Improving Vulnerability Remediation Through Better Exploit Prediction

Jay Jacobs, Cyentia
Sasha Romanosky, RAND
Idris Adjerid, Virginia Tech
Wade Baker, Virginia Tech

The Problem

- After 20 years, we security professionals and researchers are still unable to effectively measure and communicate cyber risk
- Collectively, we can't answer basic questions like:
  - Am I more secure now, relative to last year?
  - Which security controls work the best?

- In the mean time, firms are still being breached by vulnerabilities for which patches have existed for months or years

- It's a:
  - private sector *cyber security* problem
  - consumer, patient, student, and employee *privacy* problem
  -

Why is it so Difficult?

- One of the root causes is vulnerability management (VM)
  - Firms are pretty good at *finding* software vulnerabilities
  - They're just not very good at *fixing* them

- Many VM practices are based on prioritizing remediation by severity, e.g.:
  - DHS's directive requires agencies to patch based on high and critical severity vulns
  - PCI DSS requires credit card merchants patch vulns above a severity threshold

- As a decision rule, severity is good but doesn't incorporate information about whether the vuln is actually being exploited…
  -

The firm’s problem

- A firm may well have tens of thousands of open vulnerabilities
  - But only a small set will ever be exploited—5%, in fact

The firm’s problem

- A strategy based on *severity* catches many exploited vulns, but is very inefficient

The firm’s problem (again)

- While other research uses published exploits as the decision rule, it tells a similar story:
- Even if firms correctly patched all vulns with published exploits, many exploited vulns would still be missed

Inference vs Prediction

- Formally, we have a supervised learning classification problem
  - Our priority is to predict whether a vulnerability will be used in a real-world exploit,
  - rather than to develop or test theories about *why* vulnerabilities will be exploited

- But we still want to understand the model and interpret the results

Estimating Model

- Because of our class imbalance, we use gradient boosted trees, generated with extreme gradient boosting (XGBoost) (Chen and Guestrin, 2016) – which outperformed random forest and SVM models
  - We down-sampled (stratified) the majority class (exploit variable) during training (Kubat and Matwin, 2000), but tested on the full dataset
  - We evaluated models using 5-fold

Data (2009-2018)

Data Type	Source(s)	Obs(n)	Features(p)
CVSS score	NIST’s NVD	75,423	20
Vuln chars (products, vendor)	NIST’s CPE	75,582	69
Reference lists and vuln tags	MITRE’s CVE list, and URLs		

Results: Full ML Model

- Our ML model (dk blue) out performs other strategies (achieves 4.1k vulns at F1)
- We also consider approaches that favor efficiency and coverage

Next Steps

- This research isn't just about showing how ML outperforms simple heuristics
- It's about using *new* data, in *new* ways, in order to solve a chronic problem, and fundamentally change the way vulnerability management is performed
- That's a bold claim, but we believe the field is drastically in need of better solutions

- But we're not done!
- This approach is nice, but it's not very usable
- We're currently working to develop a threat scoring system that will be:
  - Transparent both the algorithms and scoring
  - Freely available possibly as an extension to CVSS, or a standalone calculator accessible through an API
- Stay tuned for BlackHat, 2019