducing the complexity of deploying and updating models. The chapter also suggests additional roles, such as designing effective UI/UX for AI components and addressing security concerns that arise in AI systems.

The most interesting concept freshes my eye is that how to tame the AI model such that it might still occasionally burn some toast, but it will not burn down the kitchen. This definitely requires further researching and i look forward to see this.

Paper Reading

Themes of Building LLM-based Applications for Production: A Practitioner’s View

Core ideas and their SE importance

This paper is a survey paper. It reaches out to industrail people for their opinion on the chanllengesof developing LLM based applications. The paper identifies 20 chanllenges and argue that they constitute to 8 main themes.

A majority of people are concerning the effect of the RAG system. From my understanding, it behaves like a database query before LLM. It

fetches relevant information, feed those information into the prompt, and then LLM generates a response. They argue that RAG allows the LLM to have access to up-to-date information and reduces hallucinations. They also dig into the detail of optimization in RAG, but those beyond my expertise.

The shortage of high-quality data remains a problem. Fei-Fei Li emphasized the importance of good training data a decade ago, and yet the issue persists. or maybe the demand for high-quality data keeps growing faster than supply.

I would say this paper is quite a systematic overview of challenges in developing LLM based applications.

Relation to My research

The paper isn't directly related to my research, but it works well as an introduction to LLMs. Reading it helped me get a handle on the key technical terms. I'd heard "RAG system" many times without really grasping it; this is the first time it clicked. Because the paper presents the topics at a high level, it's easy for me, an LLM beginner, to follow along.

Integration into a larger AI-Intensive Project

Chatgpt is the most well-known AI Intensive Project. The name nowadays speaks for itself. I would say that the project itself must work through most of the chanllenges. Some models of chatgpt have the ability of accessing the internet. I assume these models must have some RAG systems or variants to allow this. OpenAI people must exhaust all optimization techniques to reduce their running costs. My research project highly depends on Chatgpt. I constantly reach chatgpt for help, python script generation, word rephrasing, explanations of technical terms, etc.

Adaption to your research

Right now, i don't have any plans to delve into the wonderous world of AI researching. However, the paper is informative and suggests some few good and useful prompts. I plan to use those in my queries.

On Device or Remote? On the Energy Efficiency of Fetching LLM-Generated Content

Core ideas and their SE importance

This paper examines whether it’s more energy-efficient to generate text locally on a device or fetch it from a remote LLM server. In the remote case, most of the energy is spent on HTTP transmission (sending and receiving data). The results are unsurprising: fetching text from a remote server uses at least three times less energy than generating it on-device. The experiments also show that on-device energy use increases roughly linearly with the model’s runtime. For local LLMs, energy consumption is strongly positively correlated with GPU utilization when generating larger amounts of text(over 500 words), but negatively correlated for smaller outputs. The paper further confirms that LLM inference is memory-intensive: regardless of output length, energy closely tracks memory usage. According to Figure 3, CPU utilization stays below 10%, higher CPU usage has positive correlation with total energy use in this context can not be derived from the data.

This paper is interesting because its experimental results confirm our intuition about the link between energy consumption and LLM text generation: longer processing times lead to higher energy usage. These findings highlight the direction we should focus on if we want to build a more energy-efficient and sustainable society.

Relation to My research

Not really related to my research. However, it is indeed an interesting read.

Integration into a larger AI-Intensive Project

I would say this paper is relevant to all AI-intensive projects. Such projects require enormous computational power, primarily from large numbers of GPUs. GPUs consume significant amounts of energy, and energy costs translate directly into financial costs. For any project to succeed, budgeting for these expenses is essential and cannot be overlooked.

Adpation to my own research

I would ChatGPT more stupid questions with a restrction on the number of token output. From the data, we can derive that stupid questions are not that expensive as the execution time is small.

Research Ethics & Synthesis Reflection

Search and Screen Process

Because my background is not in any AI related, the scope of suitable CAIN papers shrinks significantly. I search for papers with less esoteric words in the title. I rule out `RAGProbe: Breaking RAG Pipelines with Evaluation Scenarios` immediately as i don't have any knowledge in RAG and schenarios study clearly requires expertises to follow. I chose papers presented in Question 5 because they are pratical and answer a direct question.

Pitfalls and mitigations

I didn't find any misleading titles or abstracts. Since I have no prior experience in this area, I wasn't able to identify such papers. I also didn't adjust my screening process, as the papers I selected directly addressed the questions I had in mind when I first read their titles.

Ethical Consideration

I read the lecture slides and papers manually without using an LLM to generate summaries. For each question, I first wrote my answers in this markdown file. Then, I fed them to ChatGPT with the prompt: "Make the following paragraph more clear, easy to understand, and without grammar error." Afterward, I reviewed the generated output and made modifications whenever ChatGPT altered the meaning I wanted to convey.