

Building a Better Botnet DGA Mousetrap

Separating Rats, Mice and Cheese in DNS Data



Josiah Hagen

Hewlett Packard Enterprise TippingPoint

Miranda Mowbray & Prasad Rao

Hewlett Packard Labs



Goals

- Classify domains names as benign or malicious
- Minimize false positive classifications of malicious
- Classify malicious domains according to DGA family

Not a Goal

- Determine which hosts are related with which malware



Solution

Training

- Cluster domains by preference
- Determine groups with matching syntactical features
- Classifiers for benign / malicious
- Feature selection
- Classifiers for family or origin
- Feature selection

Evaluation

- Unknown domains and at online
- Determine cluster and then syntactical features
- Classify & Realign / Malicious
- Classify family or origin

Syntactical Rules



Feature Vector

- Quantity everything related to using
- Cluster and find patterns the same for
- entire domains within a file



Conclusions

- Syntactical rules help
- Unlabeled data helps
- Results worse for HSV data
- Especially word based DGA FPs
- Some features are good for classifiers
- Aligns with linguistic or not (disproportionate)
- High values to compare
- Not a standalone solution
- Build classifiers for individual hosts

Building a Better Botnet DGA Mousetrap

Separating Rats, Mice and Cheese in DNS Data



Josiah Hagen

Hewlett Packard Enterprise TippingPoint

Miranda Mowbray & Prasad Rao

Hewlett Packard Labs



Goals

- Classify domains names as benign or malicious
- Minimize false positive classifications of malicious
- Classify malicious domains according to DGA family

Not a Goal

- Determine which hosts are related with which malware



Solution

Training

- Gather domains by preference
- Determine groups with matching syntactical features
- Classifiers for benign / malicious
- Feature selection
- Classifiers for family or origin
- Feature selection

Evaluation

- Unknown domains and at online
- Determine classes and fine syntactical features
- Classify & Realign / Malicious
- Classify family or origin

Syntactical Rules



Feature Vector

- Quantity everything related to using
- Classify and find patterns the same for
- entirely within a file



Conclusions

- Syntactical rules help
- Unlabeled data helps
- Results worse for HSV data
- Especially word based DGA FPs
- Some features are good for classifiers
- Aligns with linguistic or not (disjoint) word based
- High values to compare
- Not a standalone solution
- Build classifiers for individual hosts

Goals

Classify domain names as benign and malicious

Minimize false positive classifications of malicious

Classify malicious domains according to DGA family



Not a Goal

Determine which hosts are infected with which malware



Solution

Training

Gather domains by provenance

Determine groups with matching syntactical features

Classifiers for Benign / Malicious
Feature Selection

Classifiers for family or origin
Feature Selection

Evaluation

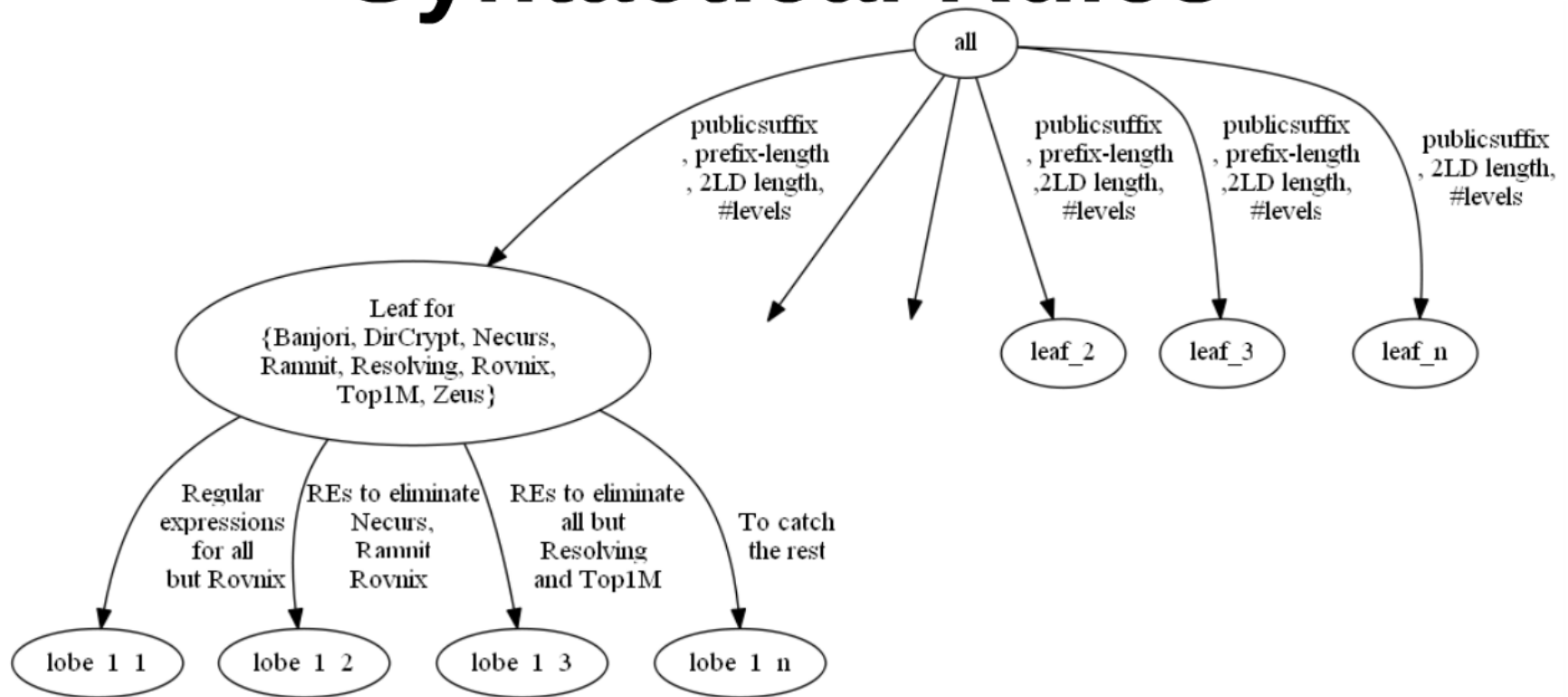
Unknown domains one at a time

Determine coarse and fine syntactical features

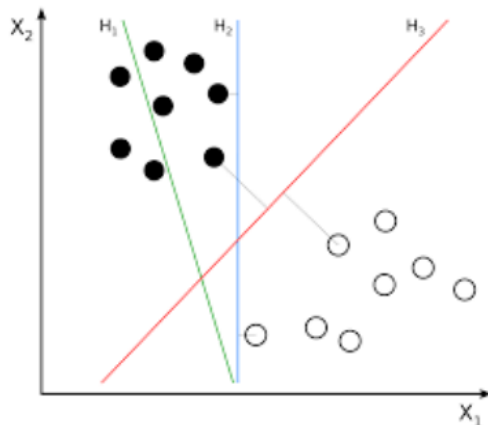
Classify if Benign / Malicious

Classify family or origin

Syntactical Rules



Feature Vector



Quantify everything about a string

Coarse and fine syntax the same for elements within a lobe

Aggregates (22)

2LD Counts	ISO-8859-1	Overall Counts
A-F	- uppercase foreign	Dots
G-Z	- lowercase foreign	Dashes
a-f	- other printable	Dots, Dashes, & Underscores
g-z	- other non-printable	
consonants		RFC 1034 Violations
vowels	doubles 'aa' - 'zz'	length > 254
digits	non-linguistic bigrams	labels not '[a-z0-9]
		empty labels '.'
		labels not '[a-z]*' invalid chars

(256) 1-grams	N-Grams (1600) 2-grams	Reduce to 40	(64000) 3-grams
Counts of characters	Counts of character pairs: 'aa' 'ab' ... 'zz'		Counts of character triples: 'aaa' 'aab' ... 'zzz'
Pros		Cons	
Separate linguistic / non-linguistic elements		Space	
Catch bias in DGA PRNG		Overfitting	

Characters by Position (10240) Forward (10240) Backward

Boolean slots for whether a given character occurs within a given position indexed from beginning and end of domain	
Pros	Cons
Classifying fixed substrings	Space
Banjori, Bankpatch, Caphaw, Web Services	Overfitting

Words

(4) 2LD	(4) Prefix
Counts of words	
Max count of non-overlapping words	
Max percentage of characters comprised of words	
Length of the longest word	
Pros	Cons
Classifying Benign vs Malicious	Time
Matsnu, Rovnix, Suppobox	Overfitting

Aggregates (22)

2LD Counts

A-F	ISO-8859-1
G-Z	• uppercase foreign
a-f	• lowercase foreign
g-z	• other printable
consonants	• other non-printable
vowels	doubles 'aa' - 'zz'
digits	non-linguistic bigrams

Overall Counts

Dots	Dashes
Dots, Dashes, & Underscores	

RFC 1034 Violations

length > 254	labels not <code>*[a-z0-9]</code>
labels > 63	empty labels <code>'..'</code>
labels not <code>[a-z].*</code>	invalid chars

x_1

N-Grams

(256)

1-grams

Counts of characters

(1600)

2-grams

Counts of character
pairs: 'aa' 'ab'... 'zz'

Reduce
to 40

(64000)

3-grams

Counts of character
triples: 'aaa', 'aab'...'zzz'

Pros

Separate linguistic / non-linguistic elements

Catch bias in DGA PRNG

Cons

Space

Overfitting

Characters by Position

(10240)

Forward

(10240)

Backward

Boolean slots for whether a given character occurs within a given position indexed from beginning and end of domain

Pros

Classifying fixed substrings

Banjori, Bankpatch, Caphaw, Web Services

Cons

Space

Overfitting

Words

(4)

2LD

Counts of words

Max count of non-overlapping words

Max percentage of characters comprised of words

Length of the longest word

Pros

Classifying Benign vs Malicious

Matsnu, Rovnix, Suppobox

(4)

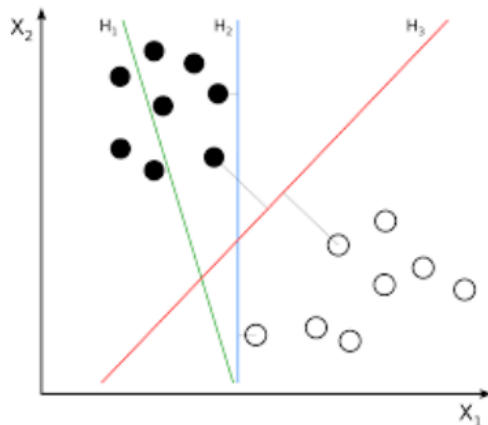
Prefix

Cons

Time

Overfitting

Feature Vector



Quantify everything about a string

Coarse and fine syntax the same for elements within a lobe

Aggregates (22)

2LD Counts	ISO-8859-1	Overall Counts
A-F	- uppercase foreign	Dots
G-Z	- lowercase foreign	Dashes
a-f	- other printable	Dots, Dashes, & Underscores
g-z	- other non-printable	
consonants		RFC 1034 Violations
vowels	doubles 'aa' - 'zz'	length > 254
digits	non-linguistic bigrams	labels not '[a-z0-9]
		empty labels '.'
		labels not '[a-z]*' invalid chars

(256) 1-grams	N-Grams (1600) 2-grams	Reduce to 40	(64000) 3-grams
Counts of characters	Counts of character pairs: 'aa' 'ab' ... 'zz'		Counts of character triples: 'aaa' 'aab' ... 'zzz'
Pros		Cons	
Separate linguistic / non-linguistic elements		Space	
Catch bias in DGA PRNG		Overfitting	

Characters by Position (10240) Forward (10240) Backward

Boolean slots for whether a given character occurs within a given position indexed from beginning and end of domain	
Pros	Cons
Classifying fixed substrings	Space
Banjori, Bankpatch, Caphaw, Web Services	Overfitting

Words (4) 2LD (4) Prefix

Counts of words	
Max count of non-overlapping words	
Max percentage of characters comprised of words	
Length of the longest word	
Pros	Cons
Classifying Benign vs Malicious	Time
Matsnu, Rovnix, Suppobax	Overfitting

Conclusions

Syntactical rules help

Unbalanced data hurts

Results worse on real data

Especially word based DGA FPs

Some features are good
for classifiers

Aggregates

Linguistic or not (bigrams)

Word based

Hash words to compress
dictionary to reduce FP

Not a standalone solution

Build classifiers for infected hosts



Syntactical rules help

Unbalanced data hurts

Results worse on real data

Conclusions

Syntactical rules help

Unbalanced data hurts

Results worse on real data

Especially word based DGA FPs

Some features are good
for classifiers

Aggregates

Linguistic or not (bigrams)

Word based

Hash words to compress
dictionary to reduce FP

Not a standalone solution

Build classifiers for infected hosts

Some features are good
for classifiers

Aggregates

Linguistic or not (bigrams)

Word based

A FPs

Hash words to compress
dictionary to reduce FP

not a standalone solution

classifiers for infected hosts



Conclusions

Syntactical rules help

Unbalanced data hurts

Results worse on real data

Especially word based DGA FPs

Some features are good
for classifiers

Aggregates

Linguistic or not (bigrams)

Word based

Hash words to compress
dictionary to reduce FP

Not a standalone solution

Build classifiers for infected hosts

Building a Better Botnet DGA Mousetrap

Separating Rats, Mice and Cheese in DNS Data



Josiah Hagen

Hewlett Packard Enterprise TippingPoint

Miranda Mowbray & Prasad Rao

Hewlett Packard Labs



Goals

- Classify domains names as benign or malicious
- Minimize false positive classifications of malicious
- Classify malicious domains according to DGA family

Not a Goal

- Determine which hosts are related with which malware



Solution

Training

- Cluster domains by preference
- Determine groups with matching syntactical features
- Classifiers for benign / malicious
- Feature selection
- Classifiers for family or origin
- Feature selection

Evaluation

- Unknown domains and at online
- Determine cluster and then syntactical features
- Classify & Realign / Malicious
- Classify family or origin

Syntactical Rules



Feature Vector

- Quantity everything related to using
- Cluster and find patterns the same for
- entire domains within a file



Conclusions

- Syntactical rules help
- Unlabeled data helps
- Results worse for HSV data
- Especially word based DGA FPs
- Some features are good for classifiers
- Algorithms
- Longer, or not (depending on word length)
- High values to compare
- Not a standalone solution
- Build classifiers for individual hosts