Detecting Phishing Attempts with SAS®: Cisco IronPort® Email Log Data
Taylor B. Anderson, Denise J. McManus
The University of Alabama

ABSTRACT
A common threat for internet users today is phishing -- the attempt of a malicious entity to acquire personal, financial, or otherwise sensitive information such as usernames and passwords from email recipients. Quickly alerting users of these attempts or even removal of these attempts from users' mailboxes can greatly reduce the risk to an organization of a user succumbing to a phishing attack. Traditional methods for detection of phishing attacks require the body of each email to be analyzed, however this poster focuses on the analysis of email header and subject line information as a method for detection of phishing attacks that survive analysis by Cisco IronPort® Email Security Appliances. Previously a collection of 52,000 emails were examined with moderate success utilizing a logistic regression model and simplistic binary inputs. This poster focuses on the modeling gains from text mining email subject lines, sender, and recipient information as a means to further reduce misclassification error. The resulting model output is compared against user feedback of suspicious emails as a method of phishing detection and alerting. Previously, near real-time email traffic was exported from Splunk® Enterprise into SAS® for analysis. This paper also explores the use of SAS® Event Stream Processing to increase response time to a phishing attack.

MOTIVATION
Phishing attacks are a constant threat to any organization. End user education can help prevent users from being phishing, but this is not always effective. Many organizations rely on users to forward suspicious emails to trained personnel for analysis, but again, this is not always performed by users. Once an email is identified, an announcement can be posted to recipients of the email so they are aware that the email should be deleted and, if they did succumb to the attack, the user should change their passwords and contact their organization’s information security (IS) personnel. The motivation behind this work is to alert IS personnel to subject lines of potential phishing emails, thus, taking a proactive approach to detecting and eliminating suspicious emails. By using minimally invasive email log data, organizations that either do not or cannot analyze message bodies may be able to detect these attempts much quicker than relying on user notification. By detecting these emails earlier, users that have received the emails can be notified sooner (ideally before they even open the phishing email) and block rules can be added to the filtering appliance to prevent further delivery of the phishing emails.

GOALS
The main objective:
• Catch phishing emails that were missed by a Cisco IronPort® Email Security Appliance and alert on them
Current goals:
• Begin basic text analysis of message subject lines
• Begin exploring integration with SAS® Event Stream Processing

THE DATA
• The models are built using 1,568 phishing emails and 10,000 legitimate emails
• Splunk® Enterprise is used to export email log data from a Cisco IronPort® ESA in csv format
• A python program is then used to
  o Clean up the date format to something usable by SAS® (E8601LX24.)
  o Add a new field called “has_attachment” indicating that the message had an attachment
  o Drop attachment type and file name fields
• Another python program is used to do some basic text mining on each subject line
  o Count punctuation
  o Count total words
  o Count capitalized words
  o Count words written in all capital letters
  o Count misspelled words (via NLTK corpus word lists)
  o Count punctuation at the end (ie. Help!!! = 3)
  o Count common phishing words (seen at UA – help, desk, helpdesk, fax, mailbox)
  o Count common phishing words (seen at UA – ":")
  o Percentage of words that are capitalized
  o Percentage of words in all capital letters

PREVIOUS WORK
• Determine effectiveness of utilizing email syslog data for detection of phishing emails that get through a Cisco® ESA
• Create a model and integrate with Splunk® Enterprise to alert IS personnel to likely phishing emails
• Create a more refined logistic regression model using SAS® Enterprise Miner
• Implement a basic threat score and database
• Determine common words and punctuation in phishing emails seen at The University of Alabama (UA)
The data is further manipulated in Base SAS®:

- New variables are added to denote:
  - Whether senders are internal staff, faculty, students or external
  - Whether or not the subject was encoded (UTF)
  - The number of recipients of the email
  - The size of the message
  - Whether or not the email was sent during work hours

SAS® Enterprise Miner is used to explore various models for determining the likelihood of an email being a phishing email.

Model comparison shows a clear winner, MBR.

Previously, SAS® Enterprise Miner was used to create a logistic regression model to determine whether or not an email is likely to be a phishing email. By comparing misclassification rate and other factors, a logistic regression model was shown to be most effective before the text analysis program was added. The misclassification rate before was 36% with a false negative rate of 11%.

In this revision, the logistic regression model’s misclassification rate has improved to approximately 11%, but a Memory-Based Reasoning model provides significantly better results. The misclassification rate is approximately 2%.

The University of Alabama

Live email traffic is fed into this resulting model using the same Python and SAS® programs with some additional changes:

- Five minutes of email traffic is captured every two minutes, resulting in a three minute overlap of the data.
- Any message subjects determined to not be part of a phishing attack by the model are removed.
- Message subjects with a probability of being part of a phishing attack less than 0.8 are removed.
- Messages sent to less than 5 people at a time are removed.
- A basic threat score is created using \( \text{MAX\_P} = \text{the max probability of all messages with a given subject line} \) and \( \text{COUNT} = \text{number of messages with a given subject line that were sent} \):
  \[
  \text{THREAT\_SCORE} = \text{MINIMUM}(100, \text{ROUND}(10 \times \text{MAX\_P} \times \text{COUNT}))
  \]
- These features were removed. The first two increased false negative alerts with the MBR model and are no longer necessary. The last was removed as no longer necessary at this time.
- The resulting subject lines are compared to those within a database of past detected subject lines. New subject lines are added to the database and emailed to IS personnel for examination. Old subject lines are removed and the database last_seen field for that subject line is updated to the current timestamp.
**Detected Phishing Attempts with SAS®: Cisco IronPort® Email Log Data**

Taylor B. Anderson, Denise J. McManus

The University of Alabama

---

**BASH Script**

```
export PYTHONPATH=/opt/splunk-sdk-python; cd /opt/reg
/opt/reg/search.py --username=xxxxx --password=xxxxx --host=xxxx --app=search --search-index=xxxx | transaction internal_message_id | table _time, recipient, sender, message_size, message_subject, spam_status, attachment_type, file_name' --earliest_time="$(date +%Y-%m-%dT%T.000" --date '-3 min')" --latest_time="$(date +%Y-%m-%dT%T.000")" --output_mode=csv > temp.csv
# replace _time with datetime and clean up datetime format
/usr/bin/sed -i 's/_time/datetime/g' temp.csv
/usr/bin/sed -i 's/....-0/....-0/g' temp.csv
# do additional row cleanup
/usr/bin/python pre_sas_cleanup.py >> temp.csv
/usr/bin/mv temp.csv temp2.csv

# Call SAS
/usr/local/SASHome/SASFoundation/9.4/bin/sas_en --nodms /opt/reg/new_esa_mbr.sas
/usr/bin/cat *.lst | egrep -v "SPAM|Undeliverable" > emailbody.tmp
if [ -s emailbody.tmp ];
    then
    # REPLACED* -- send the email and add the lines to the database
    fi
/usr/bin/rm *.lst emailbody.tmp temp.csv 2> /dev/null
/usr/bin/echo "$((/usr/bin/date) - ran)" >> log.txt
```

*Small portion of the bash script that runs the entire process. Email and database portions were removed to protect the organization.

---

**Phishing Alert**

Taylor B. Anderson

message_subject: 

```
For: new message
```

---

Sample email output from a small phishing attempt.
The next phase is to reduce complexity by implementing equivalent steps within SAS® Event Stream Processing (ESP).

Currently, the existing python code has been converted to write a file to the SAS® ESP server for ingestion. At the time of writing, the data is able to be imported successfully. Next steps with SAS® ESP include:

• Convert python code ingestion to socket or syslog input
• Convert existing filtering code to a Filter window in SAS® ESP
• Utilize a Procedure window to run the existing MBR model
• Send email notifications via the Notification window

CONCLUSION AND FUTURE WORK

In the case of this project, false positives are worse from an automation perspective while false negatives are worse from a risk perspective. Reducing the risk to end users and thus their organization, is most important. Future work with subject line analysis will ideally reduce both types of error. A fully automated backend is clearly not the correct approach until type I error is reduced further. Future work with SAS® Event Stream Processing will be focused on completing integration and simply alerting security personnel and quarantining email messages instead of removing them entirely. Additionally, grammar analysis of subject lines will be conducted. Finally, a method of continuously updating the model with new confirmed phishing emails will be explored.

REFERENCES

Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.