

# Detecting client-side e-banking fraud using a heuristic model

Tim Timmermans    Jurgen Kloosterman  
tim.timmermans@os3.nl    jurgen.kloosterman@os3.nl

University of Amsterdam

July 4, 2013

- E-banking malware;
- Man-in-the-browser attack;
  - "Owns" the browser;
  - Not possible to detect malware with web techniques, i.e JavaScript.

# Normal banking web page

The screenshot displays the ABN-AMRO banking website interface. At the top left is the ABN-AMRO logo. The navigation bar includes links for 'Prive', 'Zakelijk', and 'Private banking', along with a search box and language options 'NL | EN'. A secondary menu contains 'Home', 'Betalen', 'Sparen', 'Inkomen voor later', 'Beleggen', 'Lenen', 'Hypotheken', 'Verzekeren', and 'Contact'. The main content area features a 'Mededelingen' (Messages) section with a notification: 'Goedemorgen Bfwhl laatste bezoek: woensdag 20 mrt '13, 11:25 uur' and a 'Bankmail' icon with a '2' badge. Below this is a security warning titled 'Veilig Bankieren' (Safe Banking) with the following text: 'Internetcriminelen proberen op diverse manieren klanten van Nederlandse banken op te lichten. Zo kunnen ze telefonisch contact met u opnemen en zich voordoen als bankmedewerkers. Ga dus niet in op telefonische verzoeken om bijvoorbeeld codes van uw e-identificatie te verstrekken. ABN AMRO belt regelmatig vanuit servicegerichtheid naar haar klanten, maar zal nooit vragen om persoonlijke codes en dergelijke. Hetzelfde geldt voor e-mail. Ga nooit in op verzoeken per e-mail om op een link te klikken waarbij u gevraagd wordt om codes te verstrekken. ABN AMRO zal nooit per e-mail vragen om persoonlijke codes. Controleer of het slotje in uw browser aanwezig is en u verbinding hebt met ABN AMRO. Soms is het slotje niet direct zichtbaar (Mozilla Firefox). Klik dan op het ABN AMRO logo voor het slotje en beveiligingsinformatie.' Below the text is an empty input field and an 'OK' button. The footer contains links for 'Over ABN AMRO Prive', 'Werken bij ABN AMRO', 'Veiligheid', 'Toegankelijkheid', 'Privacy en cookie beleid', 'Disclaimer', and the copyright notice '© 2012 ABN AMRO'.

Figure 1: Normal banking web page

## Servicemelding:

Geachte klant!

Tot onze spijt zijn momenteel alle servers van het Internetbankieren overbelast, waardoor u een kleine vertraging bij de toegang tot uw account en zijn functies kunt oplopen.

Wacht totdat het systeem uw aanvraag volledig uitgevoerd heeft, zodat u toegang kunt krijgen tot uw account.

Opmerking:

Het proces kan enkele minuten in beslag nemen afhankelijk van de mate van belastbaarheid van de servers van de bank.

Wij bieden u onze excuses aan voor dit tijdelijke ongemak!



Figure 2: Malicious banking web page

To what extent is it possible to detect maliciously injected code into a web page using a heuristic model in order to counteract fraud and what is the performance of such technique in terms of accuracy and execution time?

- Pattern recognition;
- Cannot detect injections from unknown malware.

- CaffeineMonkey: a method to analyse and detect malicious JavaScript (Feinstein et. al.);
- Prophiler: a filter to examine millions of web pages for malicious content (Canali, Davide, et al.);
- Zozzle: a low-overhead solution that applies Bayesian analysis to detect JavaScript malware in the browser (Curtsinger, Charlie, et al.).

# Approach (1)

- Supervised machine learning;
  - Labeling of benign and malicious pages
- Server-side detection mechanism;

**Goal:** detect injections from unknown malware and difficult to bypass.



## Approach (2)

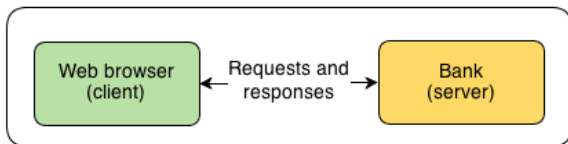


Figure 3: Normal interaction with an e-banking web site.

## Approach (3)

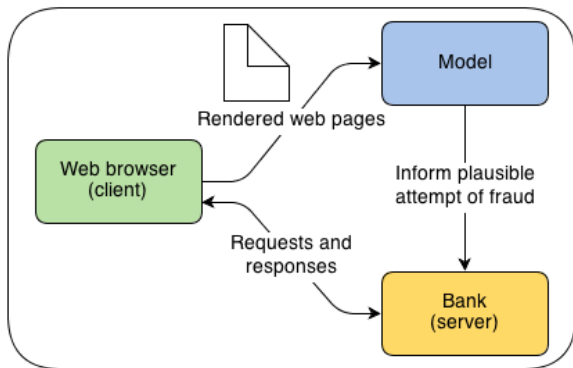


Figure 4: Overview of fraud detection implementation.

# Model overview

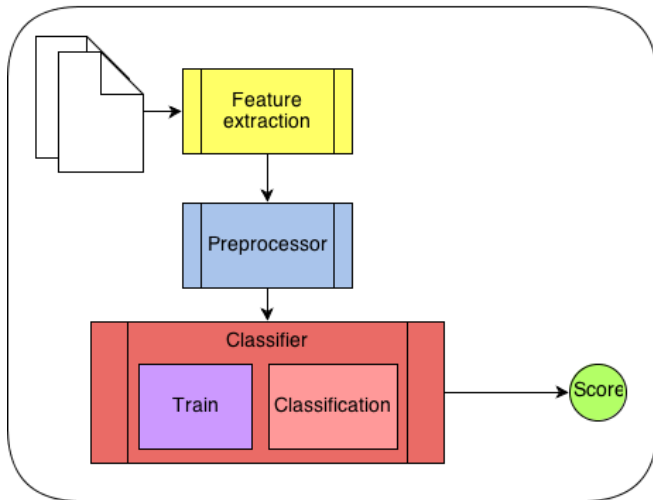


Figure 5: Overview of the fraud detection model.

# Method: feature extraction

Brief selection of features that are identified:

- iframes;
- inline styles;
- hidden elements;
- input fields;
- (obfuscated) Javascript;
- external Javascript, stylesheets and images.

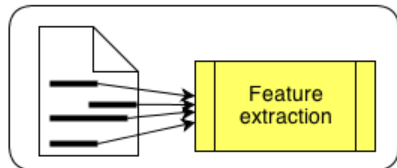


Figure 6: Feature extraction component

A total of 26 relevant features are identified from HTML, Javascript and URLs

# Method: preprocessor

- Transforms the feature data to a vector as input for the classifier;
- Assigns a maliciousness score based on the extracted URL features.

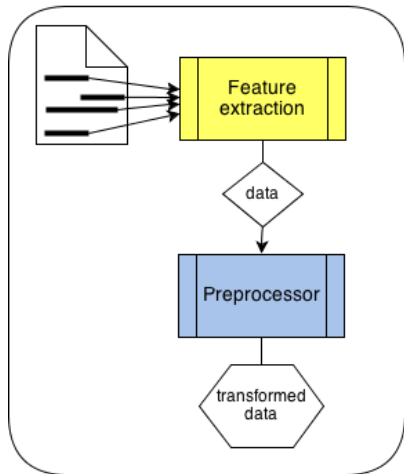


Figure 7: Preprocessor component

- Naïve Bayes learning algorithm
- Two components
  - Trainer;
  - Classification.

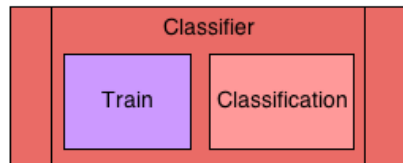


Figure 8: Classifier components

# Classifier: trainer

Train the classifier on manual labeled malicious and benign pages.

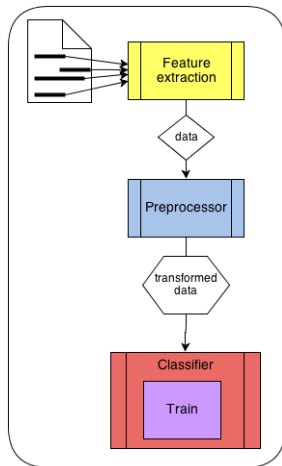


Figure 9: Classifier - trainer component

# Classifier: classification

- Classifies an unknown page against the training set using the Bayes' theorem;
- Result consists of a probability between 0 and 100% for each class.

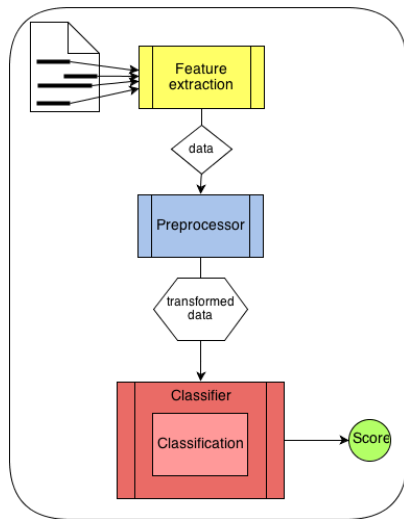


Figure 10: Classifier - classification



# Results: performance

Mean execution time to classify an unknown page: 0.176 seconds.

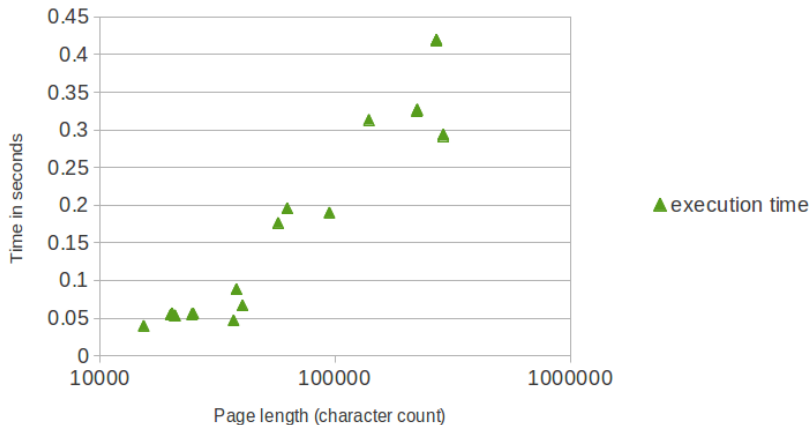


Figure 11: Execution time performance

# Results: accuracy

90% accuracy reached with  $\sim 32,000$  instances in the training set.

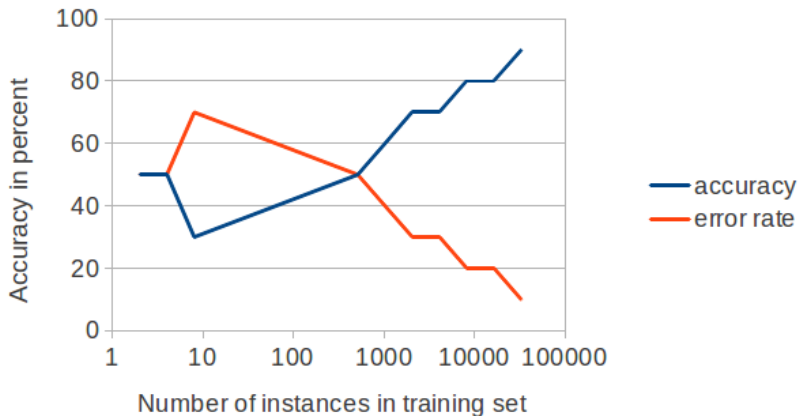


Figure 12: Accuracy measurements

Experiment to validate the developed model:

- ① Train classifier on page adapter by Zeus malware;
- ② Classify a page adapted by Citadel malware.

Result: classified as malicious with a probability of 100%.

- Classifier reaches an accuracy of 90% given the used dataset (needs validation with more complete set);
- The developed model is able to counteract fraud, caused by (unknown) malware;
- Classification process of a web page is performed with a mean of 0.176 seconds;
- Improvement of the model may lower impact on resources and optimizing executing time.

- Feinstein, Ben, Daniel Peck, and I. SecureWorks. "Caffeine monkey: Automated collection, detection and analysis of malicious javascript." Black Hat USA 2007 (2007).
- Canali, Davide, et al. "Prophiler: a fast filter for the large-scale detection of malicious web pages." Proceedings of the 20th international conference on World wide web. ACM, 2011.
- Curtsinger, Charlie, et al. "ZOZZLE: Fast and Precise In-Browser JavaScript Malware Detection." USENIX Security Symposium. 2011.