From Patching Delays to Infection Symptoms: Using Risk Profiles for an Early Discovery of Vulnerabilities Exploited in the Wild

Chaowei Xiao[1], Armin Sarabi[1], Yang Liu[2], Bo Li[3], Mingyan Liu[1], Tudor Dumitras[4]

August 16, 2018

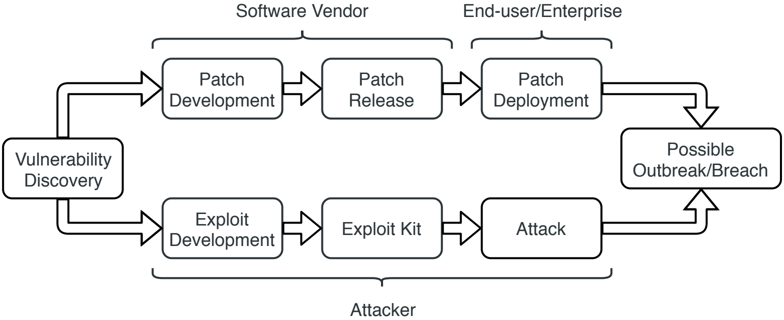
[1]University of Michigan, Ann Arbor

[2]Harvard University / UC Santa Cruz

[3]University of Illinois at Urbana–Champaign

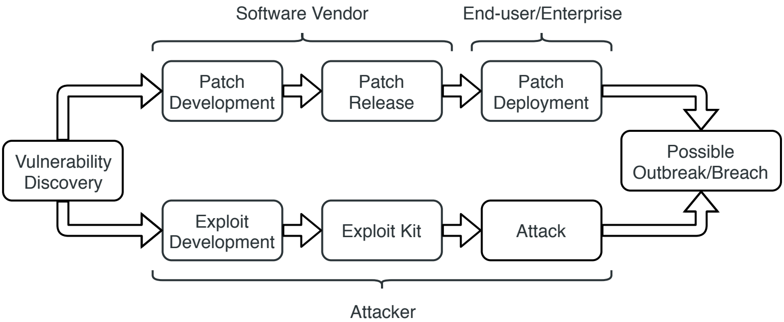
[4]University of Maryland, College Park

Introduction



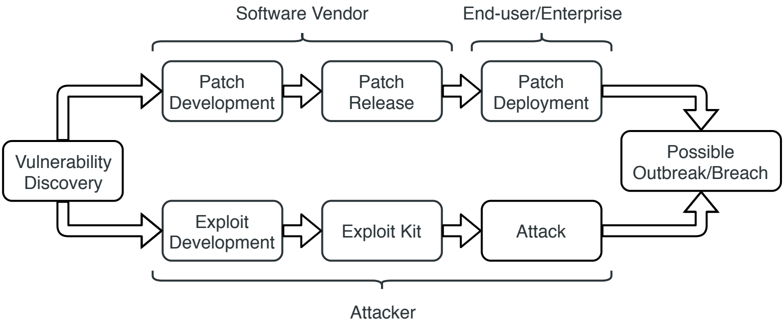
- Attackers are in a constant race with end-users/enterprises.

Background and Motivation



- Attackers are in a constant race with end-users/enterprises.
- It is estimated that on median, only 14% of vulnerable hosts are patched when exploits are made available.
    - Recent examples: WannaCry, NotPetya, Equifax.

Background and Motivation



- Attackers are in a constant race with end-users/enterprises.
- It is estimated that on median, only 14% of vulnerable hosts are patched when exploits are made available.
  - Recent examples: WannaCry, NotPetya, Equifax.
- Only a small portion of vulnerabilities are ultimately exploited.

Background and Motivation

Rank ordering vulnerabilities by severity enables prioritization of patch deployment.

Rank ordering vulnerabilities by severity enables prioritization of patch deployment.

Current state of exploit detection

- Intrinsic (a priori) attributes: Not strong predictors.
- Crawling social media sites: Only a few days of lead time.

Background and Motivation

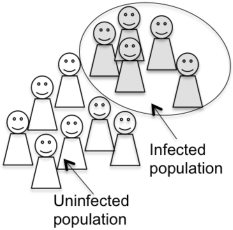
Rank ordering vulnerabilities by severity enables prioritization of patch deployment.

Current state of exploit detection

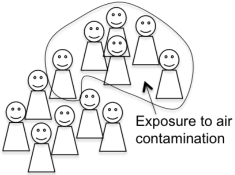
- Intrinsic (a priori) attributes: Not strong predictors.
- Crawling social media sites: Only a few days of lead time.

Our contribution

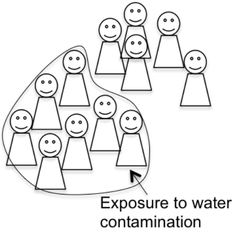
- Automated detection using statistical evidence of exploitation from real-world measurements.
- We achieve a 90% true positive rate, with a 10% positive rate using 10 days of post-disclosure observations.
  - The current median time for detection is 35 days.



Symptom pattern



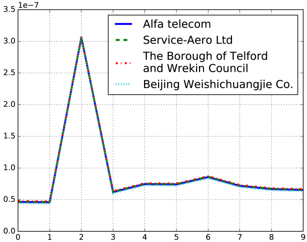
Risk behavior 1

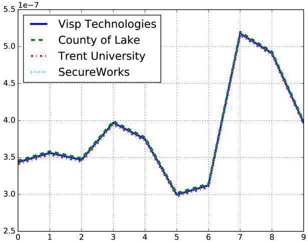


Risk behavior 2

- One can infer the main the cause of infection by comparing symptoms of infection with risk (vulnerability) patterns.

Overview of Concept





ISPs with similar symptom signals (i.e, number of infected hosts).

- One can infer the main the cause of infection by comparing symptoms of infection with risk (vulnerability) patterns.
- We combine this idea with community detection and compare symptoms of similar individuals (ISPs) with their risk behavior.

Datasets and Processing

Datasets

Symptoms

- Spam blacklists: CBL, SBL, SpamCop, UCEPROTECT, and WPBL (Jan 2013 - Present).

Datasets

Symptoms

- Spam blacklists: CBL, SBL, SpamCop, UCEPROTECT, and WPBL (Jan 2013 - Present).

Risk behavior

- Patching data for 7 applications from WINE (Feb 2008 - Jul 2014).
  - Chrome, Firefox, Thunderbird, Safari, Opera, Acrobat Reader, Flash.
- Publicly available vulnerabilities (CVEs) from NVD.

Datasets

Symptoms

- Spam blacklists: CBL, SBL, SpamCop, UCEPROTECT, and WPBL (Jan 2013 - Present).

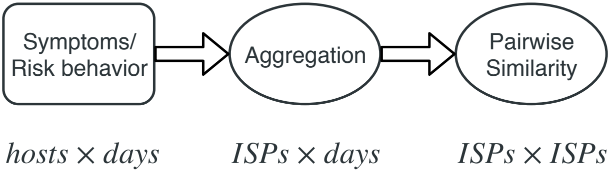
Risk behavior

- Patching data for 7 applications from WINE (Feb 2008 - Jul 2014).
    - Chrome, Firefox, Thunderbird, Safari, Opera, Acrobat Reader, Flash.
- Publicly available vulnerabilities (CVEs) from NVD.

Ground-truth

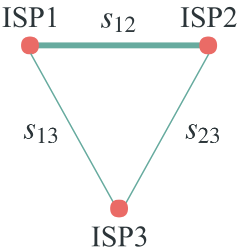
- Real-world exploits from SecurityFocus, Symantec, and Intrusion Protection Signatures (IPS).
- 56 exploited-in-the-wild (EIW) and 300 not-exploited-in-the-wild (NEIW) vulnerabilities.

Data Processing



- Reduce the number of nodes by aggregating at the ISP level.
- Compute pairwise similarity matrices for the aggregated signals.

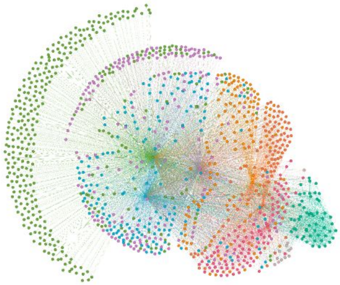
Data Processing



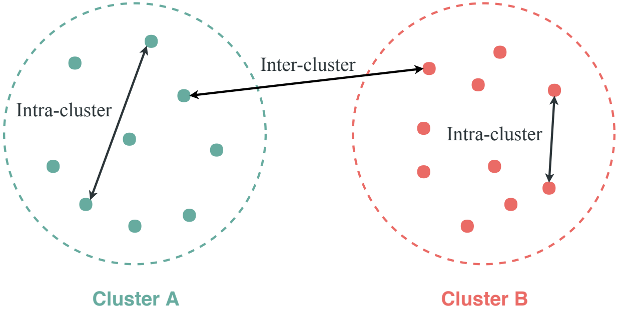
- Reduce the number of nodes by aggregating at the ISP level.
- Compute pairwise similarity matrices for the aggregated signals.
- For each CVE, this results in two weighted graphs (one for symptoms and one for risk behavior).

Methodology

Community Detection Over Symptom Similarity



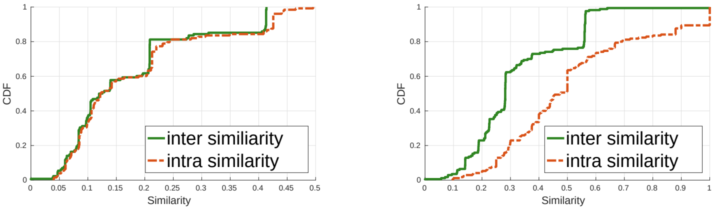
- Use community detection (BigClam) to identify groups of ISPs exhibiting similar symptoms for the 10-day period following each vulnerability disclosure.
- We investigate whether the same community structure also applies to risk behavior signals.



Intra- and inter-cluster similarities. Each node represents an ISP.

- Using the community structure obtained from symptoms, we compute the intra-cluster and inter-cluster similarities of risk behavior signals for each CVE.

Uncovering active exploitation



Distribution of intra- and inter-cluster risk similarities for a NEIW (left) and a EIW (right) vulnerability.

- We observe a statistically significant distinction between EIW and NEIW vulnerabilities.
- Conjecture: A higher intra-cluster similarity is an indication of active exploitation.

Evaluation

Feature Sets

Post-disclosure

- Community: 20-bin histogram of the difference in distribution between intra-cluster and inter-cluster similarities.

Feature Sets

Post-disclosure

- Community: 20-bin histogram of the difference in distribution between intra-cluster and inter-cluster similarities.
- Raw: Risk and symptom similarity matrices.

Post-disclosure

- Community: 20-bin histogram of the difference in distribution between intra-cluster and inter-cluster similarities.
- Raw: Risk and symptom similarity matrices.
- Direct: 20-bin histogram of row-by-row correlation between the two similarity matrices.

Feature Sets

Post-disclosure

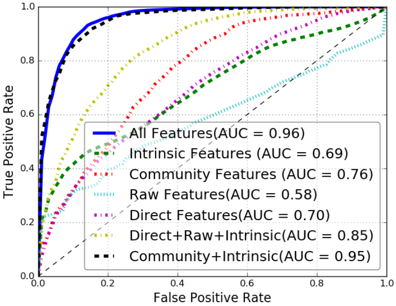
- Community: 20-bin histogram of the difference in distribution between intra-cluster and inter-cluster similarities.
- Raw: Risk and symptom similarity matrices.
- Direct: 20-bin histogram of row-by-row correlation between the two similarity matrices.

Intrinsic

- Tokens extracted from vulnerability descriptions, e.g., *remote*.
- CVSS scores summarizing the severity of each vulnerability.

Training

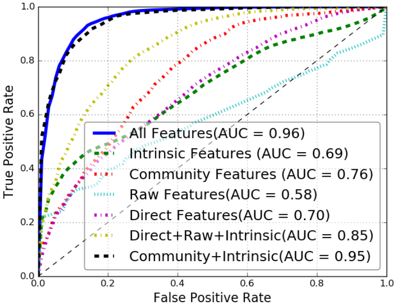
- Train Random Forests on different feature sets.
- Use 5-fold cross validation and average performance over 20 rounds.



Classifying EIW Vulnerabilities

Training

- Train Random Forests on different feature sets.
- Use 5-fold cross validation and average performance over 20 rounds.



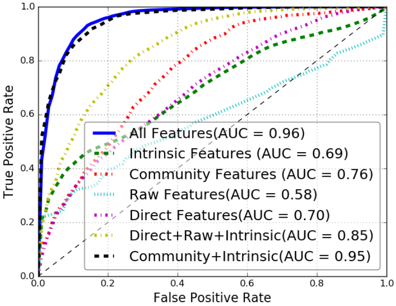
Accuracy of trained models

- Using all features we observe a 96% AUC.

Classifying EIW Vulnerabilities

Training

- Train Random Forests on different feature sets.
- Use 5-fold cross validation and average performance over 20 rounds.

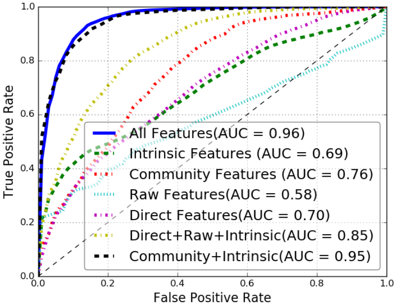


Accuracy of trained models

- Using all features we observe a 96% AUC.
- Community+Intrinsic features achieve a 95% AUC.

Training

- Train Random Forests on different feature sets.
- Use 5-fold cross validation and average performance over 20 rounds.



Accuracy of trained models

- Using all features we observe a 96% AUC.
- Community+Intrinsic features achieve a 95% AUC.
- Performance is greatly improved using both intrinsic (a priori) and post-disclosure (a posteriori) features.

Case Studies

The proposed technique can also be applied sooner/retrospectively.

Case Studies

The proposed technique can also be applied sooner/retrospectively.

CVE-2013-0640

- Disclosed on 02/13/2013, affecting Adobe Acrobat Reader.
- We detect exploitation for this CVE on the disclosure date.
- We were also able to find proof of zero-day exploits for this CVE.

Case Studies

The proposed technique can also be applied sooner/retrospectively.

CVE-2013-0640

- Disclosed on 02/13/2013, affecting Adobe Acrobat Reader.
- We detect exploitation for this CVE on the disclosure date.
- We were also able to find proof of zero-day exploits for this CVE.

CVE-2013-5330

- Disclosed on 11/12/2013, affecting Adobe Flash Player.
- The earliest exploit report date for this CVE is 01/28/2014.
- However, our system detected this vulnerability on the disclosure date, indicating a possible zero-day exploit.

Discussion and Conclusion

Discussion

Practical utility

- Enterprises: Prioritizing patch deployment, risk assessment.
- Software vendors: Development of patches for critical CVEs.
- ISPs: Identify at-risk populations to encourage prompt action.

Discussion

Practical utility

- Enterprises: Prioritizing patch deployment, risk assessment.
- Software vendors: Development of patches for critical CVEs.
- ISPs: Identify at-risk populations to encourage prompt action.

Data imperfections

- Malicious activities from multiple sources, e.g., different CVEs, pay-per-install, etc.
- Infections that do not generate spam.
- Aggregation at a coarse level can lead to only observing the averages of behavior.

Conclusion

Early exploit detection

- We can achieve a true positive rate of 90%, and a false positive rate of 10% using 10 days of post-disclosure data.
- The current median time for detection is 35 days, and 80% of reported exploits are detected beyond 10 days.
- Combining intrinsic and post-disclosure (community) features results in a robust classifier.

Conclusion

Early exploit detection

- We can achieve a true positive rate of 90%, and a false positive rate of 10% using 10 days of post-disclosure data.
- The current median time for detection is 35 days, and 80% of reported exploits are detected beyond 10 days.
- Combining intrinsic and post-disclosure (community) features results in a robust classifier.

Future directions

- Appending additional datasets of symptomatic data to build a more robust system.
- Using Internet scans to identify at-risk servers/networks.

Thank You

Questions?