LEARNING TO DETECT PHISHING EMAILS
Ian Fette, Norman Sadeh, Anthony Tomasic
Presented by: Andrew Martin
OUTLINE

• What is phishing?
• Email Filtering
• Ten classifications
• Machine-Learning
• False Positives vs. False Negatives
• Untangle
Phishing is the criminally fraudulent process of attempting to acquire sensitive information such as usernames, passwords and credit card details by masquerading as a trustworthy entity in an electronic communication.

- Occurs within e-mail or instant messaging
- Loosely defined DNS attacks are another form of phishing
PROBLEMS IDENTIFYING PHISHING

• Phishing attempts are closely targeted attacks
• Generally only sent to mere tens of thousands
  • thwarts typical traffic analysis tools
• The authenticity of the emails generate a much higher “click-through” rate than spam
• Can’t expect to reject emails based on spam MX records
COMMON TECHNIQUES

• Forge the from header
• Forge the received chain
• Insert spaces and odd characters in the name and email address
• Use services like google’s redirection service
• Insertion of images instead of text in the email, but include a text stream of random characters to confuse filtering algorithms
• Hex-coded URLs, non-matching hyperlinks
• Embed html forms
EXISTING TOOLS

- Firefox & Internet Explorer include tools
  - Only check sites against a list of known phishing sites
- Norton 360
- McAfee Internet Security Suite
- Various web toolbars: Spoofguard, Netcraft
EMAIL VS. WEB FILTERING

• Web browser filters are at a disadvantage
  • Contain no contextual information
  • Cannot obtain “why” the user clicked on the link
  • No access to headers within an email
  • Cannot completely shield the user from the decision making process
EMAIL FILTERING

• Phishing is not generally targeted specifically

• Primitive client based detections are based on linguistics only

• Thunderbird uses 3 criterion
  • IP-based URLs, Non-matching URLs and presence of HTML form element
PILFER

- Phishing Identification by Learning on Features of Email Received

- A machine-learning based approach to classification

- Intelligently deciding if an email is deceptive or not
TEN CLASSIFICATIONS

- IP-based URLs
- Age of linked-to domain names
- Non-matching URLs
- “Here” links to non-modal domain
- HTML emails
- Number of links
- Number of domains
- Number of dots
- Contains JavaScript
- Spam-filter output
SYSTEM SETUP

• Datasets are divided into ten distinct parts (based on prior criterion)

• Each dataset is tested against the ten criteria previously discussed

• Uses a random forest to create ten decision trees
TRAINING THE SYSTEM

• System is compared to SpamAssassin in both a trained and untrained state

• System tests against two publicly available data sets consisting of 7360 emails (6500 ham, 860 phishing) from 2002-2003

• Scripts extract the ten defined criterion mentioned earlier and store them in a database
**FALSE POSITIVES VS FALSE NEGATIVES**

- False positive = proportion of ham emails classified as phishing
  
  \[ fp = \frac{\text{ham}_{\text{phish}}}{\text{ham}_{\text{phish}} + \text{ham}_{\text{ham}}} \]

- False negative = proportion of phishing emails classified as ham
  
  \[ fn = \frac{\text{phish}_{\text{ham}}}{\text{phish}_{\text{ham}} + \text{phish}_{\text{phish}}} \]

- \( fp = 0.1 \) means 1 in 10 good emails would be classified as phishing

- \( fn = 0.2 \) means 2 in 10 phishing emails would be classified as good
EVALUATION

- PILFER on its own achieves an overall accuracy of ~99.5% (fp=~0.0013, fn=~0.035)

- Almost 1/4 the fn rate of the spam filter by itself

Table 1: Accuracy of classifier compared with baseline spam filter

<table>
<thead>
<tr>
<th>Classifier</th>
<th>False Positive Rate $fp$</th>
<th>False Negative Rate $fn$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PILFER, with S.A. feature</td>
<td>0.0013</td>
<td>0.036</td>
</tr>
<tr>
<td>PILFER, without S.A. feature</td>
<td>0.0022</td>
<td>0.085</td>
</tr>
<tr>
<td>SpamAssassin (Untrained)</td>
<td>0.0014</td>
<td>0.376</td>
</tr>
<tr>
<td>SpamAssassin (Trained)</td>
<td>0.0012</td>
<td>0.130</td>
</tr>
</tbody>
</table>

Table 2: Percentage of emails matching the binary features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Non-Phishing Matched</th>
<th>Phishing Matched</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has IP link</td>
<td>0.06%</td>
<td>45.04%</td>
</tr>
<tr>
<td>Has “fresh” link</td>
<td>0.98%</td>
<td>12.49%</td>
</tr>
<tr>
<td>Has “nonmatching” URL</td>
<td>0.14%</td>
<td>50.64%</td>
</tr>
<tr>
<td>Has non-modal here link</td>
<td>0.82%</td>
<td>18.20%</td>
</tr>
<tr>
<td>Is HTML email</td>
<td>5.55%</td>
<td>93.47%</td>
</tr>
<tr>
<td>Contains JavaScript</td>
<td>2.30%</td>
<td>10.15%</td>
</tr>
<tr>
<td>SpamAssassin: Output</td>
<td>0.12%</td>
<td>87.05%</td>
</tr>
</tbody>
</table>

Table 3: Mean, standard deviation of the continuous features, per-class

<table>
<thead>
<tr>
<th>Feature</th>
<th>$\mu_{\text{phishing}}$</th>
<th>$\sigma_{\text{phishing}}$</th>
<th>$\mu_{\text{non-phishing}}$</th>
<th>$\sigma_{\text{non-phishing}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of links</td>
<td>3.87</td>
<td>4.97</td>
<td>2.36</td>
<td>12.00</td>
</tr>
<tr>
<td>Number of domains</td>
<td>1.49</td>
<td>1.42</td>
<td>0.43</td>
<td>3.32</td>
</tr>
<tr>
<td>Number of dots</td>
<td>3.78</td>
<td>1.94</td>
<td>0.19</td>
<td>0.87</td>
</tr>
</tbody>
</table>
UNTANGLE

• Open source suite of applications geared toward preventing access to harmful sites, as well as protecting one’s network

• Combines a number of freely available applications for various services

• Neatly packaged and pre-configured debian based linux distro
FEATURES (FREE)

- Web Filter and Phishing Blocker
- Spam/Virus Blocking
- IPS/IDS
- OpenVPN
- Firewall/Router
- Reporting
INTERFACE

- Complete Linux distro
- Installation is guided and graphical
- Web based interface for simple configuration

linuxplanet.com: “it’s the best experience we’ve ever had”
DEMO

• http://www.untangle.com/video_overview/
CONCLUSION

• Phishing detection is tricky and currently overlooked

• Combining multiple methods increases accuracy

• Using spam filter output as input increases accuracy even further

• Many of the existing systems should make use of more advanced techniques as described in this paper

• End users should never be trusted
REFERENCES

• http://www.clamav.net/doc/latest/phishsigs_howto.pdf

• http://www.untangle.com


• http://www.linuxplanet.com/linuxplanet/reviews/6452/1/