Regex to Machine Learning: Techniques to Get Ahead of Cyber Attackers

Evan Gaustad

Introduction

- About me
  - Cyber Security Professional and Data Scientist
  - Twitter: @cyberaitech
  - LinkedIn: evangaustad
- Success for this talk:
  - Share ideas to detect bad guys
  - Create awareness of lesser known, but powerful analytics

Introduction

- About me
  - Cyber Security Professional and Data Scientist
  - Twitter: @cyberaitech
  - LinkedIn: evangaustad
- Success for this talk:
  - Share ideas to detect bad guys
  - Create awareness of lesser known, but powerful analytics
- Failure:
  - Drag people through an hour of math and code for no reason

Phishing in the cyber kill chain

B. Zeng. (2014). “Zero-Day Attacks are not the same as Zero-Day Vulnerabilities”, FireEye
https://www.fireeye.com/blog/executive-perspective/2014/04/zero-day-attacks-are-not-the-same-as-zero-day-vulnerabilities.html

Phishing in the cyber kill chain

B. Zeng. (2014). “Zero-Day Attacks are not the same as Zero-Day Vulnerabilities”, FireEye
https://www.fireeye.com/blog/executive-perspective/2014/04/zero-day-attacks-are-not-the-same-as-zero-day-vulnerabilities.html

MITRE ATT&CK Framework Plug

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
10 items	31 items	56 items	28 items	59 items	20 items	19 items	17 items	13 items	9 items	21 items
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Password Policy Discovery	Pass the Ticket	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Permission Groups Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	InstallUtil	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Input Capture	Process Discovery	Remote Services	Input Capture		Multi-hop Proxy
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Prompt	Query Registry	Replication Through Removable Media	Man in the Browser		Multi-Stage Channels
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files or Information	Kerberoasting	Remote System Discovery	Shared Webroot	Screen Capture		Multiband Communication
	LSASS Driver	Create Account	Image File Execution Options Injection	Disabling Security Tools	Keychain	Security Software Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning	System Information Discovery	Taint Shared Content			Port Knocking
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing	System Network Configuration Discovery	Third-party Software			Remote Access Tools
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL	System Network Connections Discovery	Windows Admin Shares			Remote File Copy
	Regsvr32	File System Permissions Weakness	Plist Modification	Extra Window Memory Injection	Private Keys		Windows Remote Management			Standard Application Layer Protocol
	Rundll32	Hidden Files and Directories	Port Monitors	File Deletion	Replication Through Removable Media					Standard Cryptographic Protocol
	Scheduled Task	Hooking	Process Injection	File System Logical Offsets	Securityd Memory					Standard Non-Application Layer Protocol
	Scripting	Hypervisor		Gatekeeper Bypass	Two-Factor Authentication Interception					Uncommonly Used Port
	Service Execution	Image File Execution Options Injection		Hidden Files and Directories						
	Signed Binary Proxy Execution			Hidden Users						
	Signed Script Proxy Execution									

MITRE ATT&CK Framework Plug

Initial Access

10 items

- Drive-by Compromise
- Exploit Public-Facing Application
- Hardware Additions
- Replication Through Removable Media
- Spearphishing Attachment
- Spearphishing Link
- Spearphishing via Service
- Supply Chain Compromise
- Trusted Relationship
- Valid Accounts

Initial Access	Execution				Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
10 items	31 items				20 items	19 items	17 items	13 items	9 items	21 items
Drive-by Compromise	AppleScrip			ulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP				Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-				Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Pa				Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic D			ory	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution				Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Execution Module Lo				Exploitation for Credential Access	Password Policy Discovery	Pass the Ticket	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitatio Execution			odel	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical				Hooking	Permission Groups Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	InstallUtil				Input Capture	Process Discovery	Remote Services	Input Capture		Multi-hop Proxy
	Launchctl				Input Prompt	Query Registry	Replication Through Removable Media	Man in the Browser		Multi-Stage Channels
	Local Job			Files or	Kerberoasting	Remote System Discovery	Shared Webroot	Screen Capture		Multiband Communication
	LSASS Dri				Keychain	Security Software Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
	Mshta			ols	LLMNR/NBT-NS Poisoning	System Information Discovery	Taint Shared Content			Port Knocking
	PowerShel			acking	Network Sniffing	System Network Configuration Discovery	Third-party Software			Remote Access Tools
	Regsvcs/R				Password Filter DLL	System Network Connections Discovery	Windows Admin Shares			Remote File Copy
	Regsvr32	External Remote Services	Image File Execution Options Injection	Exploitation for Defense Evasion	Private Keys		Windows Remote Management			Standard Application Layer Protocol
	Rundll32	File System Permissions Weakness	Launch Daemon	Extra Window Memory Injection	Replication Through Removable Media					Standard Cryptographic Protocol
	Scheduled Task	Hidden Files and Directories	New Service	File Deletion	Securityd Memory					Standard Non-Application Layer Protocol
	Scripting	Hooking	Path Interception	File System Logical Offsets	Two-Factor Authentication Interception					Uncommonly Used Port
	Service Execution	Hypervisor	Plist Modification	Gatekeeper Bypass						
	Signed Binary Proxy Execution	Image File Execution Options Injection	Port Monitors	Hidden Files and Directories						
	Signed Script Proxy Execution		Process Injection	Hidden Users						

MITRE ATT&CK Plug (T1192)

Spearphishing Link

Spearphishing with a link is a specific variant of spearphishing. It is different from other forms of spearphishing in that it employs the use of links to download malware contained in email, instead of attachment malicious files to the email itself, to avoid defenses that may inspect email attachments.

All forms of spearphishing are electronically delivered social engineering targeted at a specific individual, company, or industry. In this case, the malicious emails contain links. Generally, the links will be accompanied by social engineering text and require the user to actively click or copy and paste a URL into a browser, leveraging User Execution. The visited website may compromise the web browser using an exploit, or the user will be prompted to download applications, documents, zip files, or even executables depending on the pretext for the email in the first place. Adversaries may also include links that are intended to interact directly with an email reader, including embedded images intended to exploit the end system directly or verify the receipt of an email (i.e. web bugs/web beacons).

Spearphishing Link

Technique

ID	T1192
Tactic	Initial Access
Platform	Linux, Windows, macOS
Data Sources	Packet capture, Web proxy, Email gateway, Detonation chamber, SSL/TLS inspection, DNS records, Mail server
CAPEC ID	CAPEC-163

https://attack.mitre.org/wiki/Technique/T1192
https://mitre.github.io/attack-navigator/enterprise/

Phishing Prevalence and Click Rate

- Phishing and pretexting represent 98% of social incidents and 93% of breaches.[1]
- 91% of attacks begin with phishing[2]
- On average 4% of people click on a link in a given campaign.
- Targeted phishing was able to generate up to 54% click rates[3]
- Techniques include:
  - Malicious Links in emails
  - Attachments to emails
  - Soliciting information via email

[1] Verizon Data Breach Investigations Report. (2018). http://www.verizonenterprise.com/verizon-insights-lab/dbir/2018/
[2] PhishMe (2016). "Enterprise Phishing Susceptability and Resiliency Report" https://cofense.com/enterprise-phishing-susceptibility-report
[3] Friedrich-Alexander Universitat Erlangen-Nurnberg (2016). https://www.fau.eu/2016/08/25/news/research/one-in-two-users-click-on-links-from-unknown-senders/

The lure

- Often a sense of action, urgency, or curiosity is used to entice people
- Others are crafted to the individual, sometimes creating fake online identities or websites

Common Email Subject Lines

Symantec Internet Security Threat Report. (2017). https://digitalhubshare.symantec.com/content/dam/ent/collat/reports/RPT_ISTR-Main-Report_EN.pdf

The Lure

Wed 10/26/2016 5:04 PM

Janice Daniels <oudnzzqj@young-life.ru>

FedEx Postage and shipment – payment completed [*EXTERNAL*]

To [redacted]

You forwarded this message on 10/26/2016 8:12 PM.

You a ll you that your previous online purchase h

Take y ddress details, payment details and also you

/"https:/docs.google.com/uc?export=download&id=0b-vykm_mj2smckrwoepymwnitfe
Click or tap to follow link.

https://fedex.com/pay/38045192/info.zip

Have a wonderful day,

The FedEx intl. team

S. Polley. (2017). "Dissect the Phish to Hunt Infections" SANS Reading Room. https://www.sans.org/reading-room/whitepapers/awareness/dissect-phish-hunt-infections-37587

Common Response

*"Phishers will randomize credential harvesting links sent through email, making it even more difficult to implement blocks when email Sender and Subject correlations don't detect all of the messages from a particular phishing campaign. As you search and sanitize, if you notice **URL patterns begin to emerge**, consider implementing Splunk **RegEx searches** to help fill in the gaps in your detections."*

S. Polley. (2017). "Dissect the Phish to Hunt Infections" SANS Reading Room.
https://www.sans.org/reading-room/whitepapers/awareness/dissect-phish-hunt-infections-37587

Uniform Resource Locator (URL)

https://help-protect01.000webhostapp.com:8443/payment-update.html?tab=payments&ref=settings_nav

Protocol 3rd lvl Domain 2nd lvl Domain TLD Port Query Key / Value Pairs

Uniform Resource Locator (URL)

https://help-protect01.000webhostapp.com:8443/payment-update.html?tab=payments&ref=settings_nav

Protocol | 3rd lvl Domain | 2nd lvl Domain | TLD | Port | Query | Key / Value Pairs

Host

Uniform Resource Identifier

Regular Expressions (RegEx) for Patterns

- Characters
  - [A-Za-z0-9] to match alpha-numeric characters
  - . Matches any single character
- Repeated Characters
  - * match the specified character zero or more times
  - {N, M} match the preceding character at least N and no more than M times

Wikipedia. (2018). https://en.wikipedia.org/wiki/Regular_expression

Regular Expressions (RegEx) for Patterns

- Characters
  - [A-Za-z0-9] to match alpha-numeric characters
  - . Matches any single character
- Repeated Characters
  - * match the specified character zero or more times
  - {N, M} match the preceding character at least N and no more than M times

- Examples:

[a-z]{1, 6}	would match the string “regex” but “Regex” would not
.*	matches any string
[A-Z]\.php	matches a upper case letter followed by .php

Wikipedia. (2018). https://en.wikipedia.org/wiki/Regular_expression

Common Practitioner Response

New Search

```
daysago=30 index=web_proxy \.php\?id
| regex url=".*\/[a-zA-Z0-9]{5,15}\/[a-zA-Z0-9]{5,15}\.php\?id=.*[a-zA-Z0-9]{1,15}@[a-zA-Z0-9]{1,10}\.[a-z]{2,3}$"
| table _time user url
```

url
http://pgeseeourprogress.com/r7T2y/9xVmGsBJT.php?id=
http://musee-verre.fr/7C8wDZX/6WqtQhOl0F.php?id=
http://metadroloficial-br.com/VECkg/m8HYUB.php?id=
http://gleeseason3.org/cmiRx/gTBX4su6QW.php?id=
http://coderedsecurityrgv.com/ZerQL1ntD/6upnz3k.php?id=
http://virtualfuture.ru/bRiesW/7c0D8FREPd.php?id=

S. Polley. (2017). “Dissect the Phish to Hunt Infections” SANS Reading Room.
https://www.sans.org/reading-room/whitepapers/awareness/dissect-phish-hunt-infections-37587

Text Similarity

- Regular Expressions are one tool

Text Similarity

- Regular Expressions are one tool
- …there are many others

Fig 1: String-Based Similarity Measures

W. Gomaa, A. Fahmy. (2013). "A Survey of Text Similarity Approaches". Intl Journal of Computer Applications.
https://pdfs.semanticscholar.org/5b5c/a878c534aee3882a038ef9e82f46e102131b.pdf

Defender’s Task

- Given some examples of what’s malicious, find similar malicious examples in your own environment (if they exist)

Events in the millions or billions per day

Relatively small volume of true malicious events

Which set of tools to use?

- Needle in a haystack: given a handful of examples

- Needle-stack of needles: …but I have thousands or more examples

Given a few needles, find a needle in a haystack

- Given a handful of examples, find similar examples in another environment
- String similarity
  - Levenshtein
  - Jaro
- Distance / similarity measurements
  - Cosine
  - Euclidean

String Distance / Similarity Measurements

- Levenshtein Distance ("Edit Distance")
  - String metric representing the minimum number of edits e.g. insertions, deletions, and substitutions required to turn one string into another.

- Jaro Similarity
  - Measures several aspects of string similarity:
    - Number of matching characters
    - String length
    - Sequence of characters and transpositions

Wikipedia. (2018). https://en.wikipedia.org/wiki/Levenshtein_distance

String Distance / Similarity Measurements

String1	String2	Levenshtein	Jaro	Note
google.com	google.com	0	1.0	Exact match
google.com	g00gle.com	2	0.867	Replace letter o with number zero
google.com	google.badguy.com	7	0.863	Subdomain mimicking legitimate registered domain
google.com	malware.ru	9	0.533	Completely unrelated

Wikipedia. (2018). https://en.wikipedia.org/wiki/Levenshtein_distance

Wikipedia. (2018). https://en.wikipedia.org/wiki/Jaro%E2%80%93Winkler_distance

Text to Numeric Vector

- Many techniques only work with numbers as input, rather than text
- This requires transforming words into numbers

Word Counts

badsite.pl/index.php?pl

URL #1	
Term	Term Count
badsite	1
index	1
pl	2
php	1

words in this sentence: 5

badsite.br/index?click=br&br=click

URL #2	
Term	Term Count
badsite	1
index	1
click	2
br	3

term in this URL: 7

String Distance Measurements

- Text > Numbers > Cosine Similarity or Euclidean Distance

Term Frequency-Inverse Document Frequency

badsite.pl/index.php?pl

URL #1	
Term	**Term Count**
badsite	1
index	1
pl	2
php	1

words in this sentence: 5

badsite.br/index?click=br&br=click

URL #2	
Term	**Term Count**
badsite	1
index	1
click	2
br	3

term in this URL: 7

Term Frequency-Inverse Document Frequency

badsite.pl/index.php?pl

URL #1	
Term	**Term Count**
badsite	1
index	1
pl	2
php	1

words in this sentence: 5

URL #1	
Term	**TF**
badsite	1/5 = .2
index	1/5 = .2
pl	2/5 = .4
php	1/5 = .2

badsite.br/index?
click=br&br=click

URL #2	
Term	**Term Count**
badsite	1
index	1
click	2
br	3

term in this URL: 7

URL #2	
Term	**TF**
badsite	1/7 = 0.14
index	1/7 = 0.14
click	2/7 = .29
br	3/7 = .43

Term Frequency-Inverse Document Frequency

badsite.pl/index.php?pl

URL #1	
Term	**Term Count**
badsite	1
index	1
pl	2
php	1

words in this sentence: 5

URL #1		
Term	**TF**	**IDF**
badsite	1/5 = .2	log (2 / 2) = 0
index	1/5 = .2	log (2 / 2) = 0
pl	2/5 = .4	log (2 / 1) = 0.301
php	1/5 = .2	log (2 / 1) = 0.301

IDF = $\log_e$(# docs / # docs with word)

badsite.br/index?
click=br&br=click

URL #2	
Term	**Term Count**
badsite	1
index	1
click	2
br	3

term in this URL: 7

URL #2		
Term	**TF**	**IDF**
badsite	1/7 = 0.14	log (2 / 2) = 0
index	1/7 = 0.14	log (2 / 2) = 0
click	2/7 = .29	log (2 / 1) = 0.301
br	3/7 = .43	log (2 / 1) = 0.301

Term Frequency-Inverse Document Frequency

badsite.pl/index.php?pl

URL #1	
Term	Term Count
badsite	1
index	1
pl	2
php	1

words in this sentence: 5

URL #1			
Term	TF	IDF	TF*IDF
badsite	1/5 = .2	log (2 / 2) = 0	0
index	1/5 = .2	log (2 / 2) = 0	0
pl	2/5 = .4	log (2 / 1) = 0.301	0.1204
php	1/5 = .2	log (2 / 1) = 0.301	0.0602

IDF = $\log_e$(# docs / # docs with word)

badsite.br/index?
click=br&br=click

URL #2	
Term	Term Count
badsite	1
index	1
click	2
br	3

term in this URL: 7

URL #2			
Term	TF	IDF	TF*IDF
badsite	1/7 = 0.14	log (2 / 2) = 0	0
index	1/7 = 0.14	log (2 / 2) = 0	0
click	2/7 = .29	log (2 / 1) = 0.301	0.0873
br	3/7 = .43	log (2 / 1) = 0.301	0.1294

Term Frequency-Inverse Document Frequency

badsite.pl/index.php?pl

URL #1	
Term	Term Count
badsite	1
index	1
pl	2
php	1

words in this sentence: 5

URL #1			
Term	TF	IDF	TF*IDF
badsite	1/5 = .2	log (2 / 2) = 0	0
index	1/5 = .2	log (2 / 2) = 0	0
pl	2/5 = .4	log (2 / 1) = 0.301	0.1204
php	1/5 = .2	log (2 / 1) = 0.301	0.0602

IDF = $\log_e$(# docs / # docs with word)

badsite.br/index?click=br&br=click

URL #2	
Term	Term Count
badsite	1
index	1
click	2
br	3

term in this URL: 7

URL #2			
Term	TF	IDF	TF*IDF
badsite	1/7 = 0.14	log (2 / 2) = 0	0
index	1/7 = 0.14	log (2 / 2) = 0	0
click	2/7 = .29	log (2 / 1) = 0.301	0.0873
br	3/7 = .43	log (2 / 1) = 0.301	0.1294

Text to Numeric Vector

badsite.pl/index.php?pl

URL #1	
Term	**TF*IDF**
badsite	0
index	0
pl	0.1204
php	0.0602

badsite.br/index?click=br&br=click

URL #2	
Term	**TF*IDF**
badsite	0
index	0
click	0.0873
br	0.1294

URL	badsite	index	pl	php	click	br
#1	0	0	0.1204	0.0602	0	0
#2	0	0	0	0	0.0873	0.1294

Given a needle stack of needles, find a needle in a haystack

- Given many examples of “benign” and “malicious”, find “bad” examples in another environment
- Text Classification (“benign” or “malicious”)
  - Naïve Bayes
  - Logistic Regression
  - Decision Trees
  - Random Forest
  - Neural Networks
  - and more…

Naïve Bayes Example

Corpus

1. I loved the movie
2. I hated the movie
3. A great movie. A good movie.
4. Poor acting
5. Great acting, a good movie

Doc	I	loved	the	movie	hated	a	great	poor	acting	good	label
#1	1	1	1	1							+
#2	1		1	1	1						-
#3				2		1	1			1	+
#4								1	1		-
#5				1		1	1		1	1	+

Example: What's the value of "I hated the poor acting"?

$V(+) = P(+)P(I|+)P(hated|+)P(the|+)P(poor|+)P(acting|+) = 6.05*10^{-7}$

$V(-) = P(-)\ P(I|-)P(hated|-)P(the|-)P(poor|-)P(acting|-) = 1.22*10^{-5}$

Francisco Iacobelli. (2015). Text Classification Using Naïve Bayes
https://www.youtube.com/watch?v=EGKeC2S44Rs

URL Retrieval: Thug (honeyclient)

```
# Pull Docker Image
docker pull honeynet/thug

# Run container, mounting local logs directory
docker run -it -v ${PWD}/logs:/logs honeynet/thug

# Inside container run thug, with desired options
python /opt/thug/src/thug.py -u win7ie100 -T 300 -w 5000 -FZ https://www.google.com
```

Benefits:

- Rich logging (e.g. all redirects and additional analysis)
- Can identify attempted exploitation
- Multiple “personalities”
- …and more

https://buffer.github.io/thug/doc/docker.html

URL Retrieval: sURLi

```
# docker run --rm -v ${PWD}/data:/surli/results -it surli -u
https://www.google.com
```

Benefits:

- Final rendered page content
- Zip encrypted content, pw: “infected”
- Final rendered screenshot of page
- Browser logs of each intermediate step of URL retrieval

https://github.com/egaus/sURLi

Questions?

Data, Code, Slides:

https://github.com/egaus/malicious_url_analysis

@cyberaitech

Naïve Bayes Example

Given these definitions…

N_{k+} = # times word K occurs in the + class

N_{k-} = # times word K occurs in the - class

|vocab| = # unique words in vocabulary = 10

$P_+ = 3 / 5 = .6$

$P_- = 2 / 5 = .4$

n_+ = sum of word counts for that class = 14

n_- = sum of word counts for that class = 6

W_k = a word, like "loved"

…we get the probability of any word for it's label

$P(W_k \mid +) = (N_{k+} + 1) / (n_+ + |vocab|)$

$P(W_k \mid -) = (N_{k-} + 1) / (n_- + |vocab|)$

Example for word "good":

$P(\text{"good"} \mid -) = (N_{k-} + 1) / (n_- + |vocab|)$

$= 0 + 1 / 6 + 10 = 1 / 16 = 0.0625$

$P(\text{"good"} \mid +) = (N_{k+} + 1) / (n_+ + |vocab|)$

$= (2 + 1) / (14 + 10) = 3 / 24 = 0.125$

Example: What's the value of "I hated the poor acting"?

$V(+) = P(+)P(I|+)P(hated|+)P(the|+)P(poor|+)P(acting|+) = 6.05*10^{-7}$

$V(-) = P(-)\ P(I|-)P(hated|-)P(the|-)P(poor|-)P(acting|-) = 1.22*10^{-5}$

*Based on the Naïve Bayes model the sentence is more likely to be negative (-), since $1.22 * 10^{-5}$ is bigger than $6.05*10^{-7}$.