Security Analysis of HTML and JS Contents of Alexa top 1m Websites

Team 6

April 20, 2017

Outline

Problem Statement

Content analysis of Alexa 1m domains to

- Detect security relevant **modifications** in the contents of web pages

Problem Statement

Content analysis of Alexa 1m domains to

- Detect security relevant **modifications** in the contents of web pages

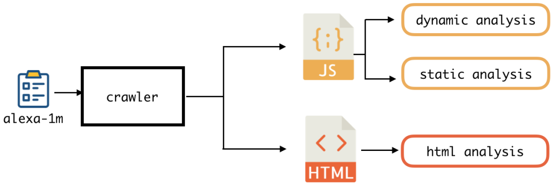
and

- Classification of websites as **malicious** or **benign** on the basis of scores from the classifier

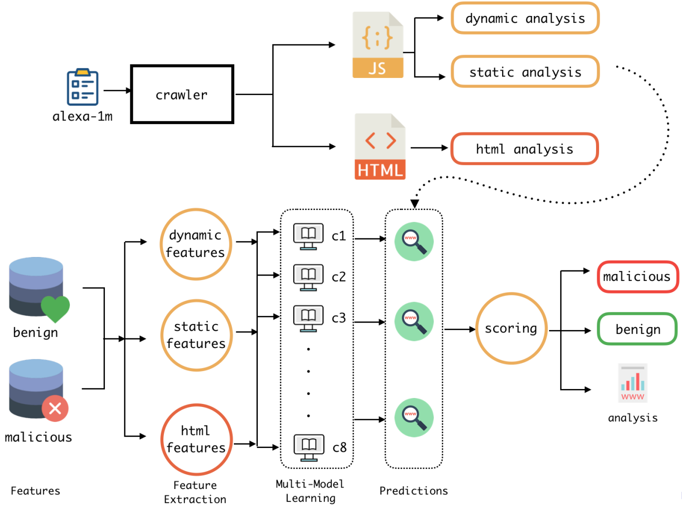
Strategy and Approach

- Crawler for data collection
  - Categorized Search
- Feature extraction
  - HTML Features
  - JS Features
- Designed and implemented a system to detect content modifications having security relevance
- Machine Learning Classifier
  - Off-line training mode
  - On-line testing mode

Architecture: The Big Picture



Architecture: The Big Picture



Crawler - Golang

- We chose `Go` to write crawler
  - It is succinct, minimalistic and fast compared to others
  - Designed and optimized for scaling
  - Supports to write a multi-threaded programs
- On an average the crawler takes **1.2 days** to crawl the alexa-1m websites (HTML & JS contents).

HTML Analysis: Feature Extraction

1. Number of iframe tags
2. Number of hidden elements
3. Number of elements with small area
4. Number of script elements
5. Percentage of scripting content in a page
6. Percentage of whitespace in a page
7. Presence of meta refresh tags
8. Number of embed or object tag
9. Number of elements with src on external domain
10. Number of included urls
11. Number of characters in a page

Static JS Analysis

1. Ratio of keywords to non-keywords
2. Number of strings with lengths larger than 40
3. Number of suspicious tags
4. Number of iframe tags
5. Entropy of the total script
6. Number of suspicious strings
7. Number of decodeURIcomponent
8. Number of functions clearAttributes, insertAdjacentElement, and replaceNode
9. Number of setTimeOut
10. Number of exec calls
11. Number of applets and scripts

Dynamic JS Analysis

1. Number of dynamic function calls
2. Number of Wscript saved files
3. Number of URLs
4. Number of Wscript objects
5. Number of *setTimeout()* calls
6. Number of *eval()* calls
7. Number of *unescape()* calls
8. Number of browser documents

Classification Overview

- Converting scripts to points in euclidean space
- Classifier operates on euclidean space
- Each classifier trained with subset of the benign and malicious samples
- Each classifier is tested with rest of the samples

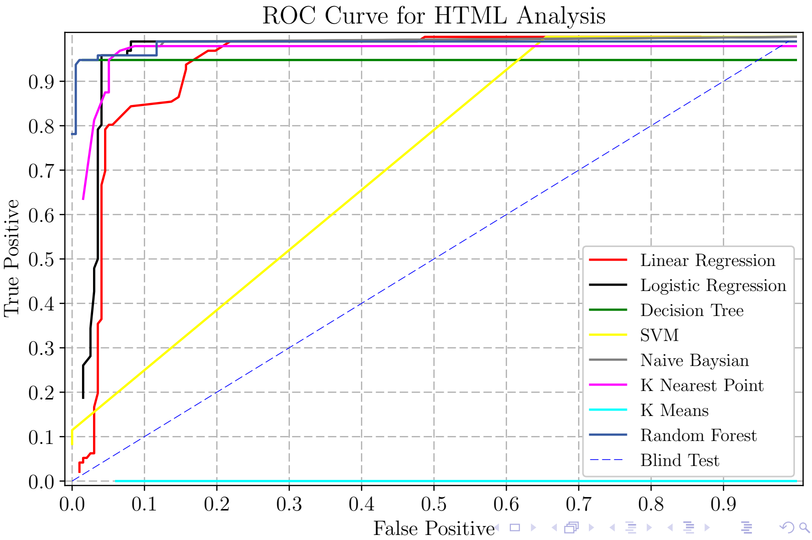
Multi-model Training and Classification

- Usage of seven supervised and one unsupervised classifier
- Classifier exhibiting best results chosen
  - HTML Analysis - Random Forest
  - Static JS Analysis - Random Forest
  - Dynamic JS Analysis - Decision Tree

Results (1)

- Used 80 percent of our the known samples for training and 20 for testing our classifiers

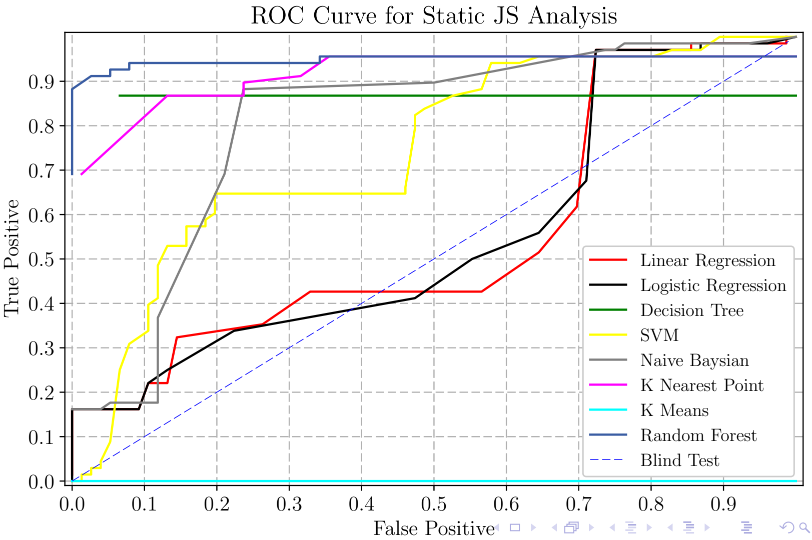
Results (1)


ROC Curve for HTML Analysis
True Positive
False Positive
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Linear Regression
Logistic Regression
Decision Tree
SVM
Naive Baysian
K Nearest Point
K Means
Random Forest
Blind Test

Results (1)

- Used 80 percent of our the known samples for training and 20 for testing our classifiers

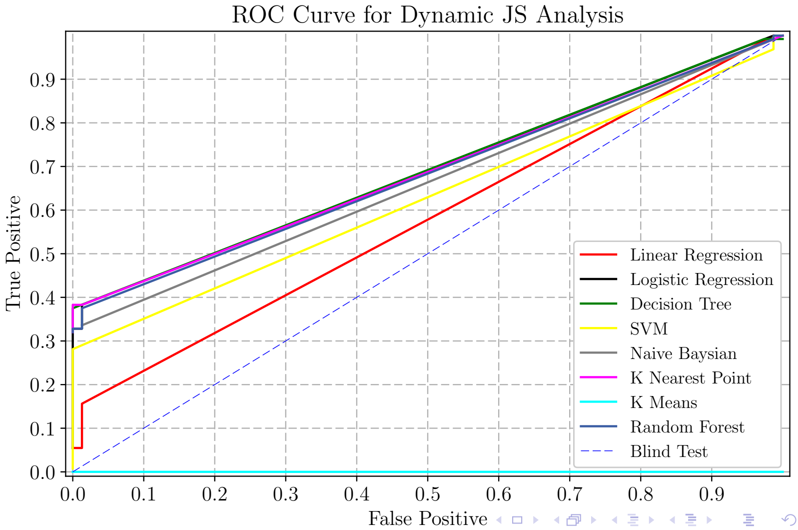
Results (1)


ROC Curve for Static JS Analysis
True Positive
False Positive
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Linear Regression
Logistic Regression
Decision Tree
SVM
Naive Baysian
K Nearest Point
K Means
Random Forest
Blind Test

Results (1)

- Used 80 percent of our the known samples for training and 20 for testing our classifiers

Results (1)


ROC Curve for Dynamic JS Analysis
True Positive
False Positive
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Linear Regression
Logistic Regression
Decision Tree
SVM
Naive Baysian
K Nearest Point
K Means
Random Forest
Blind Test

Results (2)

	Benign	Malicious
HTML analysis	998	467
Static JS analysis	363	353
Dynamic JS analy	377	648

Figure: Number of Samples for Training

	Precision
HTML analysis	0.013
Static JS analysis	0.019
Dynamic JS analy	0.016

Figure: Precision of Error Probabilities

Results (3)

Project	FP	FN	Static	Dynamic	Obfuscation resilient
Our Work	4.85%	7.4%	✓	✓	✓
Zozzle	4.56%	4.51%	✓	✗	✗

- Our static JS analyzer provides almost same false positive rate as Zozzle - a fast precise static JS analysis tool
- In addition, we facilitate dynamic analysis - absent in Zozzle.

Results (4)

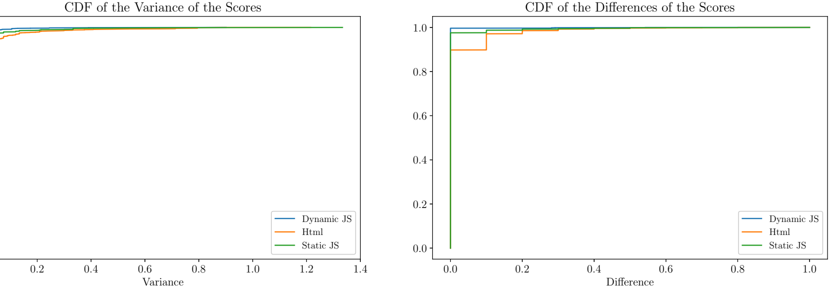


Figure: CDF's of Variance and Difference of Scores

Security Problems (1)

- *Day 1 to Day 2 in webdade.com*: Lot of scripts in day 1 vanished in day 2 and there were several calls in day 1 that point to www.smartsuppchat.com which is a black-listed website across multiple blacklists.
- *Day 3 to Day 4 in webdade.com*: The references are gone and the website is back to normal.
- *Day 1 to Day 2 in Torrent-ultra.com*: Day two contained hidden inputs that are script generated to change privacy settings of the user in privacy.html this may be to get access to restricted information about the user.

Security Problems (2)

- *Day 3 to Day 4 in Torrent-ultra.com* Malicious scripts gone and there is no hidden input or reference to any sort of privacy settings
- *Across 6 days in showplay.tk* This websites takes user cookies and references find better results.com a black listed website several times based on user interaction with the website to send cookie information divulging sensitive user data.

Some Interesting Facts

- Blocked internet access for performing dynamic JS analysis.

Some Interesting Facts

- Blocked internet access for performing dynamic JS analysis.

Possible Infection: Ransomware.Cerber.Downloader Ransomware.Downloader.Locky Trojan.Dov
ISSUE=539803 PROJ=5

GD GT Service Desk <servicedesk@gatech.edu>
Sun 4/16, 4:00 PM
Giri, Ravi Prakash

Attachments

Please type your response above this line. Do not delete this line of text.

Notification of Incident Registration

Workspace: GT-Service Desk
Incident: Possible Infection: Ransomware.Cerber.Downloader Ransomware.Downloader.Locky Trojan.Downloader.Donoff Trojan.Ga
Incident Number:539803

Priority: No SLA **Status:**Assigned
Date: 04/16/2017 **Time:** 16:00:07
Created By:API-OIT-IS

Some Interesting Facts

- Blocked internet access for performing dynamic JS analysis.
- Got notification emails from OIT tracing the crawler activities.

Some Interesting Facts

- Blocked internet access for performing dynamic JS analysis.
- Got notification emails from OIT tracing the crawler activities.
- During dynamic analysis of JS, encountered a script making 32 calls to `eval()` function.

Conclusion

- Content Modifications impact the classifier scores
- Fed our classifier with crawled data spanning across one month
- Find websites with the most security score changes
- Manually finding security issues

Future Work

- Seeding the classifier with larger set of data and verifying the consistency of results
- Utilizing the sandboxing environment to extract more features that can have security relevance

Acknowledgment

- We would like to thank Professor Manos Antonakakis for extending his insight and expertise to us which was integral for us in conceiving this project. We would like to offer our special thanks to the TAs Panos K and Thanos A, whose timely advised and extensive support helped keep the project on track.

Team Members

- Brenden Rauleron
- Kee Wook Lee
- Mehrdad Tahmasbi
- Nagendra Posani
- Prahathess Rengasamy
- Ravi Prakash Giri
- Swarnim Vyas
- Vishal Llewellyn Seshadri