Research in ML and Network Security

Srikanth V. Krishnamurthy
Department of Computer Science and Engineering
University of California, Riverside

UC RIVERSIDE

General research interests

1. Network security:
- What types of network wide attacks are viable ? /* Recent work on DPI evasion, firewall evasion, DNS Exfiltration, Mesh vulnerabilities
- How to overcome those attacks ? /* Work on context aware DPI evasion detection, IoT safety, Intrusion detection at scale
- Need to integrate ML

2. ML and Security
- What types of vulnerabilities can be exploited in ML models ? How to prevent them?
- How can ML be effectively used for network and systems security ?
- Deployment of ML models at scale in networks – how ?

3. Software system security
- Working with other researchers on finding vulnerabilities in software, mitigation methods.

Big question: How to develop secure autonomous networked systems ?

In this talk …

- Primarily focus on the role of ML security and use of ML in security.
- Posters from students on specific projects.

Subversion of ML models

- Adversarial ML has recently received a lot of attention.
- Small perturbations that an adversary can add to inputs can cause the model to make wrong inferences.
- Primarily related to images/videos.

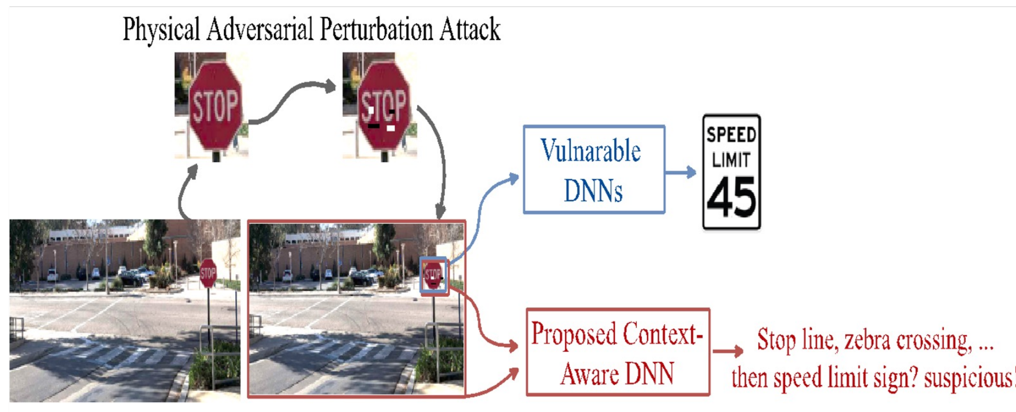
- How to protect inferences made in models deployed in nextG?
- Are over the air attacks possible ?
    - Key challenge seems to be synchronization
    - Still possible to cause wrong conclusions

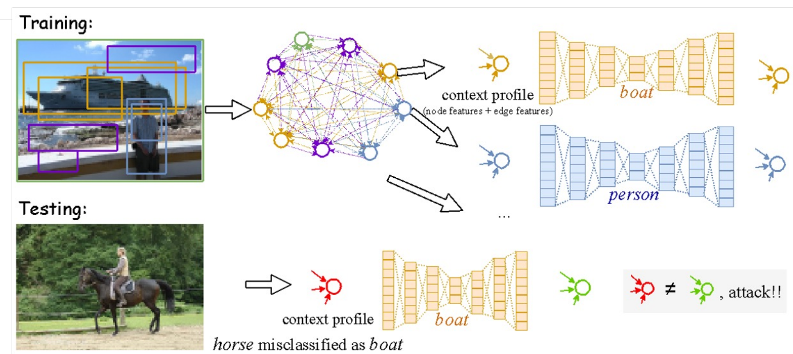
Can context help ?

- Account for relationships across features.
- For example, in the case of stop signs, there are likely to be crosswalks, etc. that co-occur.
- Similar relationships exist between packet fields, between temporal series of packets (TCP)

- Look for inconsistencies!

- However, it is possible for the attacker to launch context-aware attacks
  - Much more effective than traditional attacks on images/video
  - Whether context can help in protecting models used in networks/nextG is an open question.





Model poisoning is a real issue

- Training data likely to come from users
- The model may need to be refined online (dynamics make it hard to deploy pre-trained models that are static.

- Bots could target misclassifying models -- can be of significant impact
  - For example, autonomous cars may depend on inferences from such models (both from sensors on board and outside)
  - Compromise of any model can result in catastrophic effects.

- How to make sure that online data that is received is trustworthy and usable to refine/retrain models?

Using ML to help assess problems

- Program analysis can provide determinism – but is often either complex (e.g., model checking) or imprecise (e.g., static analysis has a lot of false positives).

- When to apply what ? Can ML guide the application of the correct technique (e.g., modularize interaction related code that requires certain types of safety guarantees) ?

- ORAN → code can become very complex – different vendors will contribute
  -- Understanding and protecting interactions between code is not a trivial exercise.
  -- New types of issues beyond what is extensively studied by the security community (e.g., deadlocks as opposed to out of bound memory access)
  -- Benefits of flexibility should not be undermined by security problems
  -- Need for principled ways to incorporate ML into testing (e.g., fuzzing) or to guide program analysis methods.

  -- Prioritize easy to find bugs for fixing (e.g., using automated patches), while deferring more complex (hard to find) bugs for offline analysis (isolation of components needing such analysis) ?

A glimpse into our network security work : C2Store

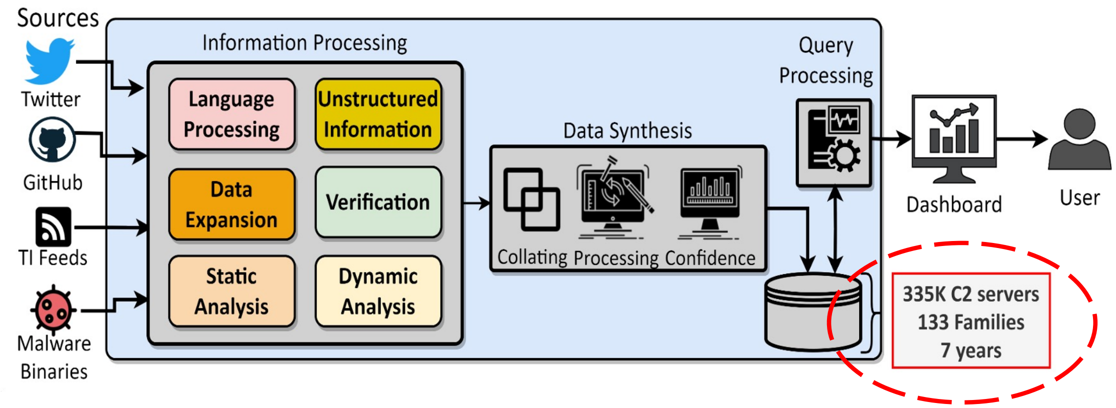
https://c2store.github.io/

Key Contributions

#1: Identify untapped sources (GitHub, Twitter)

#2: Develop methods to efficiently mine these untapped sources

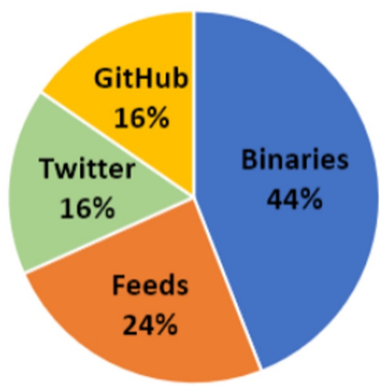
#3: Synthesize information to create C2 complete server profile



Scope and Evaluation

- **Ground truth:** malicious if 5+ AV engines in VirusTotal
- **Precision:**

  a) Twitter: 97% **b)** GitHub: 94%



Source	Initial Step 1	Dedup Step 2	Filter Step 3	Final Step 4
Binaries	238,746	187,578	167,821	148,003
Feeds	142,669	106,053	95,238	81,481
Twitter	69,193	57,762	56,424	54,731
GitHub	62,870	56,708	54,874	51,752
Total	513,478	408,101	374,357	***335,967***

Our social-media-based sources provide significant & correct information.

UC RIVERSIDE

ML can help in our future work here

- Training models based on data on C2 store.
- Using those to detect active C2 servers – based on profiles.
- How to use reinforcement learning to probe C2 servers during the active detection phase?

Thank you