Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-World Exploits

Carl Sabottke Octavian Suciu Tudor Dumitraș
University of Maryland
Problem

- Increasing number of vulnerabilities discovered
  - Since 2014 CVEID (the standard vulnerability identifier) is no longer limited to 10,000 vulnerabilities/year
Problem

• Increasing number of vulnerabilities discovered
  ◦ Since 2014 CVEID (the standard vulnerability identifier) is no longer limited to 10,000 vulnerabilities/year

• Multi-vendor coordinated disclosure
  ◦ 254 vulnerabilities disclosed on October 14 2014*

* - http://blog.osvdb.org/2015/02/02/vendors-sure-like-to-wave-the-coordination-flag-revisiting-the-perfect-storm/
Problem

• Increasing number of vulnerabilities discovered
  ◦ Since 2014 CVEID (the standard vulnerability identifier) is no longer limited to 10,000 vulnerabilities/year

• Multi-vendor coordinated disclosure
  ◦ 254 vulnerabilities disclosed on October 14 2014*

• Fewer than 15% of vulnerabilities are exploited [Nayak+, 2014]
  ◦ Need patching prioritization

* - http://blog.osvdb.org/2015/02/02/vendors-sure-like-to-wave- the-coordination-flag-revisiting-the-perfect-storm/
Research question

• Can we use Twitter analytics for early detecting exploits?
Intuition behind Twitter (1)

- Vulnerabilities disclosure for open-source software
- Discussion is about the patch itself

HHVM Commits
@HHVMCommits

github.com/facebook/hhvm/… Fix heap corruption in exif_ifd_make_value
Summary: CVE-2014-3670 The logic for writing float/double arrays in ex...
2:07 PM - 14 Oct 2014

Timeline

<table>
<thead>
<tr>
<th>Disclosure Date</th>
<th>2014-10-16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to Patch</td>
<td>18 days</td>
</tr>
</tbody>
</table>
Intuition behind Twitter (2)

• Leaks from the coordinated disclosure process

• More dangerous: miscreants can try to purchase exploit
Challenges for Exploit Prediction

• [Bozorgi+, 2010]: SVM classifier for predicting proof-of-concept exploits
  ◦ We aim to predict exploits active in the wild
  ◦ Only 1.3% of 2014 vulnerabilities labeled as exploited in our data set (compared to >50% in [Bozorgi+, 2010])
Challenges for Exploit Prediction

• [Bozorgi+, 2010]: SVM classifier for predicting proof-of-concept exploits
  ◦ We aim to predict exploits active in the wild
  ◦ Only 1.3% of 2014 vulnerabilities labeled as exploited in our data set (compared to >50% in [Bozorgi+, 2010])

• Additional challenges:
  ◦ Only 10% of 2014 vulnerabilities have more than 50 tweets
  ◦ Ground truth quality
Adversarial Interference

- Can adversaries post false information in order to trick a detector?
Contributions

• Design and implementation of a technique for early exploit detection using social media
Contributions

• Design and implementation of a technique for early exploit detection using social media

• Performance evaluation for detecting exploits found in the wild
Contributions

• Design and implementation of a technique for early exploit detection using social media

• Performance evaluation for detecting exploits found in the wild

• Analysis of system robustness to adversarial interference
Contributions

- Design and implementation of a technique for early exploit detection using social media
- Performance evaluation for detecting exploits found in the wild
- Analysis of system robustness to adversarial interference
- Performance evaluation for detecting proof-of-concept exploits
Contributions

• Design and implementation of a technique for early exploit detection using social media

• Performance evaluation for detecting exploits found in the wild

• Analysis of system robustness to adversarial interference

• Performance evaluation for detecting proof-of-concept exploits

• Categorize information leaks before vulnerabilities disclosure
Contributions

• Design and implementation of a technique for early exploit detection using social media

• Performance evaluation for detecting exploits found in the wild

• Analysis of system robustness to adversarial interference

• Performance evaluation for detecting proof-of-concept exploits

• Categorize information leaks before vulnerabilities disclosure
Twitter Dataset

• Twitter Public Stream
  ◦ February 2014 - January 2015
  ◦ 1.1 billion tweets
Twitter Dataset

• Twitter Public Stream
  ◦ February 2014 - January 2015
  ◦ 1.1 billion tweets

• Tracking the **cve** keyword
Twitter Dataset

- Twitter Public Stream
  - February 2014 - January 2015
  - 1.1 billion tweets

- Tracking the **cve** keyword

- Collected **unsampled** corpus
  - 287,717 tweets
  - 5,865 vulnerabilities
System Design

Features
System Design

CVSS v2 Base Metric

Availability Impact

Authentication
System Design

- Database Features
  - OSVDB Category
  - # of References in NVD

Features

[Image of National Vulnerability Database]
System Design

Features

- National Vulnerability Database
- Twitter
System Design

Twitter Word Features

exploit

web

...
System Design

Features

Twitter Traffic Features

- Number of Tweets
- Number of Retweets
...
System Design
System Design

Web Attack: Adobe Flash CVE-2011-2140

Severity: High
This attack could pose a serious security threat. You should take immediate action to stop any damage or prevent further damage from happening.

Description
This signature detects a buffer overflow vulnerability in Adobe Flash Player.
System Design
System Design

1. **Features**
   - National Vulnerability Database
   - osvdb

2. **Ground Truth**
   - Symantec Security Response
   - EXPLOIT DATABASE
   - Microsoft Security Advisories

3. **Training**
   - Twitter

4. **Linear SVM**
System Design

- Features
- Ground Truth
- Linear SVM
- Classification Results
Detecting Exploits in the Wild

1. Features
2. Ground Truth
3. Linear SVM
4. Classification Results

Training

Testing

Symantec Security Response

Microsoft Security Advisories
Classifier evaluation

- **Precision** = fraction of vulnerabilities marked as exploited that are *actually exploited*
Classifer evaluation

- **Precision** = fraction of vulnerabilities marked as exploited that are *actually exploited*
  - False Positive = marked as exploited but *not* exploited in the wild
  - False positives hurt precision
Classifier evaluation

- **Precision** = fraction of vulnerabilities marked as exploited that are *actually exploited*
  - False Positive = marked as exploited but *not* exploited in the wild
  - False positives hurt precision

- **Recall** = fraction of exploited vulnerabilities that are *marked as exploited*
Baseline Classifier

- Using CVSS Score as indicator of an exploit
Baseline Classifier

- Using CVSS Score as indicator of an exploit
- CVSS marks many vulnerabilities as **exploitable**

\[
\text{Precision} \leq 9\%
\]
Baseline Classifier

- The Exploitability sub-score measures the likelihood of an exploit.
Baseline Classifier

- The Exploitability sub-score measures the likelihood of an exploit.
- Precision is still low.

\[
\text{Precision} \leq 9\% \}
\]
Utility of Features

- **Mutual Information** measures how much knowing a feature reduces uncertainty about the class.

Details about these features: [http://ter.ps/sec15exploit](http://ter.ps/sec15exploit)
Detecting Exploits in the Wild

- CVSS Score: very low precision, high recall
Detecting Exploits in the Wild

- Database Information: High recall, low precision
Detecting Exploits in the Wild

- Twitter Word features: low recall, high precision
Detecting Exploits in the Wild

- Twitter Traffic features: higher recall, lower precision
Detecting Exploits in the Wild

• Combining all features: variable regularization results in a precision/recall tradeoff
Exploits in the Wild Classification

• Can we improve the classification performance?
Exploits in the Wild Classification

- Filtering based on ground truth coverage improves performance
Exploits in the Wild Classification

- Thresholding based on tweet volume improves performance even further

[Graph showing precision vs. recall for different categories: All CVE, MS and Adobe Only, Min 50 Tweets]
Early Prediction of Exploits
Early Prediction of Exploits

- National Vulnerability Database
- Features
- Ground Truth
- Symantec Security Response
- Date
- EXPLOIT DATABASE
- Microsoft Security Advisories
- Twitter
Early Prediction of Exploits

Features

Ground Truth

Linear SVM

Date

Training
Early Prediction of Exploits
Early Prediction of Exploits

- Features
- Ground Truth
- Linear SVM
- Prediction Threshold

Integration with:
- National Vulnerability Database
- Symantec Security Response
- Exploit Database
- Twitter

Training and Streaming Processes
Tweets before Signatures

- Reference point: when attacks are reported in the wild
Tweets before Signatures

- Time difference between first tweets and attacks

![Graph showing the percentage of real-world exploits against day difference between Twitter and attack signature dates. The x-axis represents the day difference, and the y-axis represents the percentage of real-world exploits. The graph shows a sharp increase in the percentage of exploits as the day difference decreases.]
Tweets before Signatures

- Time difference between first tweets and attacks

Opportunity for Early Detection
Early Prediction for Heartbleed

• First tweet posted 2 days ahead of ground truth
Early Prediction for Heartbleed

• Evolution of the classifiers’ prediction probability
Early prediction for Heartbleed

- Possible detection: 3 hours after first tweet

![Diagram showing early detection of Heartbleed vulnerability](image)
Tweets before Signatures

- Tradeoff between precision and detection lead time
Tweets before Signatures

- Tradeoff between precision and detection lead time
- Median detection 2 days ahead of Symantec signatures
Tweets before Signatures

- Tradeoff between precision and detection lead time
- Median detection 2 days ahead of Symantec signatures
- 45% classification precision

Detection mostly occurs earlier than the signatures
Adversarial Interference

1. Features
2. Ground Truth
3. Linear SVM
4. Classification Results

- Training
- Testing

- National Vulnerability Database
- Symantec Security Response
- Exploit Database
- Microsoft Security Advisories
Adversarial Interference

- Features
- Ground Truth
- Linear SVM
- Classification Results
- Exploratory Attack

- National Vulnerability Database
- Symantec Security Response
- Microsoft Security Advisories

Testing

Training
Adversarial Interference

Classification Results

Causative Attack
Attacks Against the Exploit Detector

• Can we prevent the adversary from poisoning the training dataset?
Attacks Against the Exploit Detector

• Can we prevent the adversary from poisoning the training dataset?
  ◦ No. Twitter is a free and open service.
Attacks Against the Exploit Detector

• Can we prevent the adversary from poisoning the training dataset?
  ◦ No. Twitter is a free and open service.

• Can we keep the features secret?
Attacks Against the Exploit Detector

- Can we prevent the adversary from poisoning the training dataset?
  - No. Twitter is a free and open service.

- Can we keep the features secret?
  - No. Our ground truth comes from public sources.
Attacks Against the Exploit Detector

• Can we prevent the adversary from poisoning the training dataset?
  ◦ No. Twitter is a free and open service.

• Can we keep the features secret?
  ◦ No. Our ground truth comes from public sources.

• Is the adversary resource bound?
Attacks Against the Exploit Detector

- Can we prevent the adversary from poisoning the training dataset?
  - No. Twitter is a free and open service.

- Can we keep the features secret?
  - No. Our ground truth comes from public sources.

- Is the adversary resource bound?
  - Yes. He must control the properties of multiple accounts.
Adversary Model

• Adversary’s goal: to introduce false positives
Adversary Model

• Adversary’s goal: to *introduce false positives*

• Simulation of *causative attacks*
Adversary Model

- Adversary’s goal: to introduce false positives
- Simulation of causative attacks
- Adversaries cannot prevent benign users from posting
Blabbering Adversary

- Randomly posts tweets, no knowledge about features
Blabbering Adversary

- Randomly posts tweets, no knowledge about features
- Does minimal damage to the classifier

Random noise minimally affects the system
Word Copycat Adversary

- Mirrors the statistics of words corresponding to exploited vulnerabilities
Word Copycat Adversary

- Mirrors the statistics of words corresponding to exploited vulnerabilities
- Damage is bound due to other features (Traffic, CVSS, Databases)

Resilience due to other feature types
Full Copycat Adversary

- Sybil-like: manipulates all Twitter features except account creation date and account verification
Full Copycat Adversary

- Sybil-like: manipulates all Twitter features except account creation date and account verification
- A list of trusted users needed for resilience. We found a list of ~4,000 useful users

Damage is bound only by non-Twitter features
Conclusions

• Design of a Twitter-based exploit detector
  ◦ Uses: patching prioritization, risk assessment
Conclusions

• Design of a Twitter-based exploit detector
  ◦ Uses: patching prioritization, risk assessment

• Early detection of exploits active in the wild
  ◦ Performance depends on the quality of ground truth
Conclusions

• Design of a Twitter-based exploit detector
  ◦ Uses: patching prioritization, risk assessment

• Early detection of exploits active in the wild
  ◦ Performance depends on the quality of ground truth

• Exploit detection under adversarial interference
  ◦ Security system without secrets
Thank you!

osuciu@umiacs.umd.edu

Detailed feature list available at:

http://ter.ps/sec15exploit
Public Proof of Concept Exploits Classification

![Graph showing precision and recall for different datasets: All CVE, Min 10 Tweets, Min 20 Tweets.]
Public Proof of Concept Exploits Classification

• Sharp contrast with previous work
  ◦ [Bozorgi+, 2010] reported 87.5% precision

• Reasons for the differences
  ◦ Threat landscape has changed
  ◦ Fewer vulnerabilities are exploited
  ◦ Vulnerabilities databases have worse coverage
Precision vs Early detection

- Variable threshold = precision/timeliness tradeoff
• Variable threshold = precision/timeliness tradeoff

45% precision, median 2 days ahead
Informative Users

- Whitelisting users based on utility with respect to real world exploits
Features

• CVSS Information - 7
  ◦ CVSS Score and the sub-scores (e.g., Authentication)

• Twitter Words - 31
  ◦ word features after a feature selection step

• Twitter Traffic - 12
  ◦ e.g., no. of tweets, no. of URLs, no. of distinct users

• Database Information - 17
  ◦ useful in prior work [Bozorgi+, 2010]
  ◦ e.g., vulnerability category, no. of references in NVD
User Whitelisting
User Whitelisting

• Building a trusted list of users
User Whitelisting

• Building a trusted list of users

• Users utility metric takes into account:
  ◦ Ratio of real-world exploits to all vulnerabilities
  ◦ Fraction of unique real-world exploits
User Whitelisting

• Building a trusted list of users

• Users utility metric takes into account:
  ◦ Ratio of real-world exploits to all vulnerabilities
  ◦ Fraction of unique real-world exploits

• Ranking of users based on utility
User Whitelisting

• Building a trusted list of users

• Users utility metric takes into account:
  ◦ Ratio of real-world exploits to all vulnerabilities
  ◦ Fraction of unique real-world exploits

• Ranking of users based on utility

• Whitelist contains top 20% ranked users: ~ 4,000 users
User Whitelisting