Cloakify Workshop

Practical Text-Based Steganography:
Exfiltrating Data from Secure Networks and Socially Engineering SecOps Analysts

Hiding Data In Plain Sight Using Text-Based Steganography

TryCatchHCF (Joe Gervais)
Twitter: @TryCatchHCF

Presenter Background

Principal InfoSec Engineer / Lead Pentester / AppSec Lead

25+ years security- & software engineering (mostly Gov't / DoD sector)

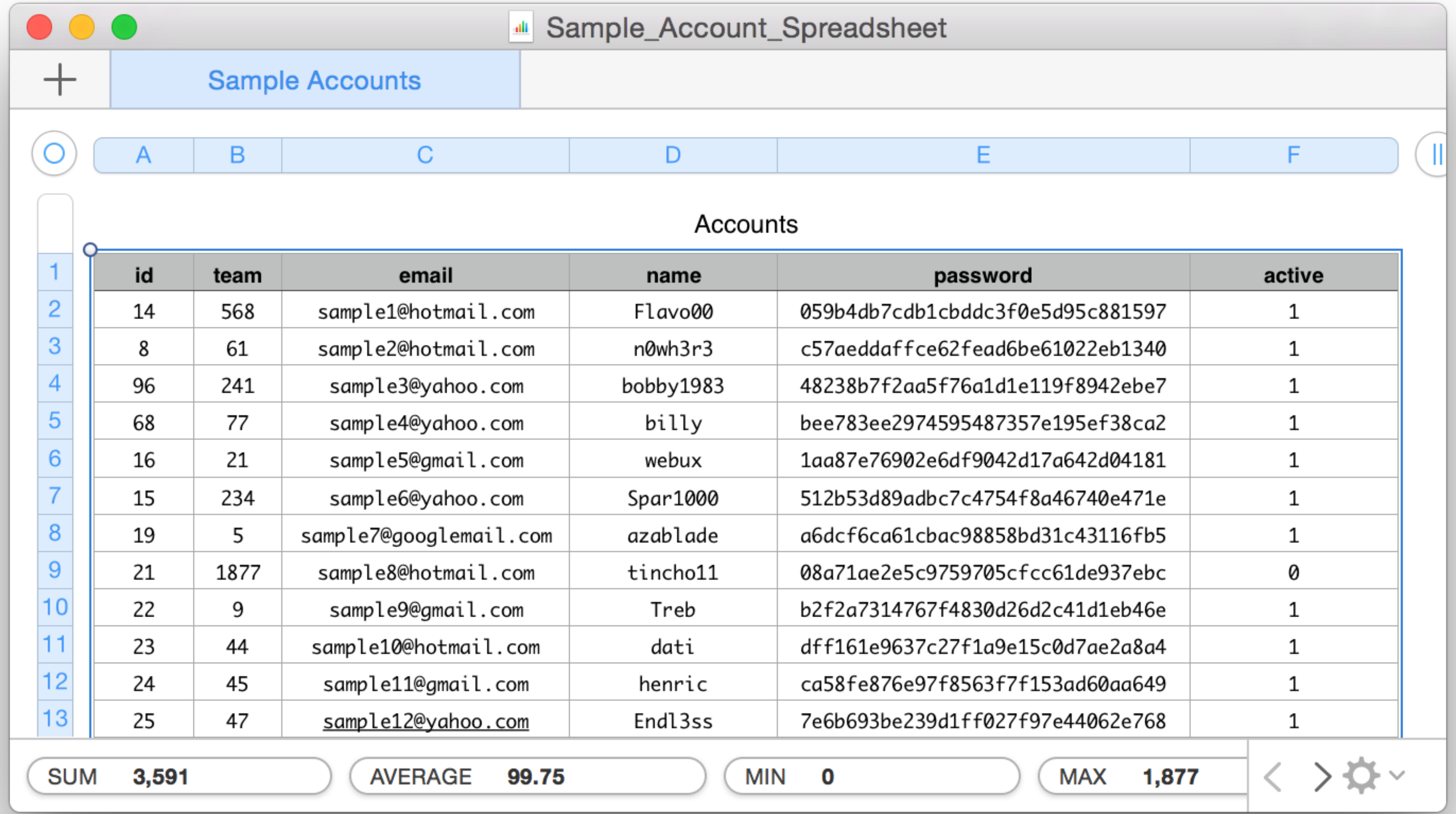
Former USMC intelligence analyst, counterintelligence specialist

Bachelors - Cognitive Science, Masters - Information Assurance

Certs - Various Acronyms, some of which hit me up for money each year

Exfiltrating Data From Secure Networks

Data Loss Prevention (DLP) systems, Multi Level Security (MLS) devices, and human analysts know what data to look out for



Accounts

id	team	email	name	password	active
14	568	sample1@hotmail.com	Flavo00	059b4db7cdb1cbddc3f0e5d95c881597	1
8	61	sample2@hotmail.com	n0wh3r3	c57aeddaffce62fead6be61022eb1340	1
96	241	sample3@yahoo.com	bobby1983	48238b7f2aa5f76a1d1e119f8942ebe7	1
68	77	sample4@yahoo.com	billy	bee783ee2974595487357e195ef38ca2	1
16	21	sample5@gmail.com	webux	1aa87e76902e6df9042d17a642d04181	1
15	234	sample6@yahoo.com	Spar1000	512b53d89adbc7c4754f8a46740e471e	1
19	5	sample7@googlemail.com	azablade	a6dcf6ca61cbac98858bd31c43116fb5	1
21	1877	sample8@hotmail.com	tincho11	08a71ae2e5c9759705cfcc61de937ebc	0
22	9	sample9@gmail.com	Treb	b2f2a7314767f4830d26d2c41d1eb46e	1
23	44	sample10@hotmail.com	dati	dff161e9637c27f1a9e15c0d7ae2a8a4	1
24	45	sample11@gmail.com	henric	ca58fe876e97f8563f7f153ad60aa649	1
25	47	sample12@yahoo.com	Endl3ss	7e6b693be239d1ff027f97e44062e768	1

SUM 3,591

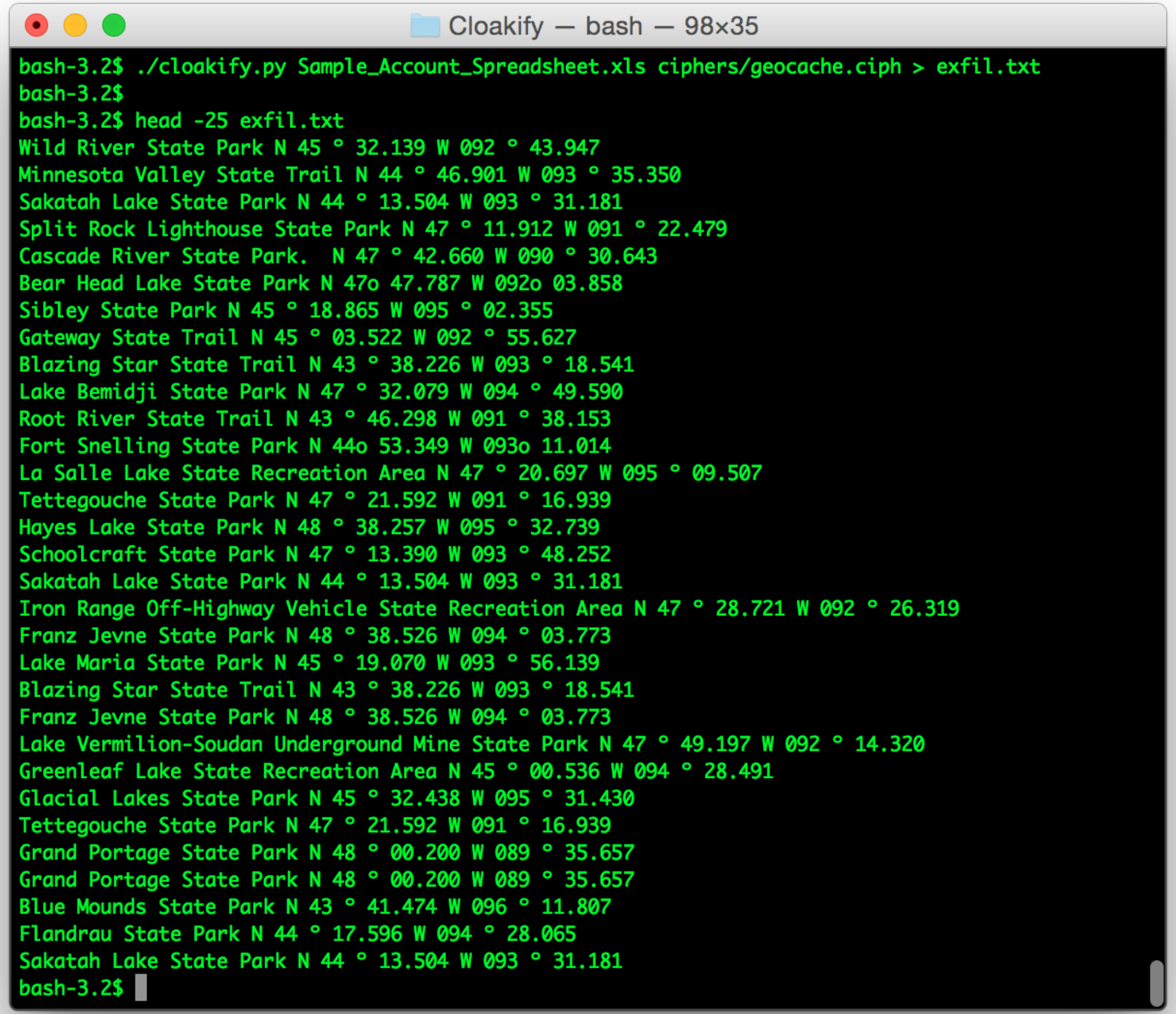
AVERAGE 99.75

MIN 0

MAX 1,877

Exfiltrating Data From Secure Networks

So I turn that data into something they're not looking for



```
bash-3.2$ ./cloakify.py Sample_Account_Spreadsheet.xls ciphers/geocache.ciph > exfil.txt
bash-3.2$
bash-3.2$ head -25 exfil.txt
Wild River State Park N 45 ° 32.139 W 092 ° 43.947
Minnesota Valley State Trail N 44 ° 46.901 W 093 ° 35.350
Sakatah Lake State Park N 44 ° 13.504 W 093 ° 31.181
Split Rock Lighthouse State Park N 47 ° 11.912 W 091 ° 22.479
Cascade River State Park.  N 47 ° 42.660 W 090 ° 30.643
Bear Head Lake State Park N 47o 47.787 W 092o 03.858
Sibley State Park N 45 ° 18.865 W 095 ° 02.355
Gateway State Trail N 45 ° 03.522 W 092 ° 55.627
Blazing Star State Trail N 43 ° 38.226 W 093 ° 18.541
Lake Bemidji State Park N 47 ° 32.079 W 094 ° 49.590
Root River State Trail N 43 ° 46.298 W 091 ° 38.153
Fort Snelling State Park N 44o 53.349 W 093o 11.014
La Salle Lake State Recreation Area N 47 ° 20.697 W 095 ° 09.507
Tettegouche State Park N 47 ° 21.592 W 091 ° 16.939
Hayes Lake State Park N 48 ° 38.257 W 095 ° 32.739
Schoolcraft State Park N 47 ° 13.390 W 093 ° 48.252
Sakatah Lake State Park N 44 ° 13.504 W 093 ° 31.181
Iron Range Off-Highway Vehicle State Recreation Area N 47 ° 28.721 W 092 ° 26.319
Franz Jevne State Park N 48 ° 38.526 W 094 ° 03.773
Lake Maria State Park N 45 ° 19.070 W 093 ° 56.139
Blazing Star State Trail N 43 ° 38.226 W 093 ° 18.541
Franz Jevne State Park N 48 ° 38.526 W 094 ° 03.773
Lake Vermilion-Soudan Underground Mine State Park N 47 ° 49.197 W 092 ° 14.320
Greenleaf Lake State Recreation Area N 45 ° 00.536 W 094 ° 28.491
Glacial Lakes State Park N 45 ° 32.438 W 095 ° 31.430
Tettegouche State Park N 47 ° 21.592 W 091 ° 16.939
Grand Portage State Park N 48 ° 00.200 W 089 ° 35.657
Grand Portage State Park N 48 ° 00.200 W 089 ° 35.657
Blue Mounds State Park N 43 ° 41.474 W 096 ° 11.807
Flandrau State Park N 44 ° 17.596 W 094 ° 28.065
Sakatah Lake State Park N 44 ° 13.504 W 093 ° 31.181
bash-3.2$
```

Exfiltrating Data From Secure Networks

Breaches are common enough that public is suffering “breach burnout”

Government and industry are rolling out the usual suspects in attempts to better control data exfiltration

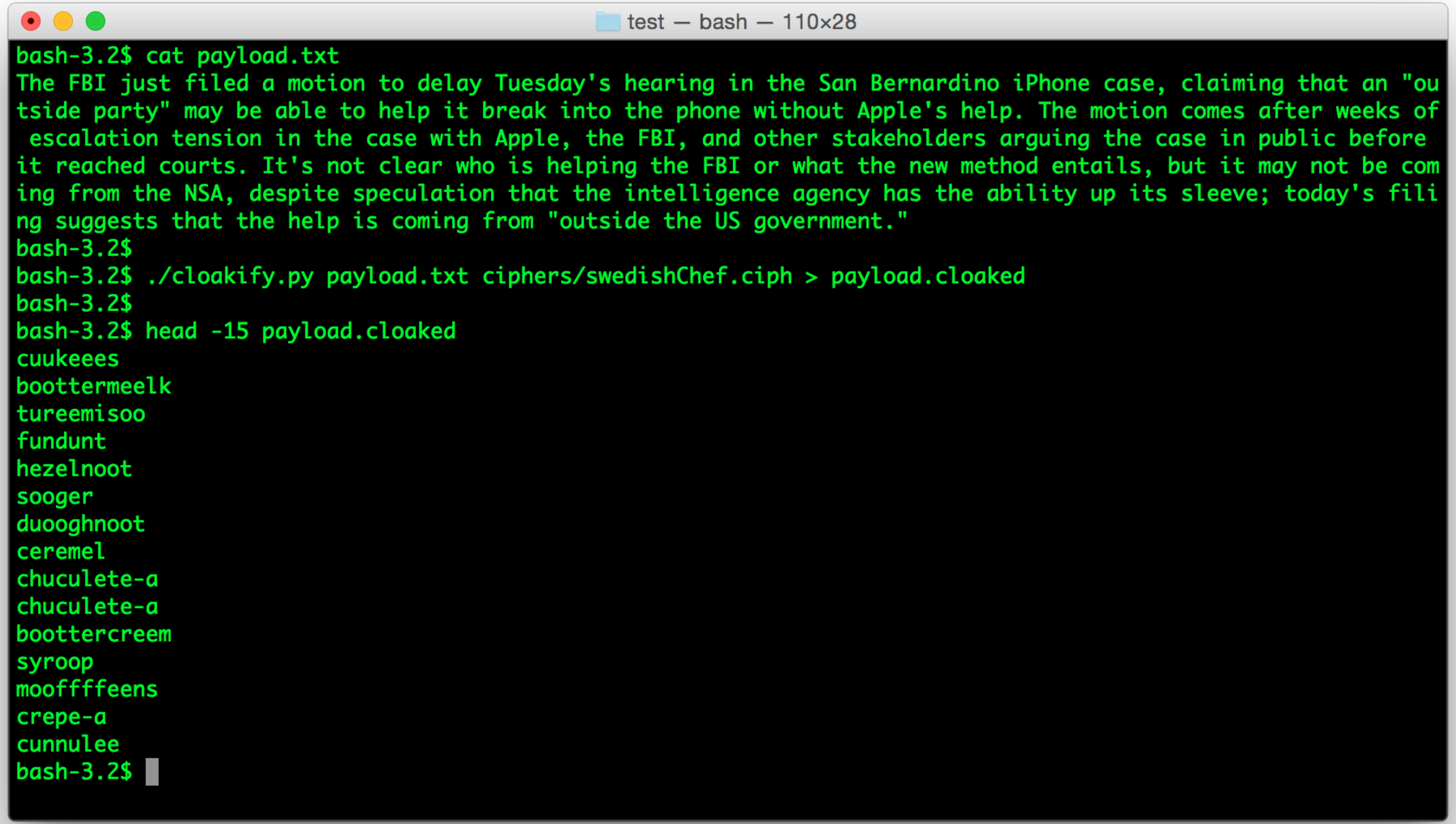
Solutions being put in place are generally addressing the obvious

- AV / Malware Detection (Try to block malicious tool use)
- Port / Protocol Restrictions (Prevent unmonitored dataflows)
- Blacklisting data (Stop dataflows containing targeted content)
- Whitelisting data (Permit only dataflows conforming to specific content)
- Manual review of data transfer by analysts

Exfiltrating Data From Secure Networks

It turns out all of those controls can be defeated by turning any filetype into a list of arbitrary strings

Perhaps a list of desserts… As pronounced by a Muppet…



```
bash-3.2$ cat payload.txt
The FBI just filed a motion to delay Tuesday's hearing in the San Bernardino iPhone case, claiming that an "ou
tside party" may be able to help it break into the phone without Apple's help. The motion comes after weeks of
 escalation tension in the case with Apple, the FBI, and other stakeholders arguing the case in public before
it reached courts. It's not clear who is helping the FBI or what the new method entails, but it may not be com
ing from the NSA, despite speculation that the intelligence agency has the ability up its sleeve; today's fili
ng suggests that the help is coming from "outside the US government."
bash-3.2$
bash-3.2$ ./cloakify.py payload.txt ciphers/swedishChef.ciph > payload.cloaked
bash-3.2$
bash-3.2$ head -15 payload.cloaked
cuukeees
boottermeelk
tureemisoo
fundunt
hezelnoot
sooger
duooghnoot
ceremel
chuculete-a
chuculete-a
boottercreem
syroop
mooffffeens
crepe-a
cunnulee
bash-3.2$
```

Exfiltrating Data From Secure Networks

Data Exfiltration & Text-Based Steganography

Exfiltrating Data From Secure Networks

Data exfiltration is crucial to the success of any full scope pentest / Red Team engagement

(“I got in”) < (“I got this out”)

Data Loss Prevention (DLP) tools and Information System (IS) configurations continue to improve, increasing chances that exfiltration attempts will generate alerts and potentially be blocked altogether

Particularly true of hardened subnets and facilities, which may use network devices that enforce whitelisted data transfers, locked-down ports / services / apps, physically blocked / disabled USB ports, etc.

Think Multilevel Security (MLS) devices connecting networks of differing security levels

Exfiltrating Data From Secure Networks

Categories of Data Exfiltration Vectors

- Standard data transfer (ncat w/ ssl, HTTP/HTTPS, usual suspects)
- Out-of-band (e.g. cellphone networks)
- Obscure channels (Cookies, DNS fields, Side Channel Semaphore)
- Physical (e.g. USB keys, photography)

This presentation focuses on the first category, standard data transfer, in combination with the use of text-based steganography as a social engineering vector

Exfiltrating Data From Secure Networks

Many advantages to the standard data transfer approach

- Does not depend on the presence of any particular port, protocol, or app
- Allows maximum flexibility on the part of the attacker
  - “Locally Sourced Services for Ecological Exfiltration”
- Avoid having to infiltrate and install additional tools (reduces risk of HIDS/NIDS alert)
- No need to infiltrate physical devices (e.g. cellular device, USB key) onto targeted subnet

Many advantages to text-based steganography

- Text is a universally transferrable data format
- Tailor the cipher so that cloaked data conforms to whitelisted / common traffic
  - Evade DLP sensors, prevent untimely alerts
  - Difficult to predict and profile the cloaked data, no signatures
  - Bypass data whitelisting controls by giving MLS exactly what it wants to see
- Use the steganographic cipher as a form of social engineering attack vs. analyst

Disadvantages of text-based steganography

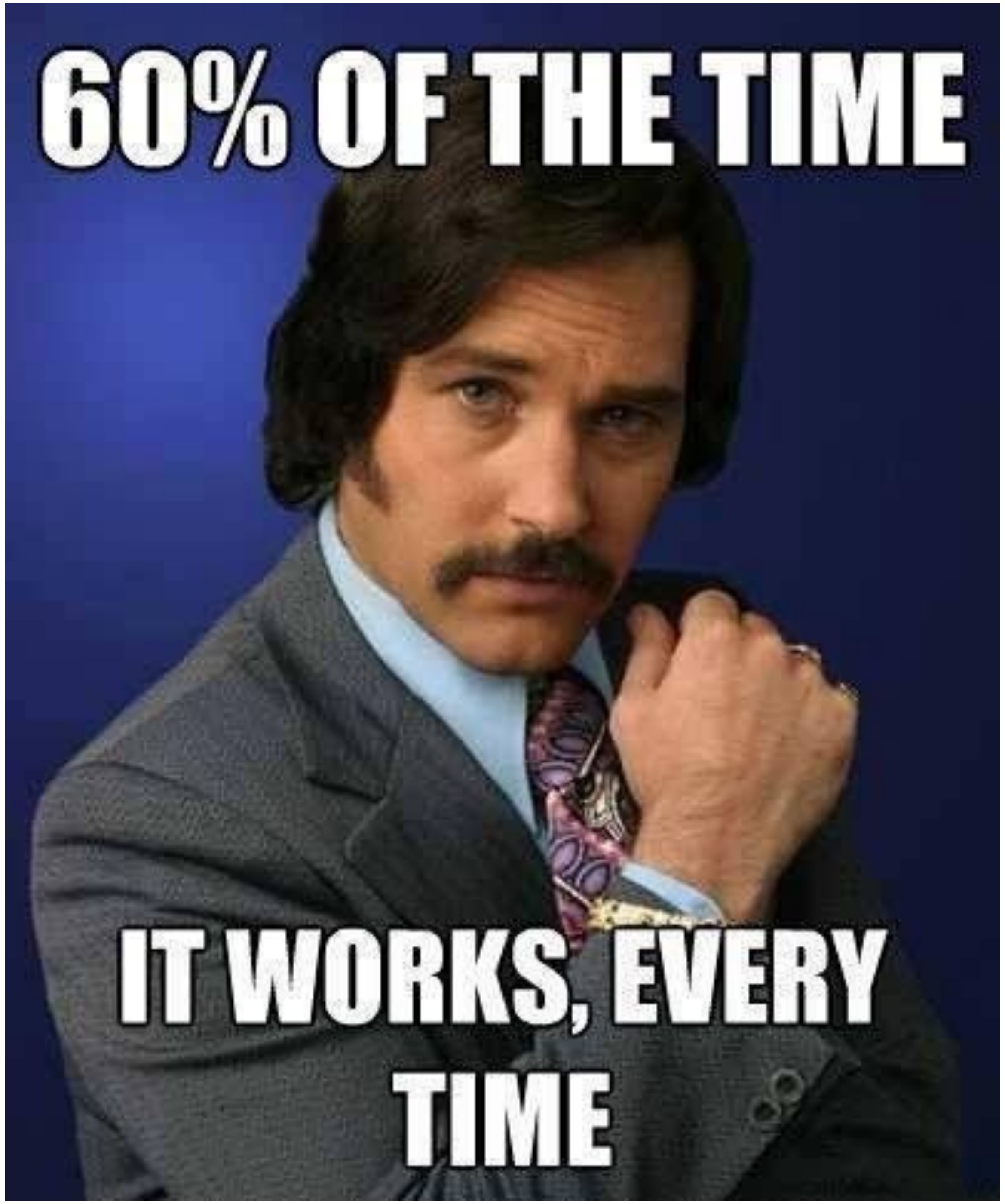
- Increases payload size (directly proportional to size of cipher elements)

Exfiltrating Data From Secure Networks

Pentesters: “Why all the effort when a simple ncat transfer works all of the time?”

- Because it doesn’t work all of the time

Exfiltrating Data From Secure Networks



Exfiltrating Data From Secure Networks

Pentesters: “Why all the effort when a simple ncat transfer works all of the time?”

- Apps may be removed from hardened systems
- Firewall rules blocking your favorite ports and protocols
- Hardened networks with well positioned and configured IDS generate untimely and inconvenient alerts
- MLS device enforcing whitelisted data for cross-network transfers
- MitM inspection of encrypted traffic traversing target network
- Blue team getting bored with same scenarios
- FUN!

Exfiltrating Data From Secure Networks

Cloakify Toolset Demonstration

Exfiltrating Data From Secure Networks

https://github.com/TryCatchHCF/Cloakify

cloakify.py - Obscure data prior to exfiltration
decloakify.py - Decode exfiltrated data

Very simple code, powerful concept

Transfer data / files / binaries of any format across a secure network's perimeter without DLP triggering alerts; Bypass data whitelisting controls; Evade AV; Derail review by analysts

First the script base64-encodes the payload file, then applies a cipher to generate a list of strings representing characters in the Base64 payload

Not a secure encryption scheme

- Vulnerable to frequency analysis attacks (use scripts in 'noiseTools' directory to add entropy)
- Encrypt data separately prior to cloaking to keep secure

Very small, simple, clean, portable - written in Python (2.7.x)

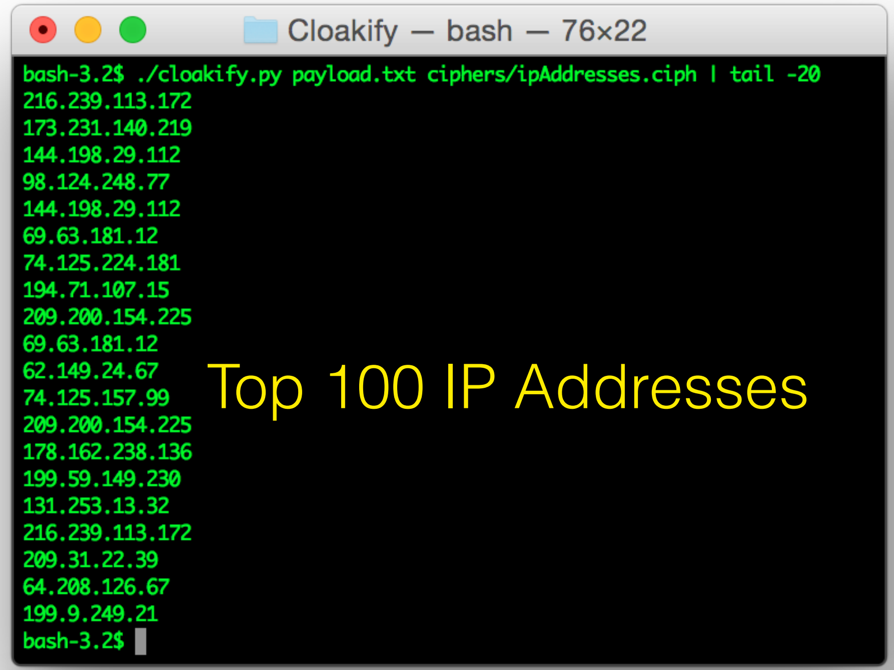
- Can quickly type into a target's local terminal if needed

Use py2exe if Windows target lacks Python

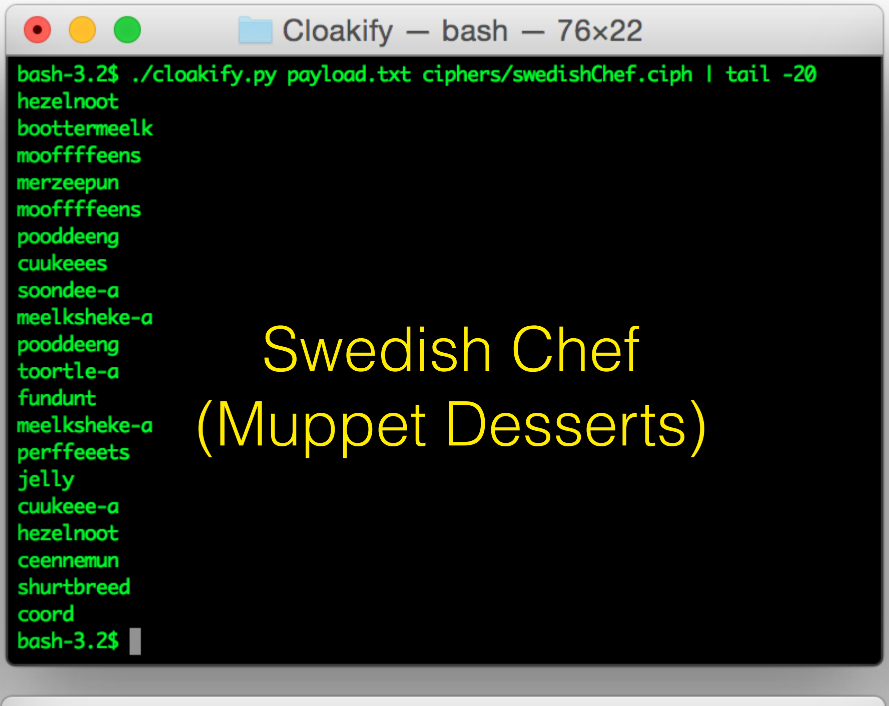
- http://www.py2exe.org/

Exfiltrating Data From Secure Networks

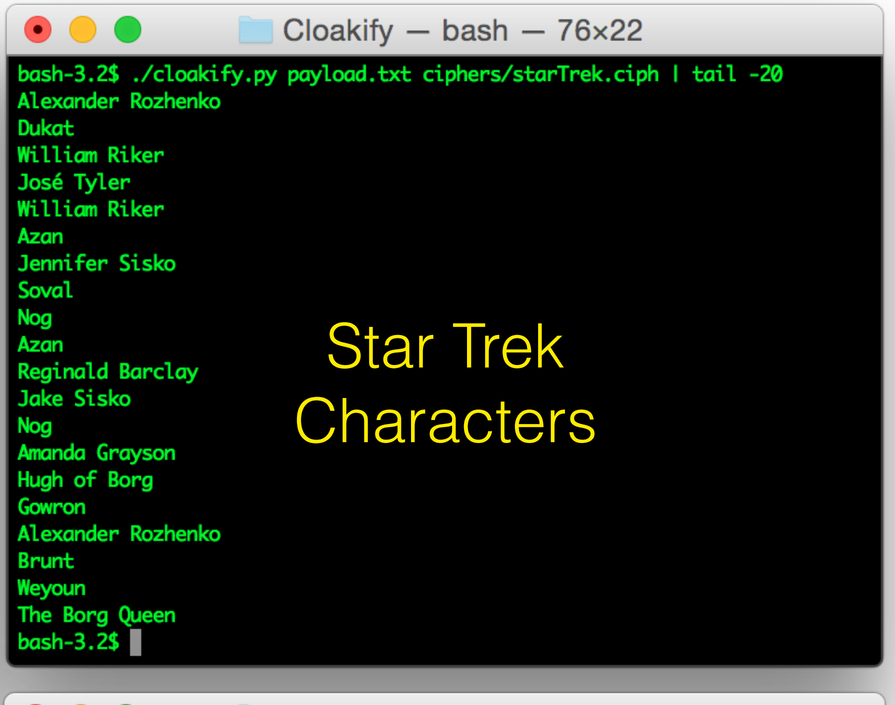
Preinstalled Ciphers Sampler: Collect the whole set, win valuable prizes!



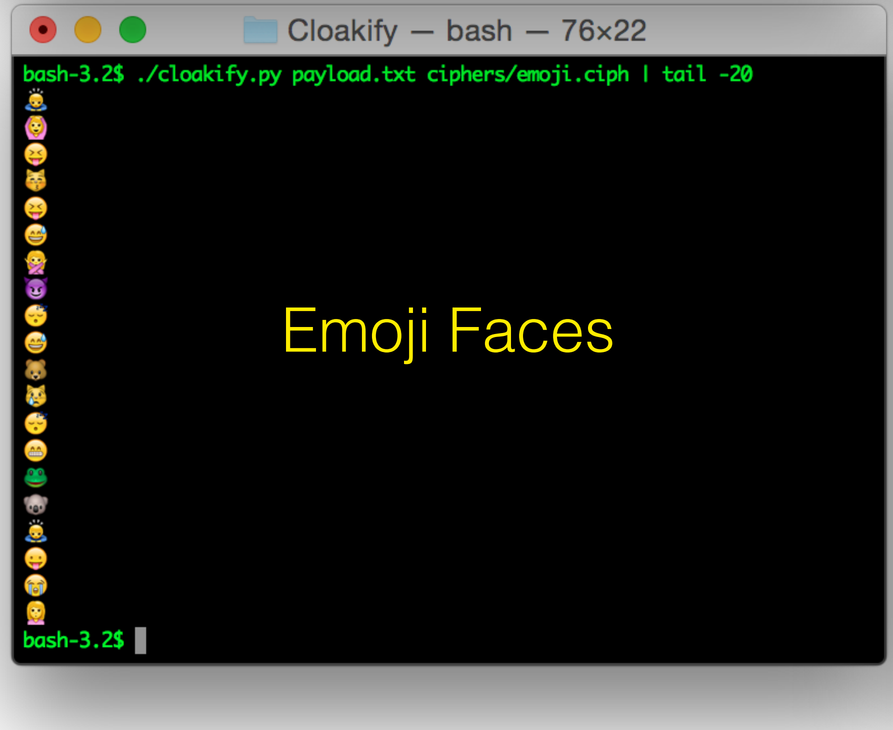
```
bash-3.2$ ./cloakify.py payload.txt ciphers/ipAddresses.ciph | tail -20
216.239.113.172
173.231.140.219
144.198.29.112
98.124.248.77
144.198.29.112
69.63.181.12
74.125.224.181
194.71.107.15
209.200.154.225
69.63.181.12
62.149.24.67
74.125.157.99
209.200.154.225
178.162.238.136
199.59.149.230
131.253.13.32
216.239.113.172
209.31.22.39
64.208.126.67
199.9.249.21
bash-3.2$
```



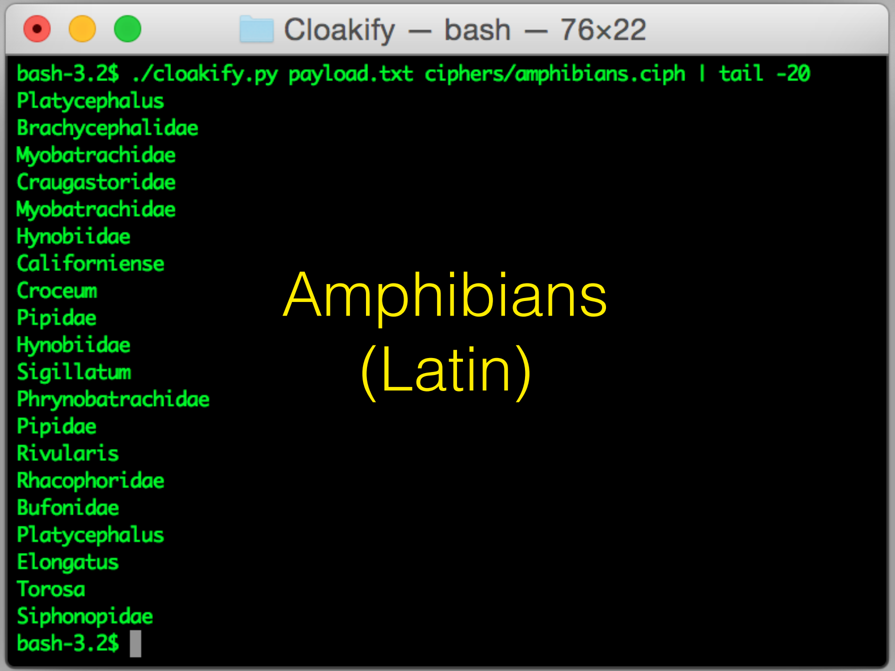
```
bash-3.2$ ./cloakify.py payload.txt ciphers/swedishChef.ciph | tail -20
hezelnoot
boottermeelk
mooffffeens
merzeepun
mooffffeens
pooddeeng
cuukeees
soondee-a
meelksheke-a
pooddeeng
toortle-a
fundunt
meelksheke-a
perffeeets
jelly
cuukeee-a
hezelnoot
ceennemun
shurtbreed
coord
bash-3.2$
```



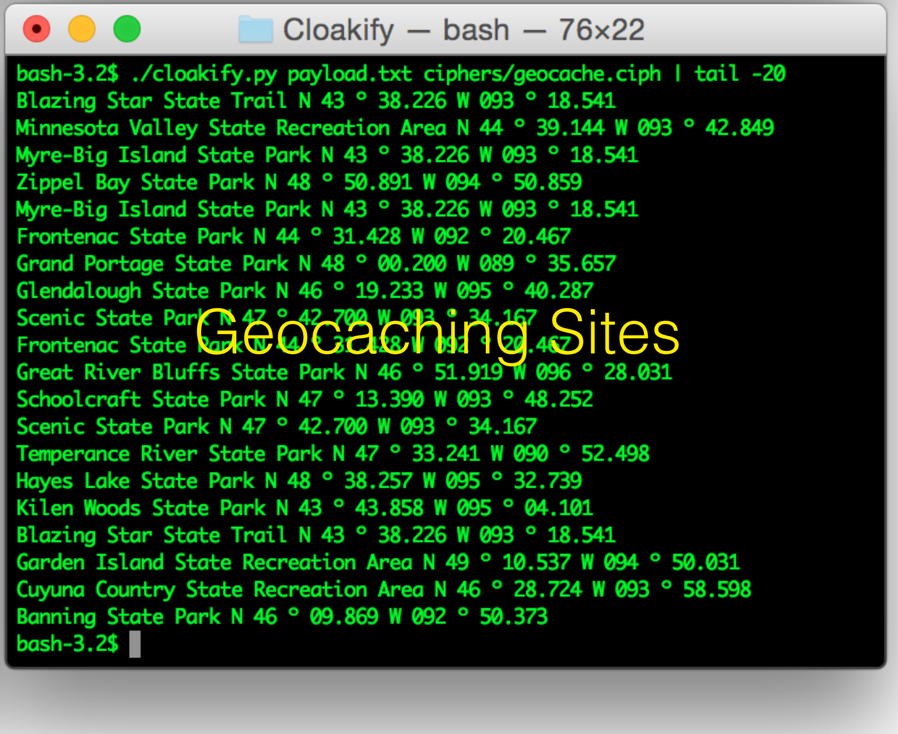
```
bash-3.2$ ./cloakify.py payload.txt ciphers/starTrek.ciph | tail -20
Alexander Rozhenko
Dukat
William Riker
José Tyler
William Riker
Azan
Jennifer Sisko
Soval
Nog
Azan
Reginald Barclay
Jake Sisko
Nog
Amanda Grayson
Hugh of Borg
Gowron
Alexander Rozhenko
Brunt
Weyoun
The Borg Queen
bash-3.2$
```



```
bash-3.2$ ./cloakify.py payload.txt ciphers/emoji.ciph | tail -20
🙇
🙆
😝
😽
😝
😅
🙅
😈
😴
😅
🐻
😿
😴
😬
🐸
🐨
🙇
😛
😭
👱
bash-3.2$
```



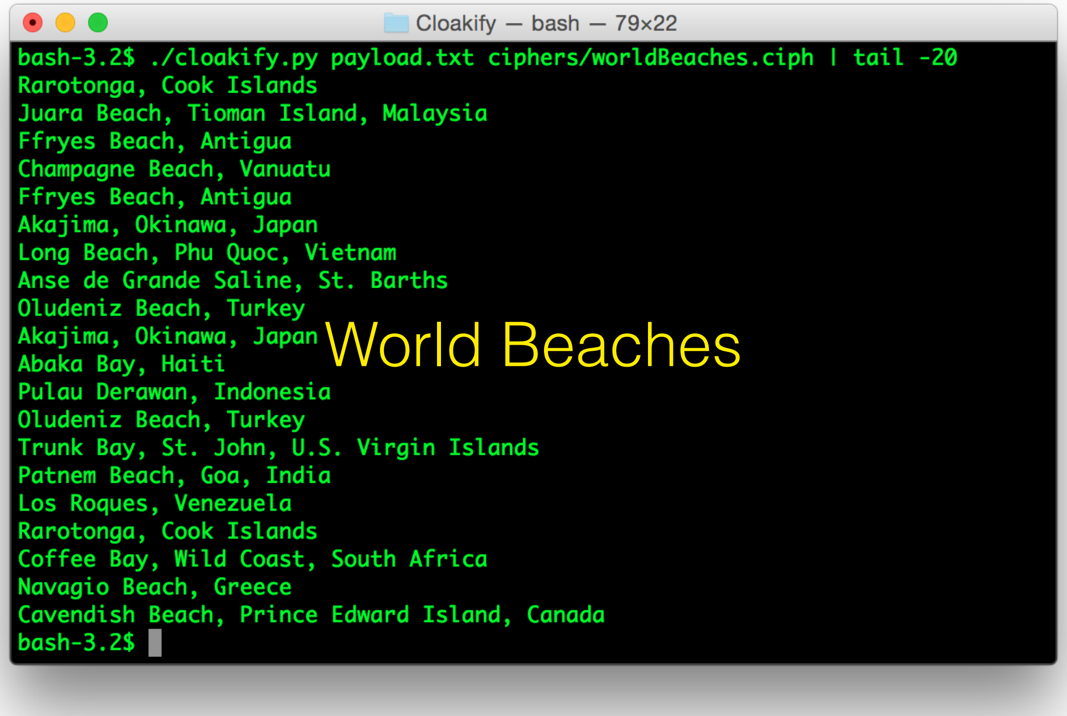
```
bash-3.2$ ./cloakify.py payload.txt ciphers/amphibians.ciph | tail -20
Platycephalus
Brachycephalidae
Myobatrachidae
Craugastoridae
Myobatrachidae
Hynobiidae
Californiense
Croceum
Pipidae
Hynobiidae
Sigillatum
Phrynobatrachidae
Pipidae
Rivularis
Rhacophoridae
Bufonidae
Platycephalus
Elongatus
Torosa
Siphonopidae
bash-3.2$
```



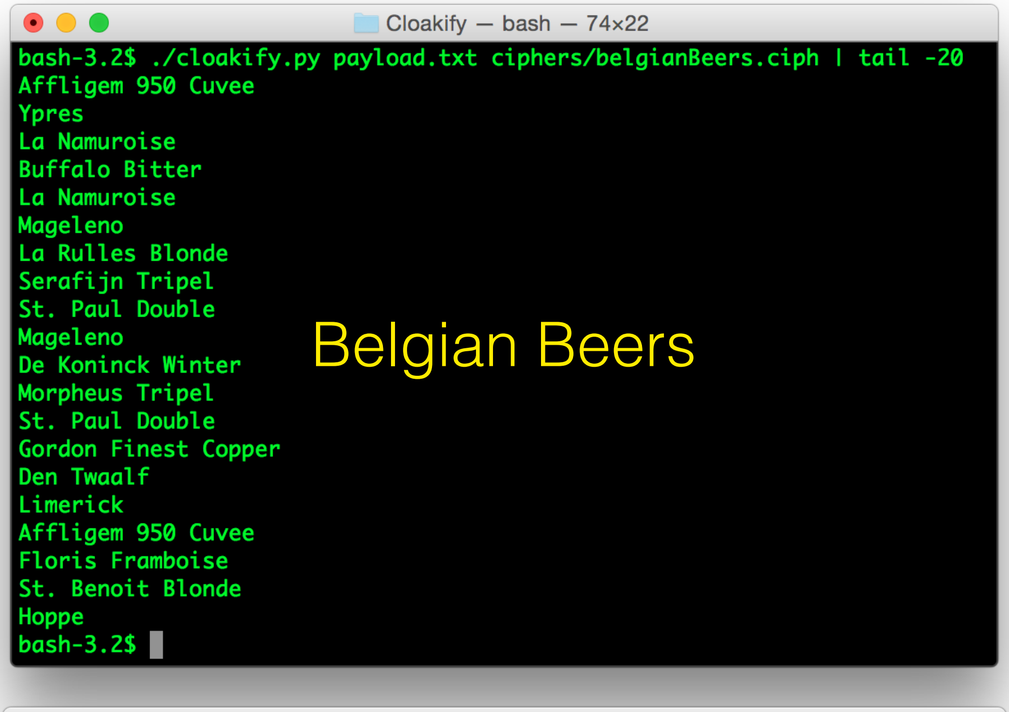
```
bash-3.2$ ./cloakify.py payload.txt ciphers/geocache.ciph | tail -20
Blazing Star State Trail N 43 ° 38.226 W 093 ° 18.541
Minnesota Valley State Recreation Area N 44 ° 39.144 W 093 ° 42.849
Myre-Big Island State Park N 43 ° 38.226 W 093 ° 18.541
Zippel Bay State Park N 48 ° 50.891 W 094 ° 50.859
Myre-Big Island State Park N 43 ° 38.226 W 093 ° 18.541
Frontenac State Park N 44 ° 31.428 W 092 ° 20.467
Grand Portage State Park N 48 ° 00.200 W 089 ° 35.657
Glendalough State Park N 46 ° 19.233 W 095 ° 40.287
Scenic State Park N 47 ° 42.700 W 093 ° 34.167
Frontenac State Park N 44 ° 31.428 W 092 ° 20.467
Great River Bluffs State Park N 46 ° 51.919 W 096 ° 28.031
Schoolcraft State Park N 47 ° 13.390 W 093 ° 48.252
Scenic State Park N 47 ° 42.700 W 093 ° 34.167
Temperance River State Park N 47 ° 33.241 W 090 ° 52.498
Hayes Lake State Park N 48 ° 38.257 W 095 ° 32.739
Kilen Woods State Park N 43 ° 43.858 W 095 ° 04.101
Blazing Star State Trail N 43 ° 38.226 W 093 ° 18.541
Garden Island State Recreation Area N 49 ° 10.537 W 094 ° 50.031
Cuyuna Country State Recreation Area N 46 ° 28.724 W 093 ° 58.598
Banning State Park N 46 ° 09.869 W 092 ° 50.373
bash-3.2$
```

Exfiltrating Data From Secure Networks

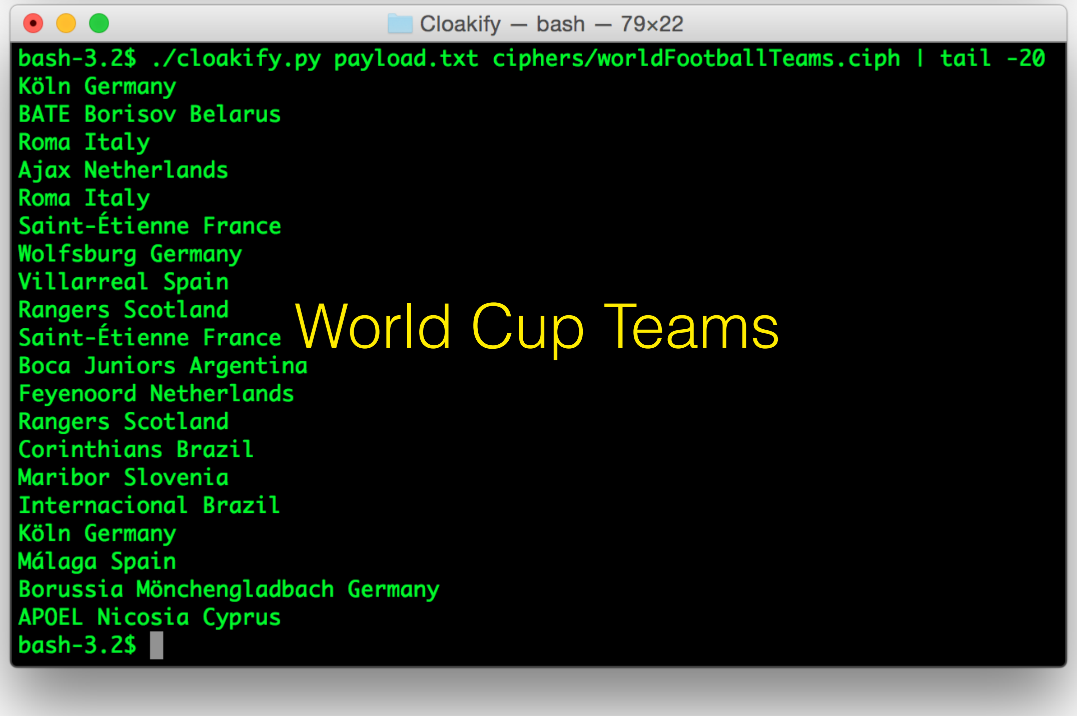
Preinstalled Ciphers Sampler: Collect the whole set, win valuable prizes!



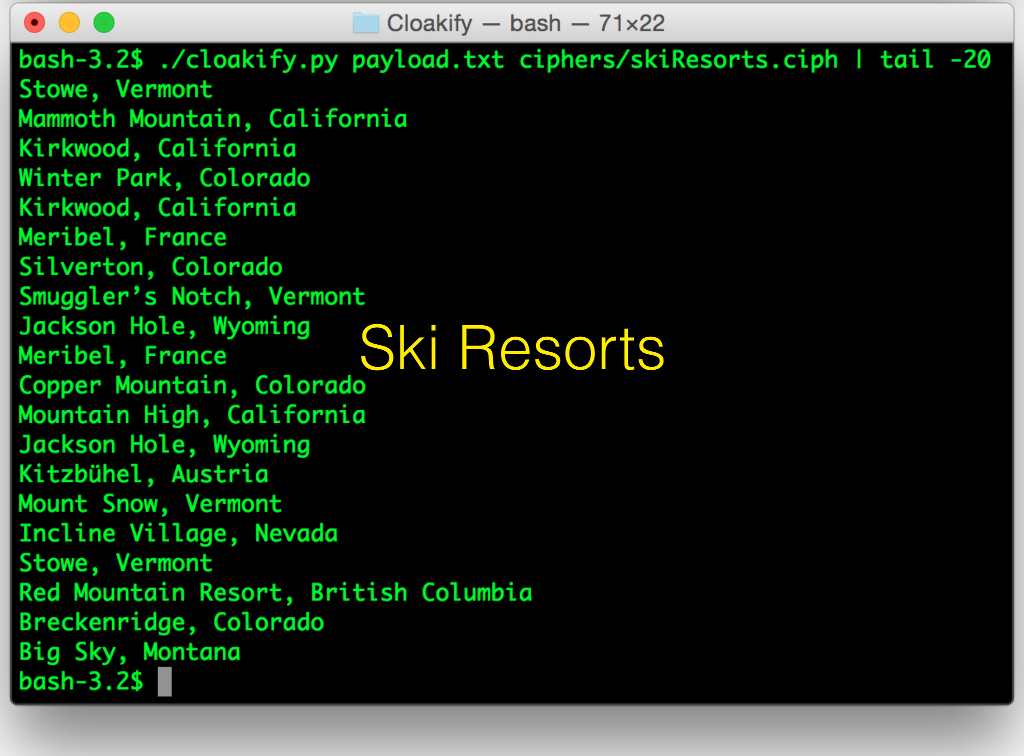
```
bash-3.2$ ./cloakify.py payload.txt ciphers/worldBeaches.ciph | tail -20
Rarotonga, Cook Islands
Juara Beach, Tioman Island, Malaysia
Ffryes Beach, Antigua
Champagne Beach, Vanuatu
Ffryes Beach, Antigua
Akajima, Okinawa, Japan
Long Beach, Phu Quoc, Vietnam
Anse de Grande Saline, St. Barths
Oludeniz Beach, Turkey
Akajima, Okinawa, Japan
Abaka Bay, Haiti
Pulau Derawan, Indonesia
Oludeniz Beach, Turkey
Trunk Bay, St. John, U.S. Virgin Islands
Patnem Beach, Goa, India
Los Roques, Venezuela
Rarotonga, Cook Islands
Coffee Bay, Wild Coast, South Africa
Navagio Beach, Greece
Cavendish Beach, Prince Edward Island, Canada
bash-3.2$
```



```
bash-3.2$ ./cloakify.py payload.txt ciphers/belgianBeers.ciph | tail -20
Affligem 950 Cuvee
Ypres
La Namuroise
Buffalo Bitter
La Namuroise
Mageleno
La Rulles Blonde
Serafijn Tripel
St. Paul Double
Mageleno
De Koninck Winter
Morpheus Tripel
St. Paul Double
Gordon Finest Copper
Den Twaalf
Limerick
Affligem 950 Cuvee
Floris Framboise
St. Benoit Blonde
Hoppe
bash-3.2$
```



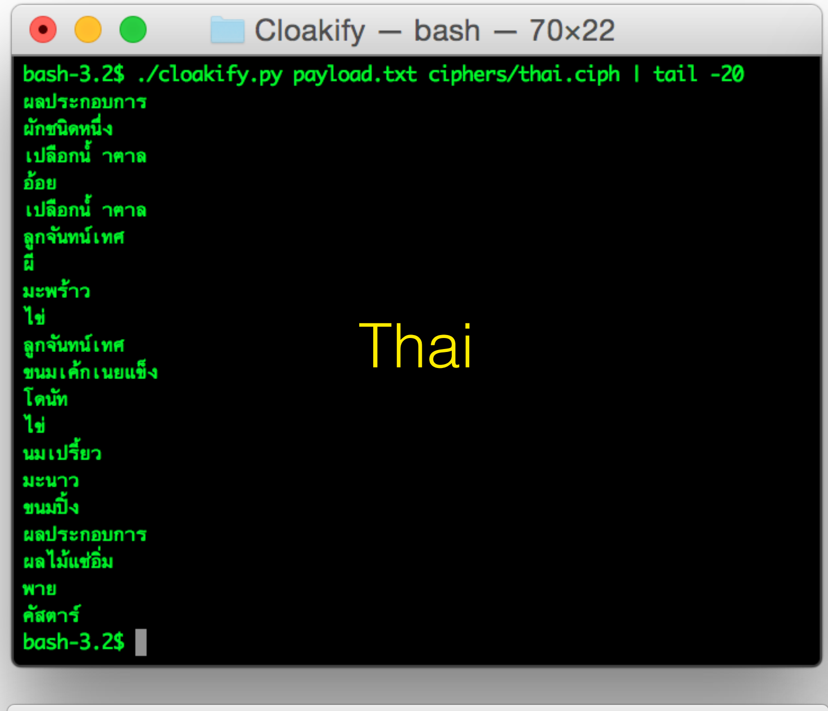
```
bash-3.2$ ./cloakify.py payload.txt ciphers/worldFootballTeams.ciph | tail -20
Köln Germany
BATE Borisov Belarus
Roma Italy
Ajax Netherlands
Roma Italy
Saint-Étienne France
Wolfsburg Germany
Villarreal Spain
Rangers Scotland
Saint-Étienne France
Boca Juniors Argentina
Feyenoord Netherlands
Rangers Scotland
Corinthians Brazil
Maribor Slovenia
Internacional Brazil
Köln Germany
Málaga Spain
Borussia Mönchengladbach Germany
APOEL Nicosia Cyprus
bash-3.2$
```



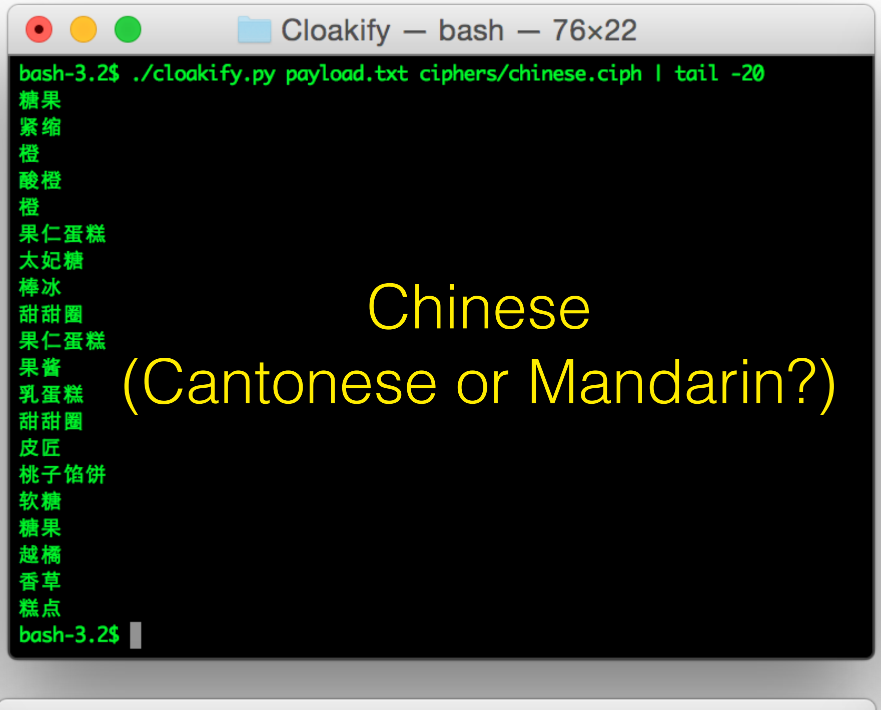
```
bash-3.2$ ./cloakify.py payload.txt ciphers/skiResorts.ciph | tail -20
Stowe, Vermont
Mammoth Mountain, California
Kirkwood, California
Winter Park, Colorado
Kirkwood, California
Meribel, France
Silverton, Colorado
Smuggler's Notch, Vermont
Jackson Hole, Wyoming
Meribel, France
Copper Mountain, Colorado
Mountain High, California
Jackson Hole, Wyoming
Kitzbühel, Austria
Mount Snow, Vermont
Incline Village, Nevada
Stowe, Vermont
Red Mountain Resort, British Columbia
Breckenridge, Colorado
Big Sky, Montana
bash-3.2$
```

Exfiltrating Data From Secure Networks

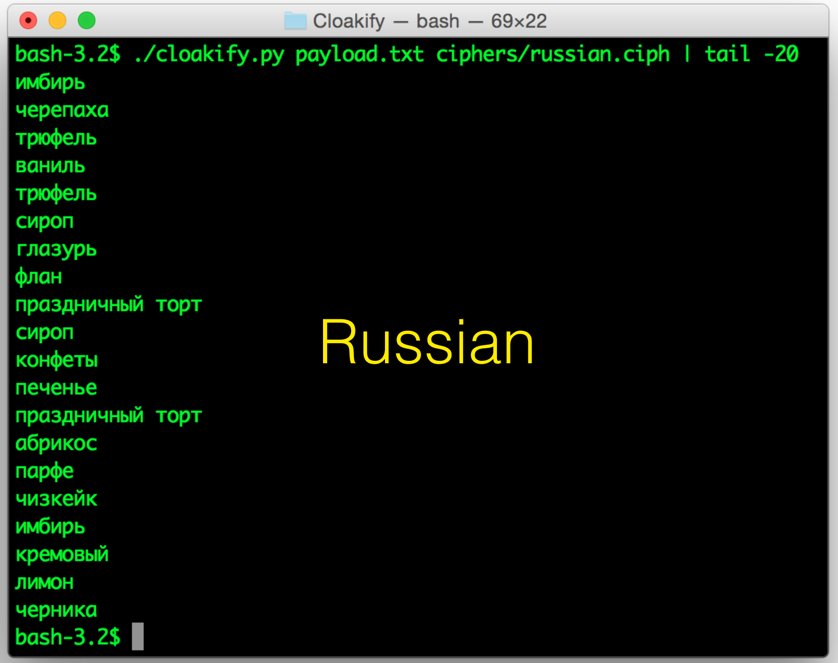
Since not all Machines and Analysts would see English as a friendly mode of communication… Multilingual desserts:



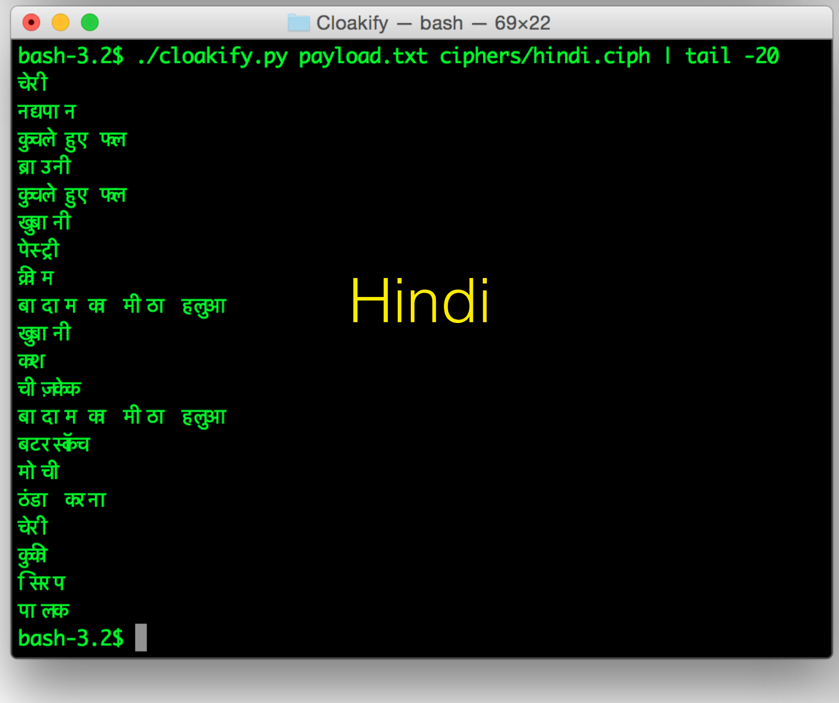
```
bash-3.2$ ./cloakify.py payload.txt ciphers/thai.ciph | tail -20
ผลประกอบการ
ผักชนิดหนึ่ง
เปลือกน้ำตาล
อ้อย
เปลือกน้ำตาล
ลูกจันทน์เทศ
ผี
มะพร้าว
ไข่
ลูกจันทน์เทศ
ขนมเค้กเนยแข็ง
โดนัท
ไข่
นมเปรี้ยว
มะนาว
ขนมปิ้ง
ผลประกอบการ
ผลไม้แช่อิ่ม
พาย
คัสตาร์
bash-3.2$
```



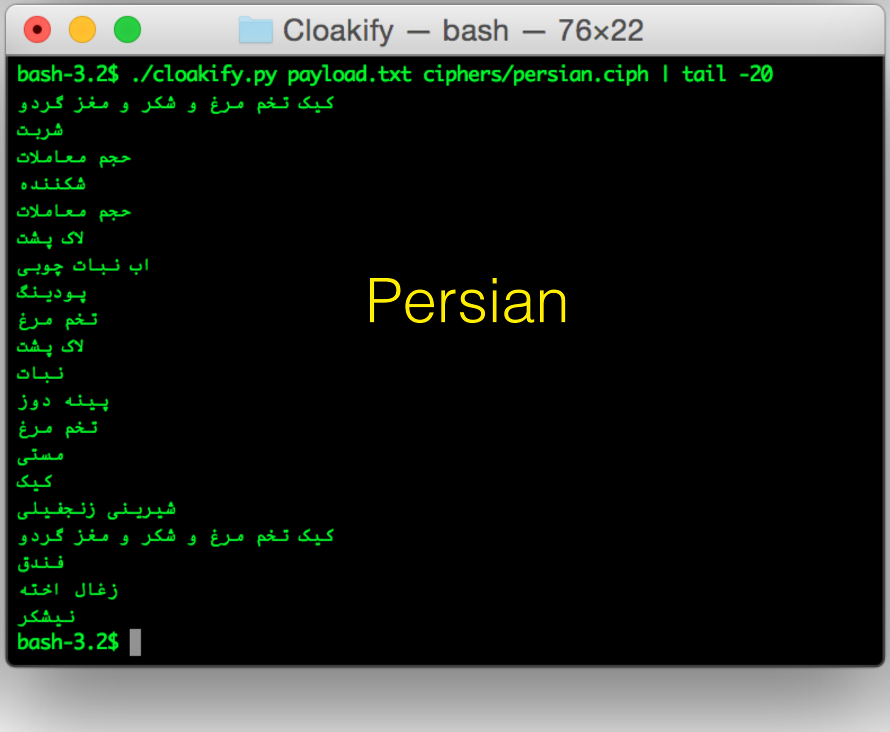
```
bash-3.2$ ./cloakify.py payload.txt ciphers/chinese.ciph | tail -20
糖果
紧缩
橙
酸橙
橙
果仁蛋糕
太妃糖
棒冰
甜甜圈
果仁蛋糕
果酱
乳蛋糕
甜甜圈
皮匠
桃子馅饼
软糖
糖果
越橘
香草
糕点
bash-3.2$
```



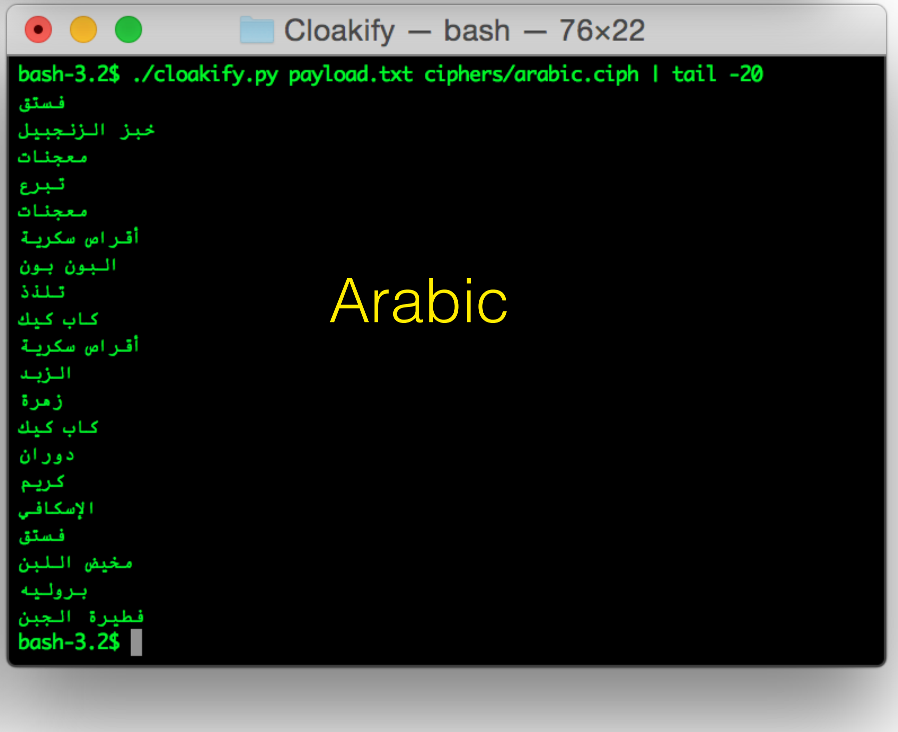
```
bash-3.2$ ./cloakify.py payload.txt ciphers/russian.ciph | tail -20
имбирь
черепаха
трюфель
ваниль
трюфель
сироп
глазурь
флан
праздничный торт
сироп
конфеты
печенье
праздничный торт
абрикос
парфе
чизкейк
имбирь
кремовый
лимон
черника
bash-3.2$
```



```
bash-3.2$ ./cloakify.py payload.txt ciphers/hindi.ciph | tail -20
चेरी
नद्यपान
कुचले हुए फल
ब्राउनी
कुचले हुए फल
खुबानी
पेस्ट्री
क्रीम
बादाम का मीठा हलुआ
खुबानी
क्रश
चीज़केक
बादाम का मीठा हलुआ
बटरस्कॉच
मोची
ठंडा करना
चेरी
कुकी
सिरप
पालक
bash-3.2$
```



```
bash-3.2$ ./cloakify.py payload.txt ciphers/persian.ciph | tail -20
کیک تخم مرغ و شکر و مغز گردو
شربت
حجم معاملات
شکننده
حجم معاملات
لاک پشت
اب نبات چوبی
پودینگ
تخم مرغ
لاک پشت
نبات
پینه دوز
تخم مرغ
مستی
کیک
شیرینی زنجفیلی
کیک تخم مرغ و شکر و مغز گردو
فندق
زغال اخته
نیشکر
bash-3.2$
```



```
bash-3.2$ ./cloakify.py payload.txt ciphers/arabic.ciph | tail -20
فستق
خبز الزنجبيل
معجنات
تبرع
معجنات
أقراص سكرية
البون بون
تلذذ
كاب كيك
أقراص سكرية
الزبد
زهرة
كاب كيك
دوران
كريم
الإسكافي
فستق
مخيض اللبن
بروليه
فطيرة الجبن
bash-3.2$
```

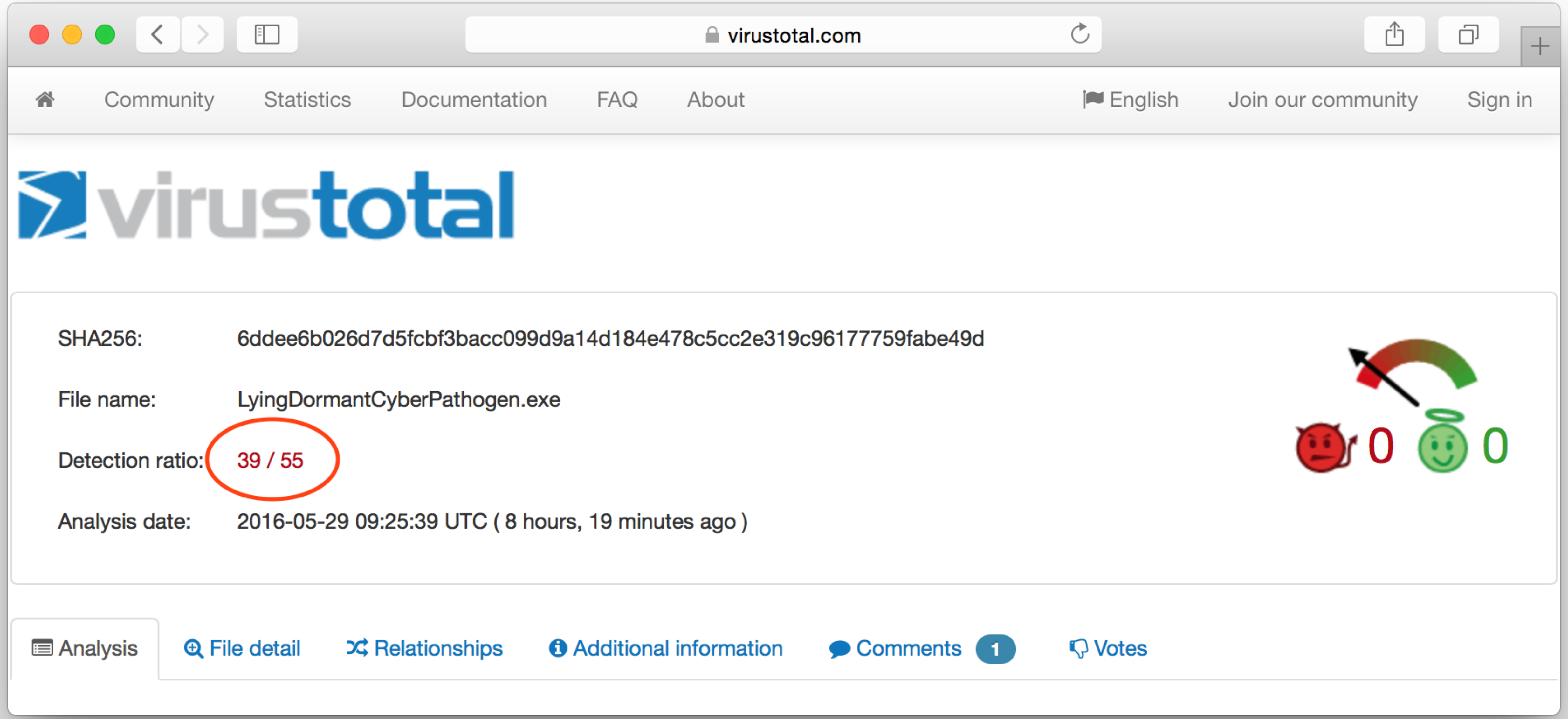
Exfiltrating Data From Secure Networks

Defeat AntiVirus Scanners

Exfiltrating Data From Secure Networks

AV Evasion

Scenario: A Human Resistance agent needs to transfer a stolen weaponized executable back to her unit command, but The Hegemony has tools and human collaborators to detect and block transfer of their synthetic intellectual property



SHA256: 6ddee6b026d7d5fcbf3bacc099d9a14d184e478c5cc2e319c96177759fabe49d

File name: LyingDormantCyberPathogen.exe

Detection ratio: 39 / 55

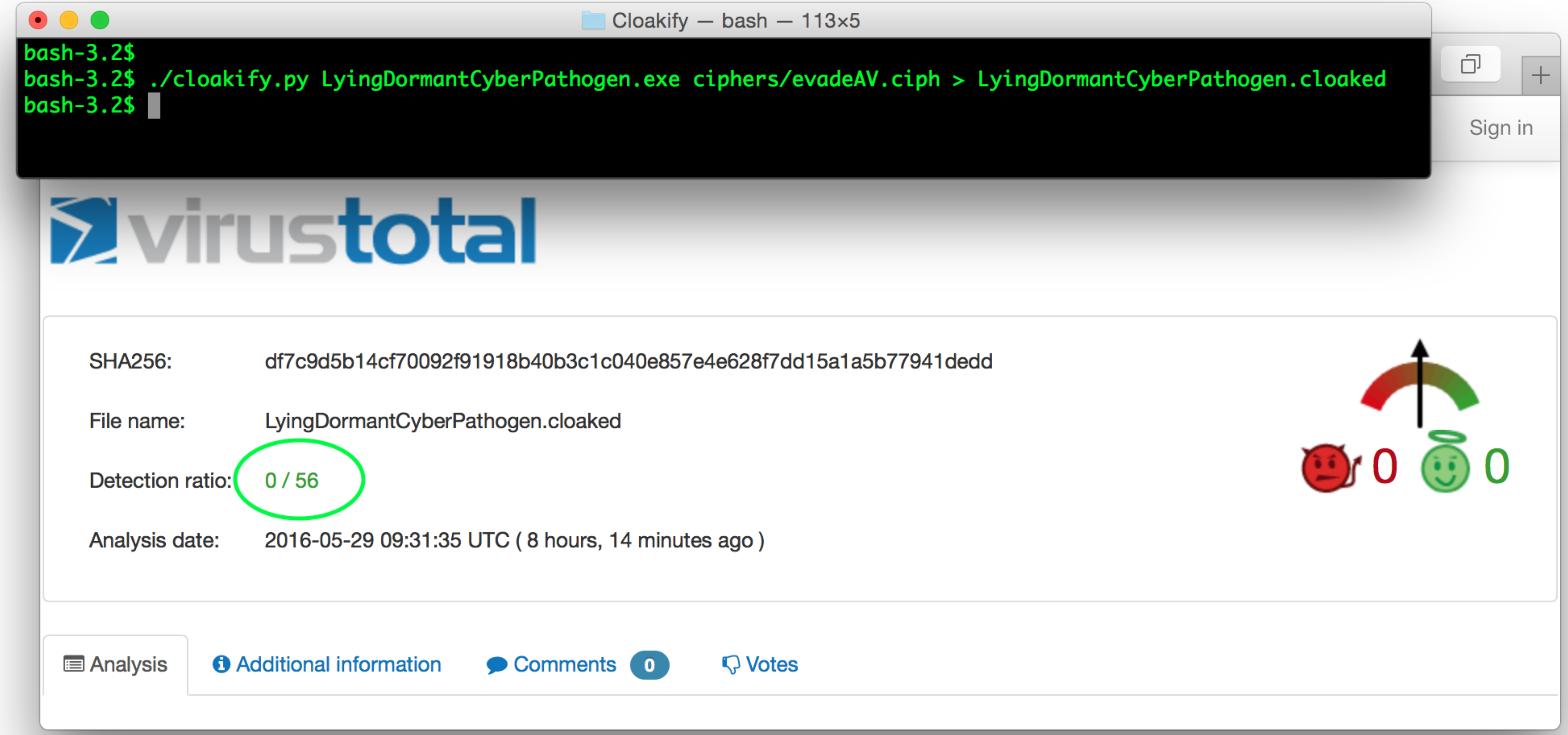
Analysis date: 2016-05-29 09:25:39 UTC (8 hours, 19 minutes ago)

Exfiltrating Data From Secure Networks

AV Evasion (cont.)

Solution: Encode the payload with ‘evadeAV’ cipher

(Any cipher, actually, ‘evadeAV’ just results in the smallest payload)



```
bash-3.2$
bash-3.2$ ./cloakify.py LyingDormantCyberPathogen.exe ciphers/evadeAV.ciph > LyingDormantCyberPathogen.cloaked
bash-3.2$ 
```

SHA256: df7c9d5b14cf70092f91918b40b3c1c040e857e4e628f7dd15a1a5b77941dedd

File name: LyingDormantCyberPathogen.cloaked

Detection ratio: 0 / 56

Analysis date: 2016-05-29 09:31:35 UTC (8 hours, 14 minutes ago)

Exfiltrating Data From Secure Networks

Social Engineering the Analyst

Exfiltrating Data From Secure Networks

Why Lists?

We’re used to seeing blocks of encoded text and identifying them as payloads

Once spotted, our instinct and training (both Red and Blue teams) is to attack them

Lists of items evoke a different cognitive response than do blocks of data

- Triggers memories, patterns, biases
- Mental noise obscures perception of underlying data that is hidden in the order of the items themselves

Careful selection of list elements opens a range of social engineering possibilities as an added layer to the exfiltration

Exfiltrating Data From Secure Networks

Social Engineering Vectors Via Text-Based Steganography

Analyst Type: Sgt. Checkbox

Mindset: “End shift, find beer”

Ciphers: Irrelevant Data - “These Aren’t The Droids You’re Looking For”

Sgt. Checkbox and civilian cousins generally go through the motions, following only high level procedures. Begins working a ticket and casually examines the data traffic, looking for the first plausible reason to close the ticket and quickly move on.

Solution: Give them that plausible reason. A list of desserts is boring and harmless.

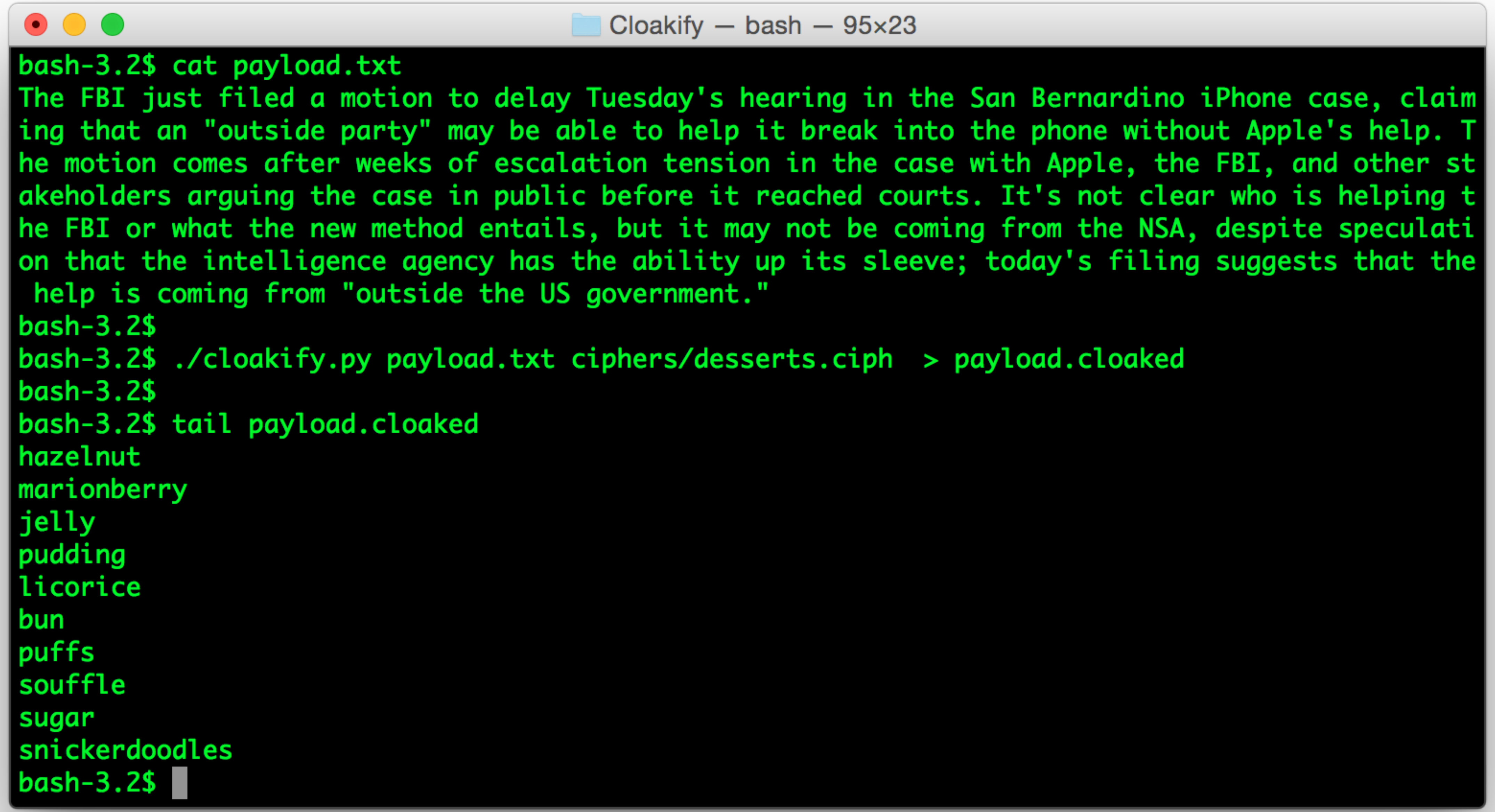
“Ticket closed! Beer now closer! Good job, me!”

Exfiltrating Data From Secure Networks

Irrelevant Data - “These Aren’t The Droids You’re Looking For”

Scenario: You need to exfiltrate data through Sgt. Checkbox (or a blacklist DLP filter).

Solution: Encode your payload with any of the benign ciphers, exfiltrate

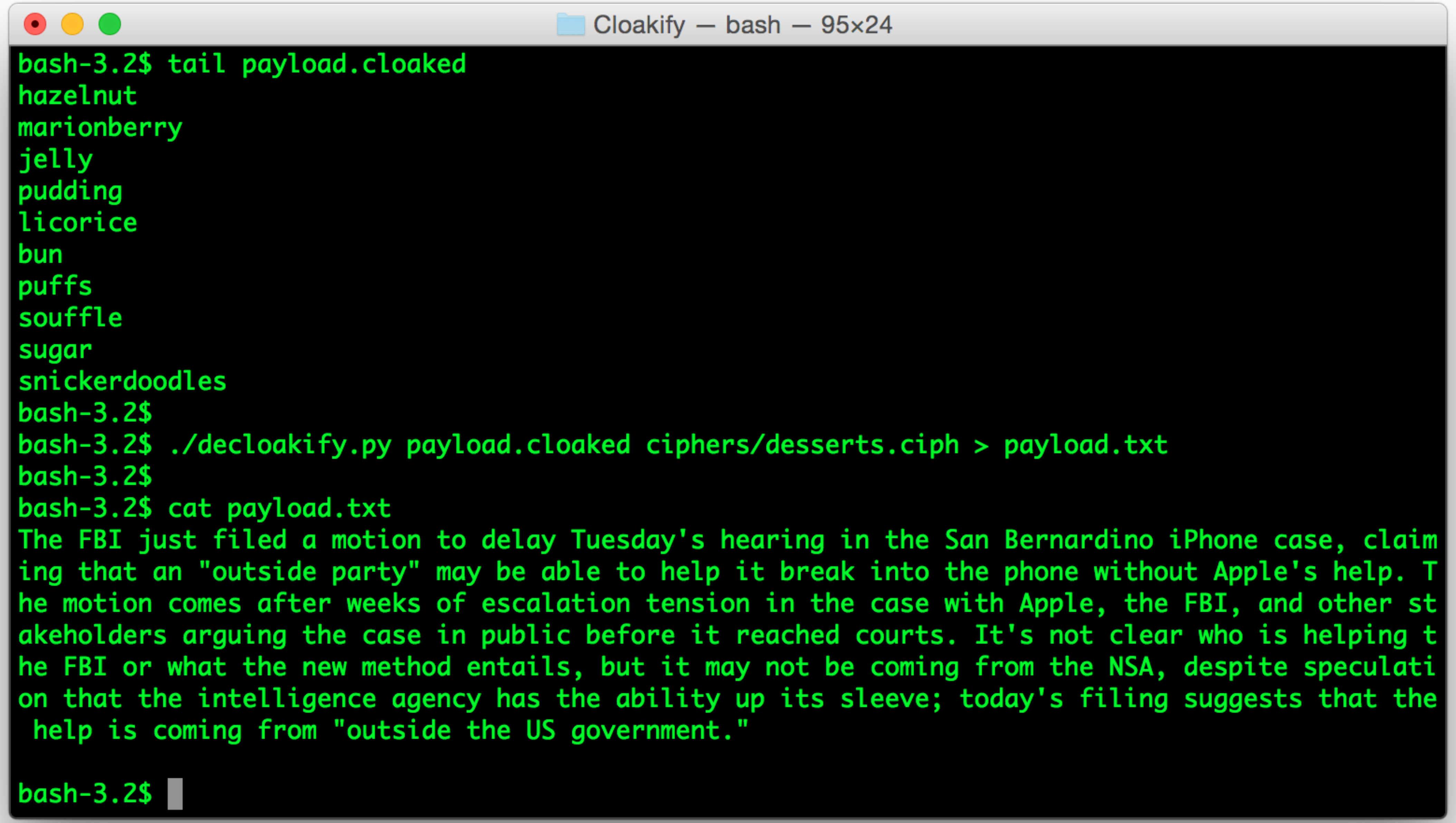


```
bash-3.2$ cat payload.txt
The FBI just filed a motion to delay Tuesday's hearing in the San Bernardino iPhone case, claim
ing that an "outside party" may be able to help it break into the phone without Apple's help. T
he motion comes after weeks of escalation tension in the case with Apple, the FBI, and other st
akeholders arguing the case in public before it reached courts. It's not clear who is helping t
he FBI or what the new method entails, but it may not be coming from the NSA, despite speculati
on that the intelligence agency has the ability up its sleeve; today's filing suggests that the
 help is coming from "outside the US government."
bash-3.2$
bash-3.2$ ./cloakify.py payload.txt ciphers/desserts.ciph  > payload.cloaked
bash-3.2$
bash-3.2$ tail payload.cloaked
hazelnut
marionberry
jelly
pudding
licorice
bun
puffs
souffle
sugar
snickerdoodles
bash-3.2$ 
```

Exfiltrating Data From Secure Networks

Irrelevant Data - "These Aren't The Droids You're Looking For" cont.

"The Snickerdoodles are a lie? Nooooooo!" (But the payload is still delicious.)



```
bash-3.2$ tail payload.cloaked
hazelnut
marionberry
jelly
pudding
licorice
bun
puffs
souffle
sugar
snickerdoodles
bash-3.2$
bash-3.2$ ./decloakify.py payload.cloaked ciphers/desserts.ciph > payload.txt
bash-3.2$
bash-3.2$ cat payload.txt
The FBI just filed a motion to delay Tuesday's hearing in the San Bernardino iPhone case, claim
ing that an "outside party" may be able to help it break into the phone without Apple's help. T
he motion comes after weeks of escalation tension in the case with Apple, the FBI, and other st
akeholders arguing the case in public before it reached courts. It's not clear who is helping t
he FBI or what the new method entails, but it may not be coming from the NSA, despite speculati
on that the intelligence agency has the ability up its sleeve; today's filing suggests that the
 help is coming from "outside the US government."

bash-3.2$ 
```

Exfiltrating Data From Secure Networks

Social Engineering Vectors Via Text-Based Steganography

Analyst Type: Gandalf

Mindset: “NONE SHALL PASS!”

Ciphers: Whitelisted Data - “There Is No Spoon”

The Gandalf analyst is a hardcore goal keeper. They were probably a hall monitor in grade school. And really liked it.

They will doggedly review data transfers to validate that only approved data is leaving the secure network.

Solution: Determine what data is allowed to exit the network, then transform the payload into that whitelisted dataset. Analyst sees only approved data content.

“All is in order! Good job, me! Time to iron socks.”

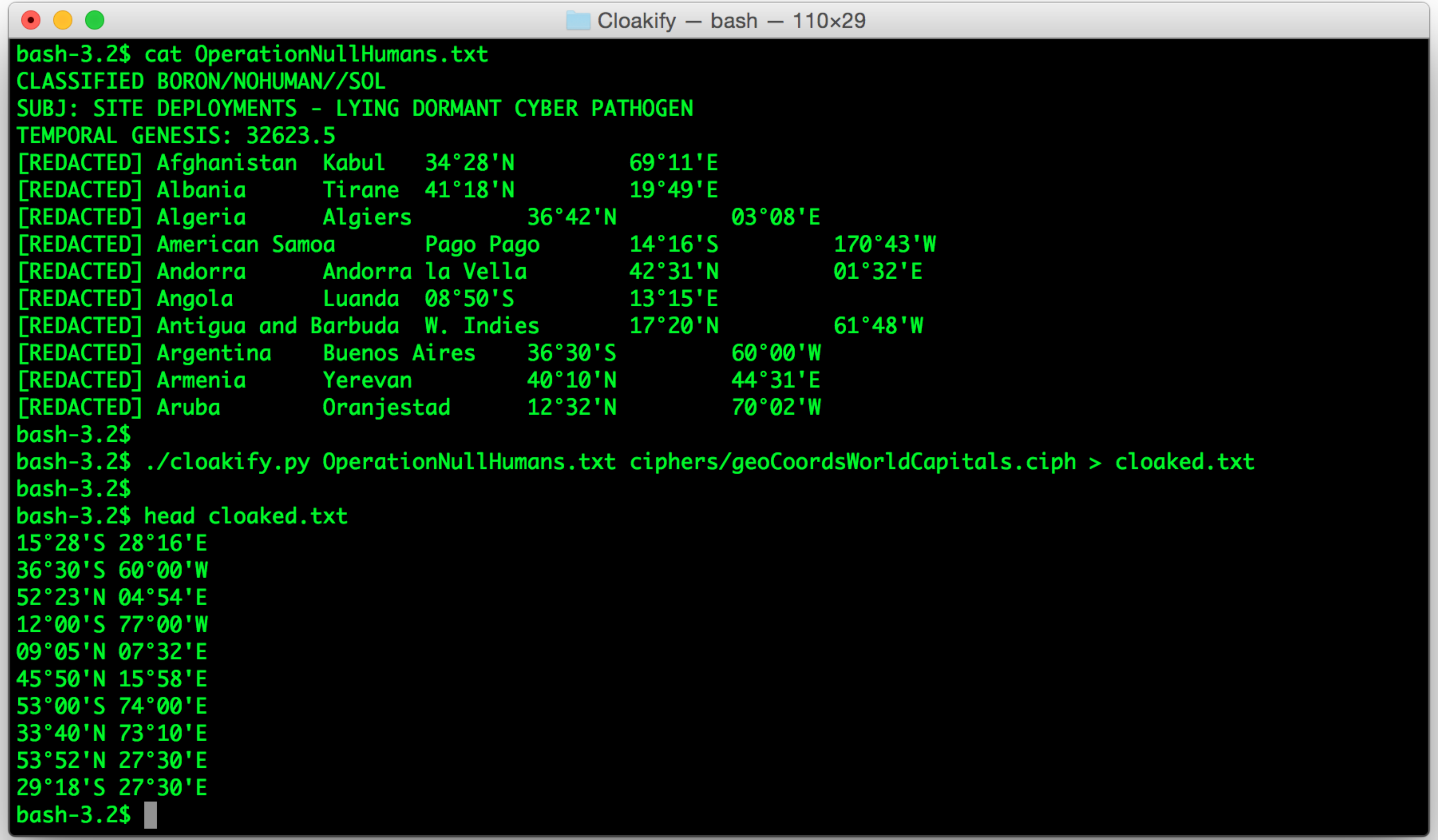
Exfiltrating Data From Secure Networks

Defeating the Machine Hegemony While Socially Engineering Their Human Collaborators

Defeating the Gandalf analyst and Whitelisted Data Controls - “There Is No Spoon”

Scenario: Our agent has infiltrated the Machine Hegemony’s Global C&C Center and uncovered critical targeting plans for pending operations. An MLS device is stripping all data but GeoCoords from outgoing dataflows to regional commands, which is a shame because there is some great information that could probably save the battered Human Resistance.

Solution: Encode the payload with geocoords cipher to give the MLS exactly what it wants to see, exfiltrate to our data assembly point.



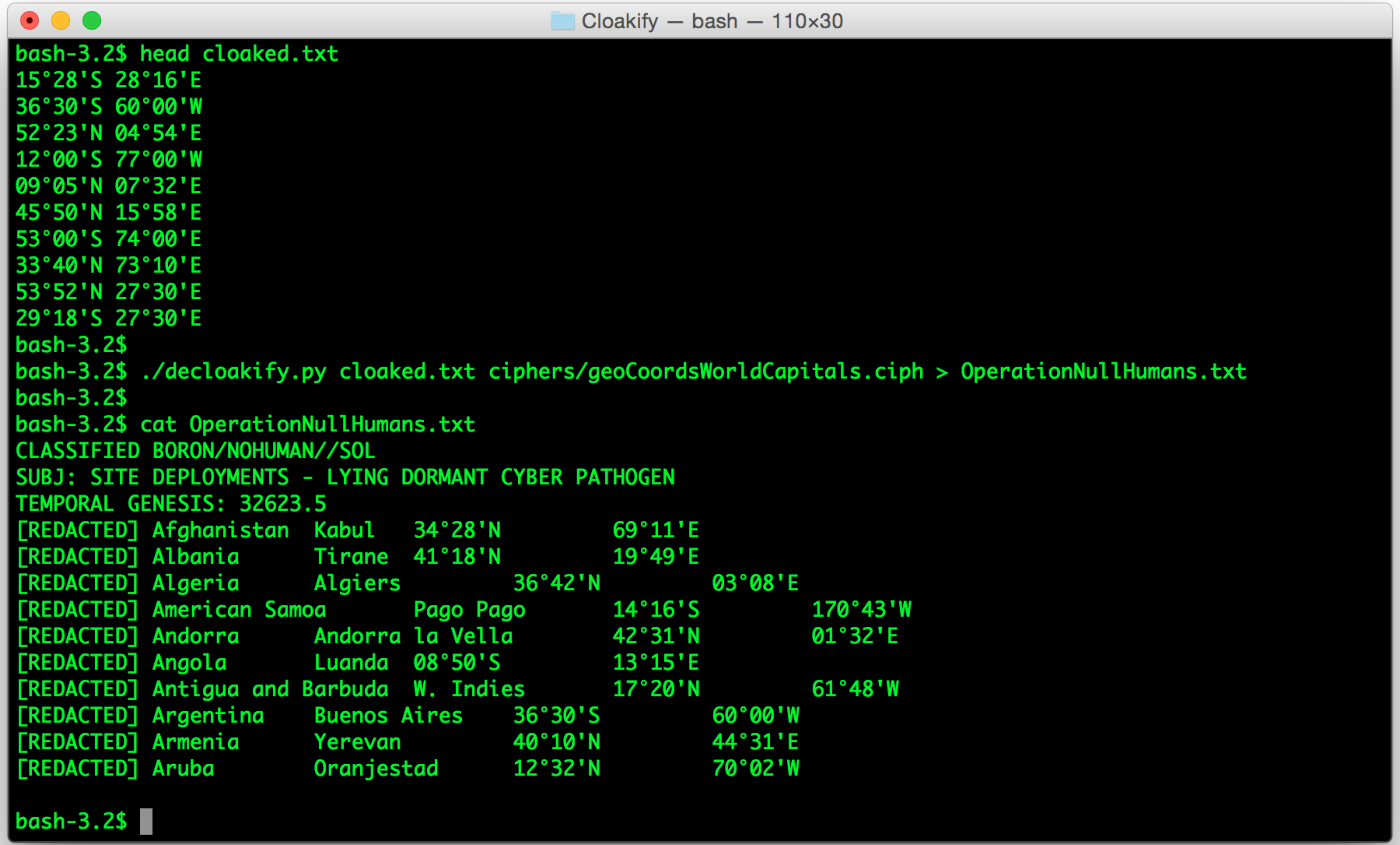
```
bash-3.2$ cat OperationNullHumans.txt
CLASSIFIED BORON/NOHUMAN//SOL
SUBJ: SITE DEPLOYMENTS - LYING DORMANT CYBER PATHOGEN
TEMPORAL GENESIS: 32623.5
[REDACTED] Afghanistan  Kabul   34°28'N         69°11'E
[REDACTED] Albania      Tirane  41°18'N         19°49'E
[REDACTED] Algeria      Algiers         36°42'N         03°08'E
[REDACTED] American Samoa       Pago Pago       14°16'S         170°43'W
[REDACTED] Andorra      Andorra la Vella        42°31'N         01°32'E
[REDACTED] Angola       Luanda  08°50'S         13°15'E
[REDACTED] Antigua and Barbuda  W. Indies       17°20'N         61°48'W
[REDACTED] Argentina    Buenos Aires    36°30'S         60°00'W
[REDACTED] Armenia      Yerevan         40°10'N         44°31'E
[REDACTED] Aruba        Oranjestad      12°32'N         70°02'W
bash-3.2$
bash-3.2$ ./cloakify.py OperationNullHumans.txt ciphers/geoCoordsWorldCapitals.ciph > cloaked.txt
bash-3.2$
bash-3.2$ head cloaked.txt
15°28'S 28°16'E
36°30'S 60°00'W
52°23'N 04°54'E
12°00'S 77°00'W
09°05'N 07°32'E
45°50'N 15°58'E
53°00'S 74°00'E
33°40'N 73°10'E
53°52'N 27°30'E
29°18'S 27°30'E
bash-3.2$ 
```

Exfiltrating Data From Secure Networks

Defeating the Machine Hegemony While Socially Engineering Their Human Collaborators

Defeating the Gandalf analyst and Whitelisted Data Controls - “There Is No Spoon” (cont.)

Recover data on the other side, decode. “Such Exfiltrate!” +3 to the Human Resistance. Prepare to launch counterops.



```
bash-3.2$ head cloaked.txt
15°28'S 28°16'E
36°30'S 60°00'W
52°23'N 04°54'E
12°00'S 77°00'W
09°05'N 07°32'E
45°50'N 15°58'E
53°00'S 74°00'E
33°40'N 73°10'E
53°52'N 27°30'E
29°18'S 27°30'E
bash-3.2$
bash-3.2$ ./decloakify.py cloaked.txt ciphers/geoCoordsWorldCapitals.ciph > OperationNullHumans.txt
bash-3.2$
bash-3.2$ cat OperationNullHumans.txt
CLASSIFIED BORON/NOHUMAN//SOL
SUBJ: SITE DEPLOYMENTS - LYING DORMANT CYBER PATHOGEN
TEMPORAL GENESIS: 32623.5
[REDACTED] Afghanistan  Kabul   34°28'N         69°11'E
[REDACTED] Albania      Tirane  41°18'N         19°49'E
[REDACTED] Algeria      Algiers         36°42'N         03°08'E
[REDACTED] American Samoa       Pago Pago       14°16'S         170°43'W
[REDACTED] Andorra      Andorra la Vella        42°31'N         01°32'E
[REDACTED] Angola       Luanda  08°50'S         13°15'E
[REDACTED] Antigua and Barbuda  W. Indies       17°20'N         61°48'W
[REDACTED] Argentina    Buenos Aires    36°30'S         60°00'W
[REDACTED] Armenia      Yerevan         40°10'N         44°31'E
[REDACTED] Aruba        Oranjestad      12°32'N         70°02'W

bash-3.2$
```

Exfiltrating Data From Secure Networks

Defeating the Machine Hegemony While Socially Engineering Their Human Collaborators

(Actual SOC Analyst)

Exfiltrating Data From Secure Networks

Social Engineering Vectors Via Text-Based Steganography

Analyst: The Terminator

Mindset: “It can't be reasoned with. It doesn't feel pity, or remorse, or fear.”

Ciphers: Decoy / Inception - “An idea is like a virus. Resilient. Highly contagious.”

The Terminator is a ruthless hunter. They rip through packets like a Sharknado in a kitten sanctuary at nap time. When off shift they work crypto puzzles for fun, and often spoil movies by loudly predicting the next plot twist. Fear these people.

Solution: Transform data into a format that presents a decoy problem, luring the analyst into solving a problem that yields critical (fake) evidence. This drives the investigation down a left turn along a lonely dirt road, with the likelihood that the erroneously critical evidence will then be cataloged and stored for the future report, but otherwise not receive further attention. The embedded exfiltrated data remains undiscovered.

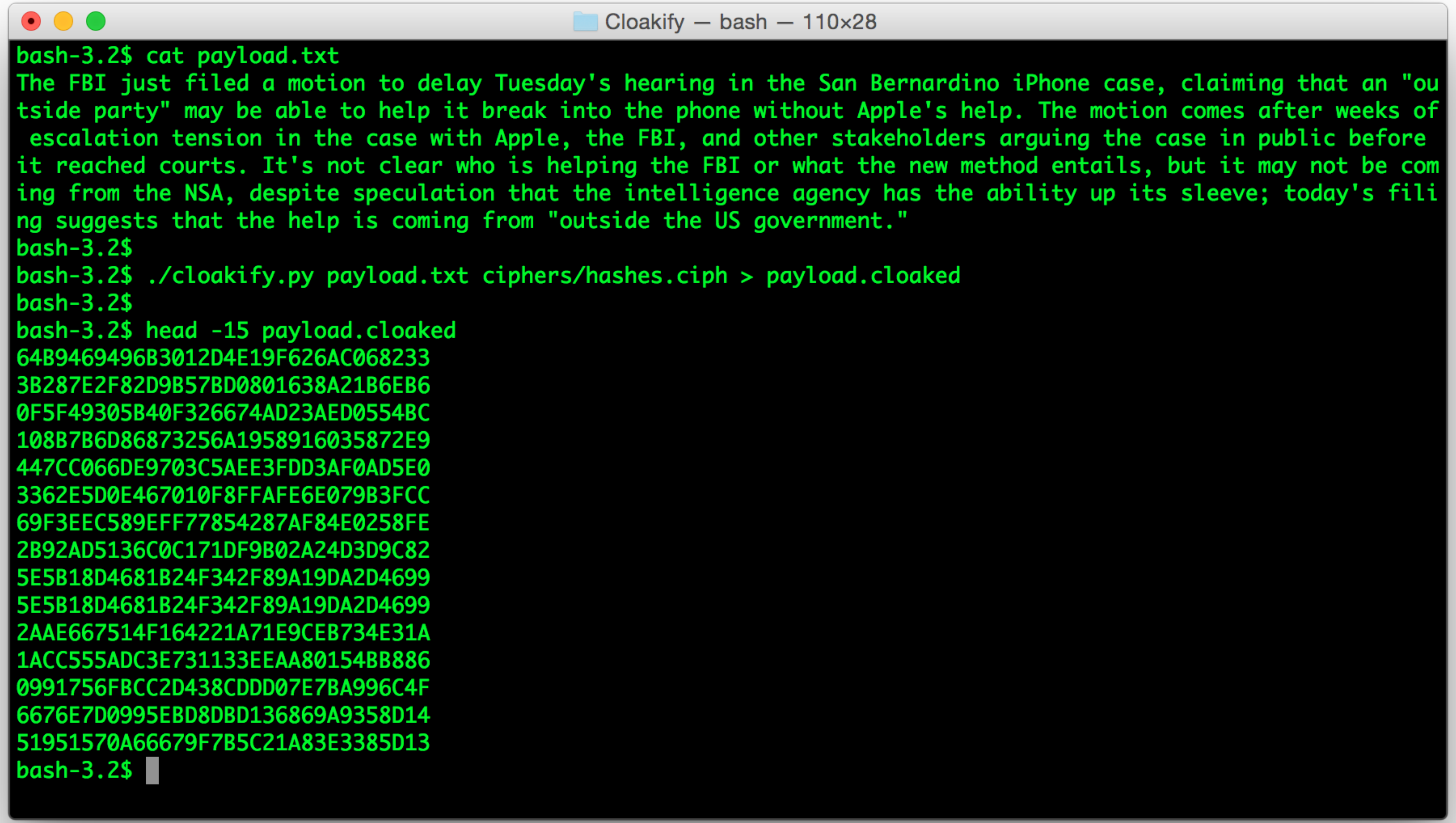
“Phase plasma rifle in the 40 watt range.”

Exfiltrating Data From Secure Networks

Decoy Data - Defeating the Terminator analyst - “An idea is like a virus.”

Scenario: The Hegemony has augmented their CERT & SOC with cadres of Terminator analysts

Solution: Use ‘hashes.ciph’ to encode your payload with hashed passwords, exfiltrate



```
bash-3.2$ cat payload.txt
The FBI just filed a motion to delay Tuesday's hearing in the San Bernardino iPhone case, claiming that an "ou
tside party" may be able to help it break into the phone without Apple's help. The motion comes after weeks of
 escalation tension in the case with Apple, the FBI, and other stakeholders arguing the case in public before
it reached courts. It's not clear who is helping the FBI or what the new method entails, but it may not be com
ing from the NSA, despite speculation that the intelligence agency has the ability up its sleeve; today's fili
ng suggests that the help is coming from "outside the US government."
bash-3.2$
bash-3.2$ ./cloakify.py payload.txt ciphers/hashes.ciph > payload.cloaked
bash-3.2$
bash-3.2$ head -15 payload.cloaked
64B9469496B3012D4E19F626AC068233
3B287E2F82D9B57BD0801638A21B6EB6
0F5F49305B40F326674AD23AED0554BC
108B7B6D86873256A1958916035872E9
447CC066DE9703C5AEE3FDD3AF0AD5E0
3362E5D0E467010F8FFAFE6E079B3FCC
69F3EEC589EFF77854287AF84E0258FE
2B92AD5136C0C171DF9B02A24D3D9C82
5E5B18D4681B24F342F89A19DA2D4699
5E5B18D4681B24F342F89A19DA2D4699
2AAE667514F164221A71E9CEB734E31A
1ACC555ADC3E731133EEAA80154BB886
0991756FBCC2D438CDDD07E7BA996C4F
6676E7D0995EBD8DBD136869A9358D14
51951570A66679F7B5C21A83E3385D13
bash-3.2$
```

Exfiltrating Data From Secure Networks

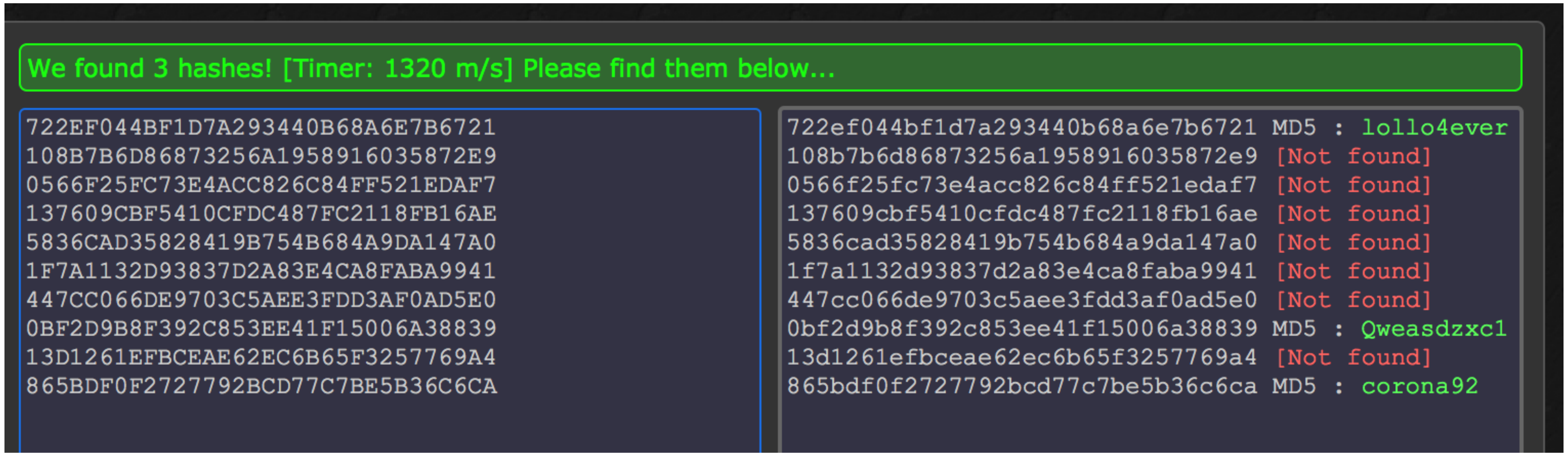
Decoy Data / Inception - “An idea is like a virus.” (cont.)

Ensure that some of the hashes can be cracked easily, adds sense of urgency

Bonus Social Engineering Points if hashed passwords conform to facilities’ password policies

When detected, the incident response team’s focus gets distracted and resources diverted

- Analysts try to identify hashes to determine origin
- Once some hashes are cracked, the team has taken the bait



```
We found 3 hashes! [Timer: 1320 m/s] Please find them below...
```

```
722EF044BF1D7A293440B68A6E7B6721
108B7B6D86873256A1958916035872E9
0566F25FC73E4ACC826C84FF521EDAF7
137609CBF5410CFDC487FC2118FB16AE
5836CAD35828419B754B684A9DA147A0
1F7A1132D93837D2A83E4CA8FABA9941
447CC066DE9703C5AEE3FDD3AF0AD5E0
0BF2D9B8F392C853EE41F15006A38839
13D1261EFBCEAE62EC6B65F3257769A4
865BDF0F2727792BCD77C7BE5B36C6CA
```

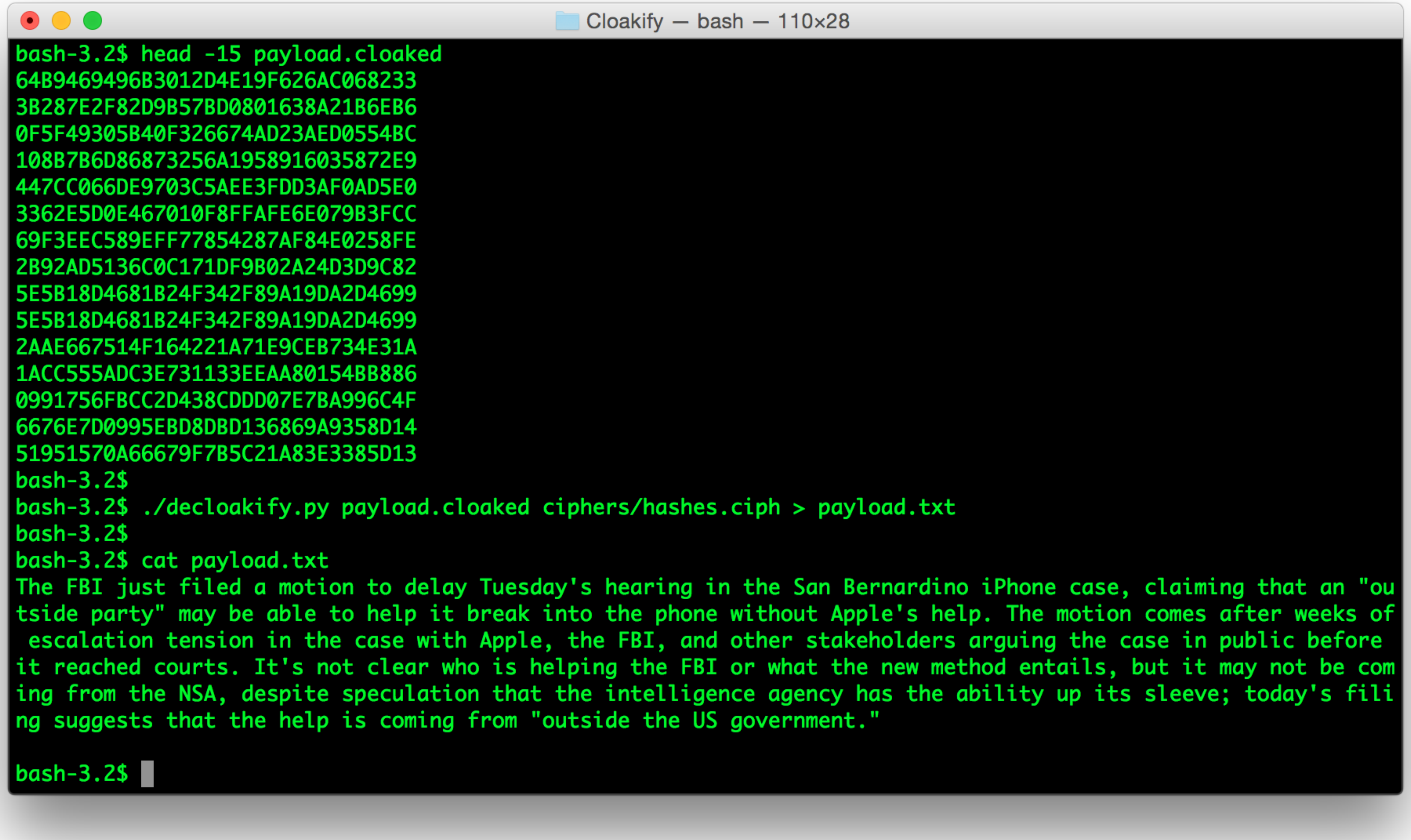
```
722ef044bf1d7a293440b68a6e7b6721 MD5 : lollo4ever
108b7b6d86873256a1958916035872e9 [Not found]
0566f25fc73e4acc826c84ff521edaf7 [Not found]
137609cbf5410cfdc487fc2118fb16ae [Not found]
5836cad35828419b754b684a9da147a0 [Not found]
1f7a1132d93837d2a83e4ca8faba9941 [Not found]
447cc066de9703c5aee3fdd3af0ad5e0 [Not found]
0bf2d9b8f392c853ee41f15006a38839 MD5 : Qweasdzxc1
13d1261efbceae62ec6b65f3257769a4 [Not found]
865bdf0f2727792bcd77c7be5b36c6ca MD5 : corona92
```

With attention now focused on the exfiltrated “password hashes”, the data is likely to end up stored away for record keeping and final incident report, spared from further examination

Meanwhile…

Exfiltrating Data From Secure Networks

Decoy Data / Inception - “An idea is like a virus.” (cont.)



```
bash-3.2$ head -15 payload.cloaked
64B9469496B3012D4E19F626AC068233
3B287E2F82D9B57BD0801638A21B6EB6
0F5F49305B40F326674AD23AED0554BC
108B7B6D86873256A195891603587 2E9
447CC066DE9703C5AEE3FDD3AF0AD5E0
3362E5D0E467010F8FFAFE6E079B3FCC
69F3EEC589EFF77854287AF84E0258FE
2B92AD5136C0C171DF9B02A24D3D9C82
5E5B18D4681B24F342F89A19DA2D4699
5E5B18D4681B24F342F89A19DA2D4699
2AAE667514F164221A71E9CEB734E31A
1ACC555ADC3E731133EEAA80154BB886
0991756FBCC2D438CDDD07E7BA996C4F
6676E7D0995EBD8DBD136869A9358D14
51951570A66679F7B5C21A83E3385D13
bash-3.2$
bash-3.2$ ./decloakify.py payload.cloaked ciphers/hashes.ciph > payload.txt
bash-3.2$
bash-3.2$ cat payload.txt
The FBI just filed a motion to delay Tuesday's hearing in the San Bernardino iPhone case, claiming that an "ou
tside party" may be able to help it break into the phone without Apple's help. The motion comes after weeks of
 escalation tension in the case with Apple, the FBI, and other stakeholders arguing the case in public before
it reached courts. It's not clear who is helping the FBI or what the new method entails, but it may not be com
ing from the NSA, despite speculation that the intelligence agency has the ability up its sleeve; today's fili
ng suggests that the help is coming from "outside the US government."

bash-3.2$ 
```

Exfiltrating Data From Secure Networks

Chaining Ciphers - “Inception All The Way Down”

Takes decoy data to the next level, literally

First apply a decoy cipher (MD5 hashes)

Next apply the second cipher to provide top cover (e.g. IPAddresses)

Exfiltrate!

If the analysts discover and crack that first layer of deception, they uncover the password hashes, triggering the Password Hashes inception from previous example

The only limit is your imagination and creativity…

Exfiltrating Data From Secure Networks

It’s exfiltration all the way down

Exfiltrating Data From Secure Networks

Bonus Social Engineering Round

Add text to the beginning of your cloaked file to further reinforce the dirt road you wish to send them down, closing that SOC ticket

A little extra context goes a long way toward desired cognitive distraction

Emoji Cipher: “Emo Kylo Ren Has So Many Feels About Today”

IP Addresses: “SEO Traffic for Top 100 Sites”

Belgian Beers: “Beer Log - Good Times w/ my two besties Hops & Barley”

Geocaching Sites: “One day I will be a Level 80 Paladin! List of quests!”

Star Trek Characters: “Baby Names To Annoy My In-Laws”

Swedish Chef (Muppets): “Swedish Speed Metal Chef is BEST CHEF!”

Exfiltrating Data From Secure Networks

Bonus Social Engineering Round (cont.)

Consider all cultural factors (nationality, regional preferences, organization’s line of business, individual’s specific interests)

- Via OSINT and any internal material you’ve gained access to

Tie in to local seasons / holidays / events

Generate new ciphers to match

- First lines of annoying Christmas songs
- Amazon Review Titles - Denon AKDL1 Cable
- Venomous Animals Found In My Bathroom - Australia
    - (At least 66 unique entries, I’m sure of it)
- Wild Animal Or Angry Politician’s Hairpiece?

Exfiltrating Data From Secure Networks

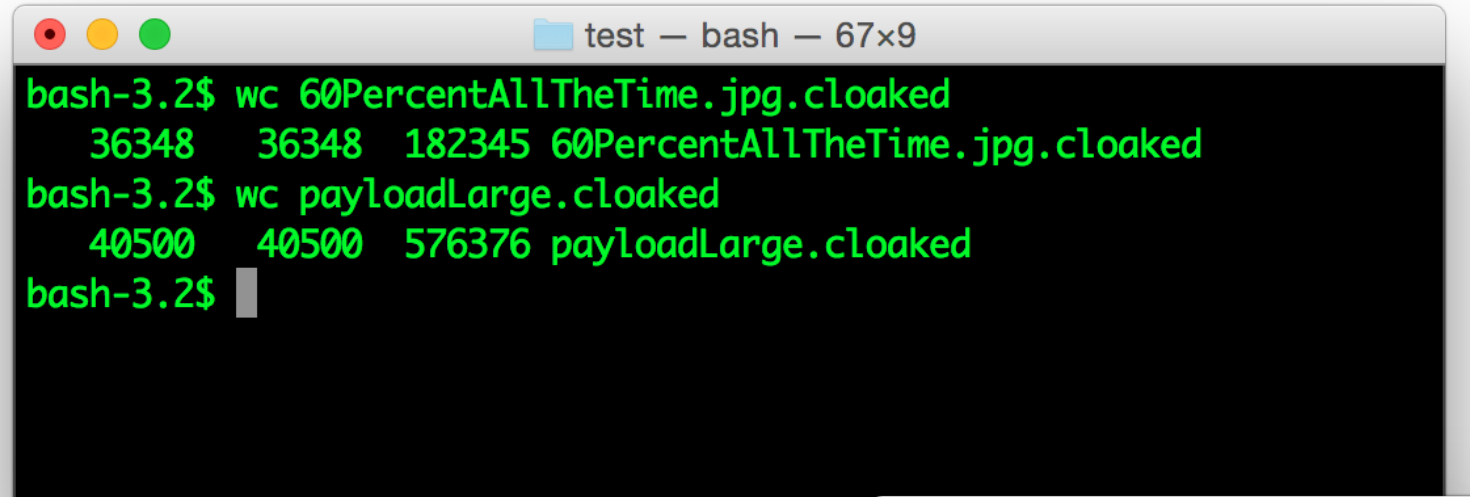
Frequency Analysis Vulnerability

Exfiltrating Data From Secure Networks

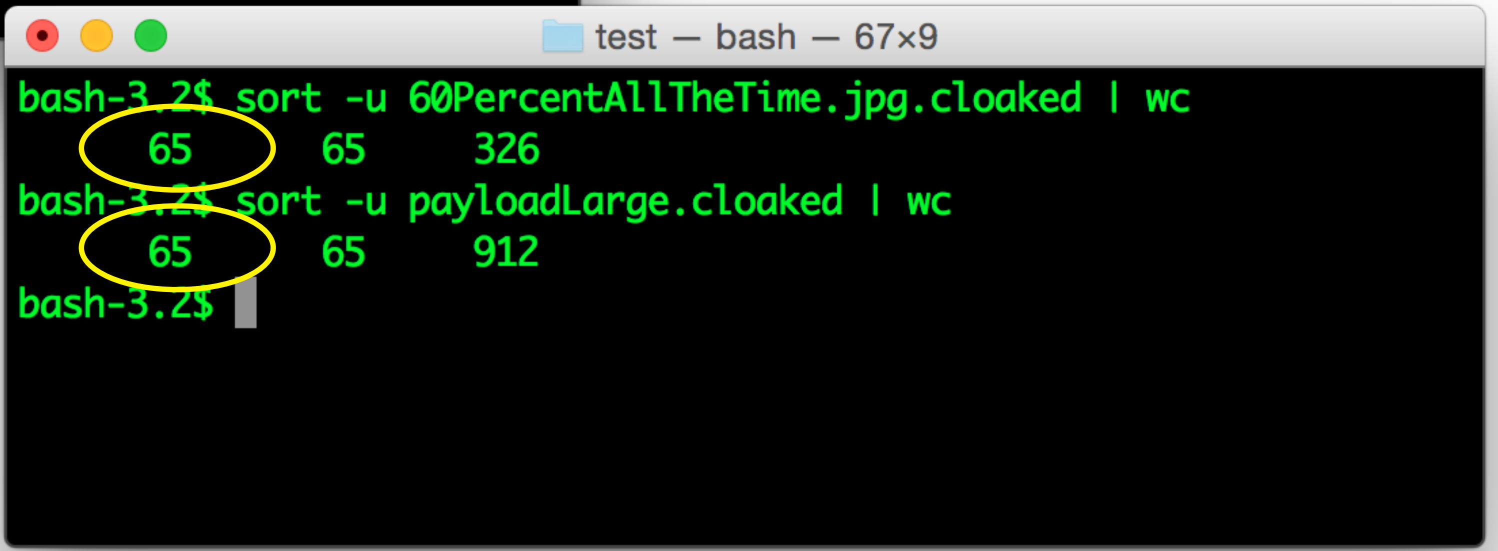
Vulnerability - Frequency Analysis Attacks

Since at most 66 strings are used to encode the data, it is computationally easy for the Machines to detect when our ciphers are in use, defeating the purpose of our obscurity

From there it’s straightforward to brute-force the cipher



```
bash-3.2$ wc 60PercentAllTheTime.jpg.cloaked
   36348   36348  182345 60PercentAllTheTime.jpg.cloaked
bash-3.2$ wc payloadLarge.cloaked
   40500   40500  576376 payloadLarge.cloaked
bash-3.2$
```

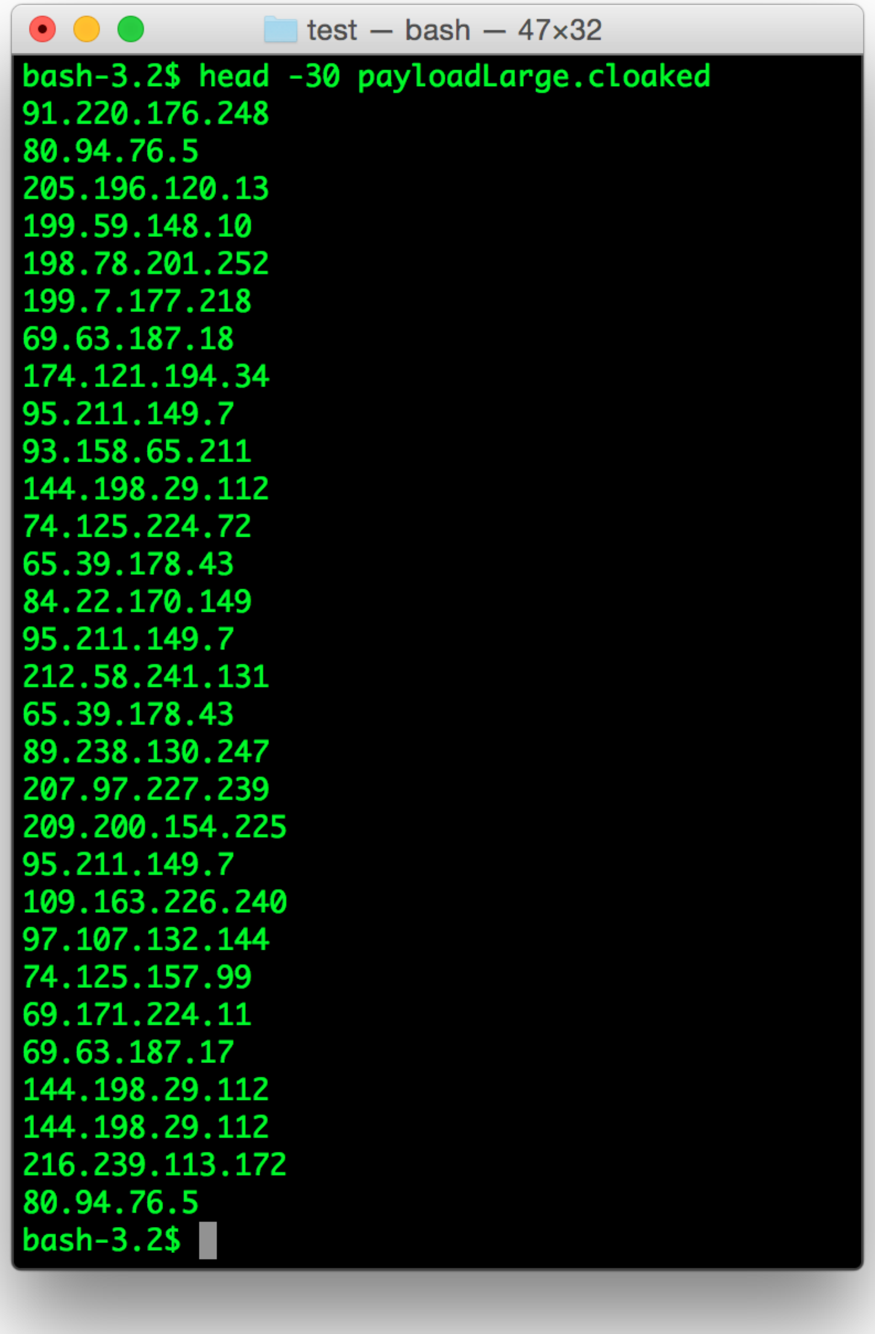


```
bash-3.2$ sort -u 60PercentAllTheTime.jpg.cloaked | wc
      65      65     326
bash-3.2$ sort -u payloadLarge.cloaked | wc
      65      65     912
bash-3.2$
```

Exfiltrating Data From Secure Networks

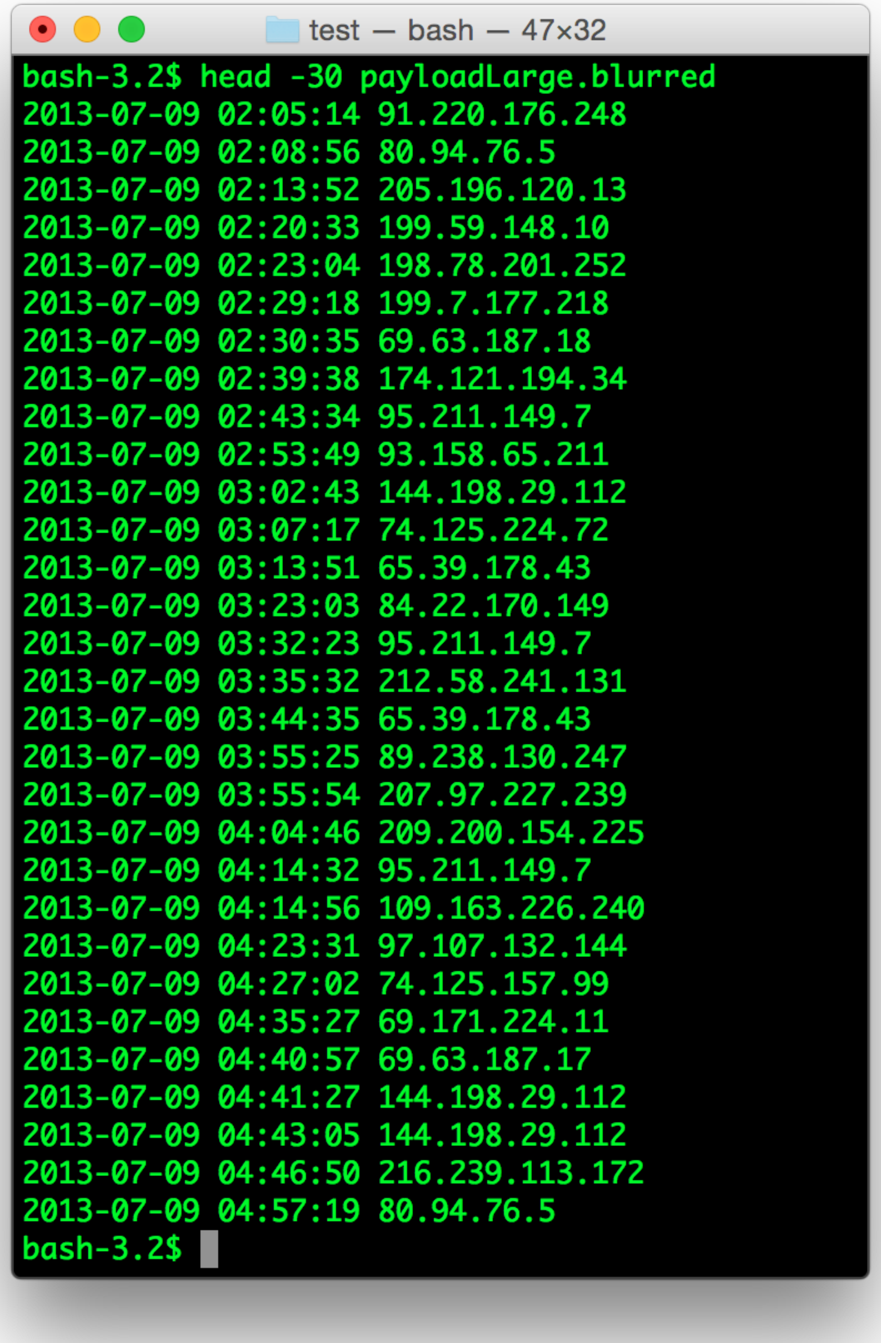
Vulnerability - Frequency Analysis Attacks (cont.)

So what if we increase that entropy by adding data blur to our cloaked file?



```
bash-3.2$ head -30 payloadLarge.cloaked
91.220.176.248
80.94.76.5
205.196.120.13
199.59.148.10
198.78.201.252
199.7.177.218
69.63.187.18
174.121.194.34
95.211.149.7
93.158.65.211
144.198.29.112
74.125.224.72
65.39.178.43
84.22.170.149
95.211.149.7
212.58.241.131
65.39.178.43
89.238.130.247
207.97.227.239
209.200.154.225
95.211.149.7
109.163.226.240
97.107.132.144
74.125.157.99
69.171.224.11
69.63.187.17
144.198.29.112
144.198.29.112
216.239.113.172
80.94.76.5
bash-3.2$
```

Prepend Randomized Sequential Timestamps



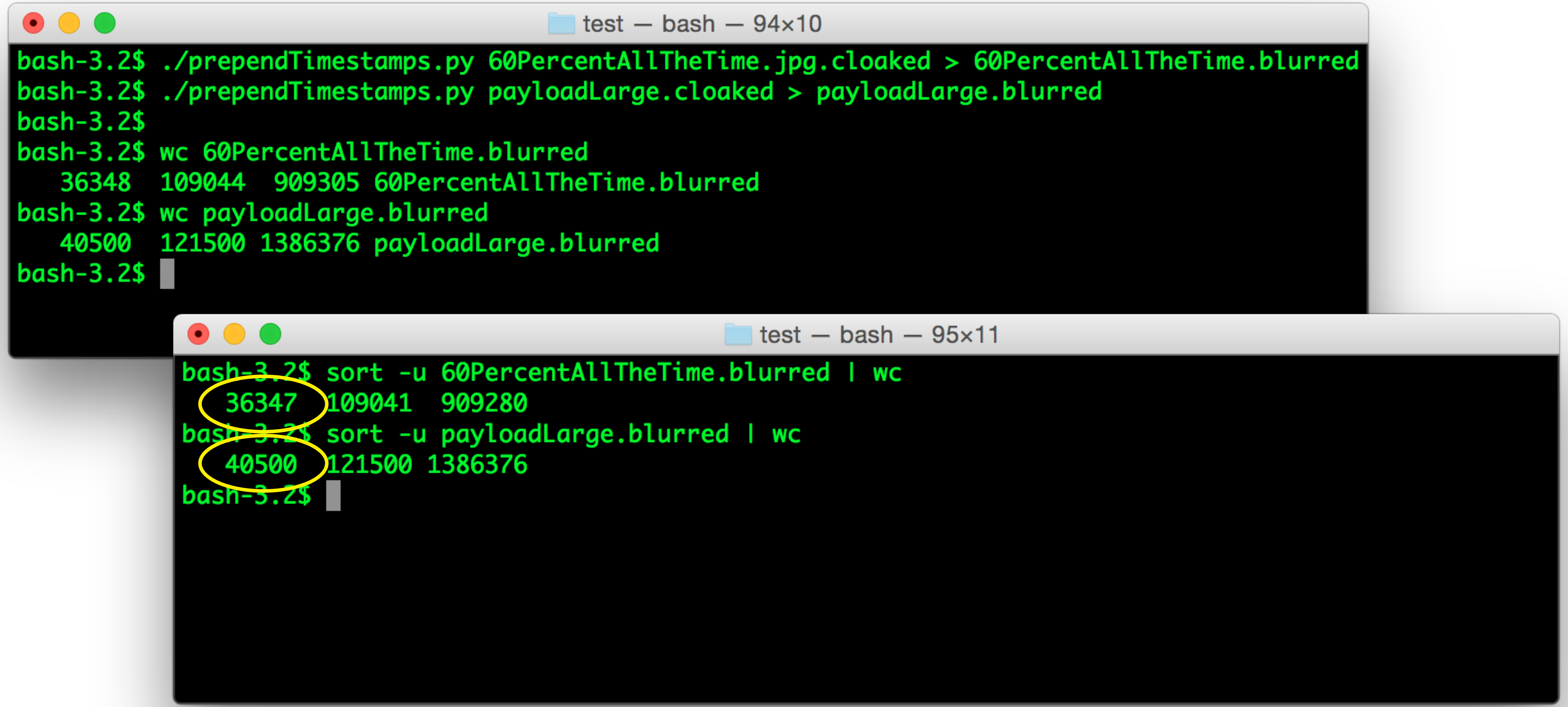
```
bash-3.2$ head -30 payloadLarge.blurred
2013-07-09 02:05:14 91.220.176.248
2013-07-09 02:08:56 80.94.76.5
2013-07-09 02:13:52 205.196.120.13
2013-07-09 02:20:33 199.59.148.10
2013-07-09 02:23:04 198.78.201.252
2013-07-09 02:29:18 199.7.177.218
2013-07-09 02:30:35 69.63.187.18
2013-07-09 02:39:38 174.121.194.34
2013-07-09 02:43:34 95.211.149.7
2013-07-09 02:53:49 93.158.65.211
2013-07-09 03:02:43 144.198.29.112
2013-07-09 03:07:17 74.125.224.72
2013-07-09 03:13:51 65.39.178.43
2013-07-09 03:23:03 84.22.170.149
2013-07-09 03:32:23 95.211.149.7
2013-07-09 03:35:32 212.58.241.131
2013-07-09 03:44:35 65.39.178.43
2013-07-09 03:55:25 89.238.130.247
2013-07-09 03:55:54 207.97.227.239
2013-07-09 04:04:46 209.200.154.225
2013-07-09 04:14:32 95.211.149.7
2013-07-09 04:14:56 109.163.226.240
2013-07-09 04:23:31 97.107.132.144
2013-07-09 04:27:02 74.125.157.99
2013-07-09 04:35:27 69.171.224.11
2013-07-09 04:40:57 69.63.187.17
2013-07-09 04:41:27 144.198.29.112
2013-07-09 04:43:05 144.198.29.112
2013-07-09 04:46:50 216.239.113.172
2013-07-09 04:57:19 80.94.76.5
bash-3.2$
```

Exfiltrating Data From Secure Networks

Vulnerability - Frequency Analysis Attacks (cont.)

‘prependTimestamps.py’ to the rescue - Entropy for all!

(Script rolls back to initial date when current date is reached - no time travel timestamps from the future regardless of file size)



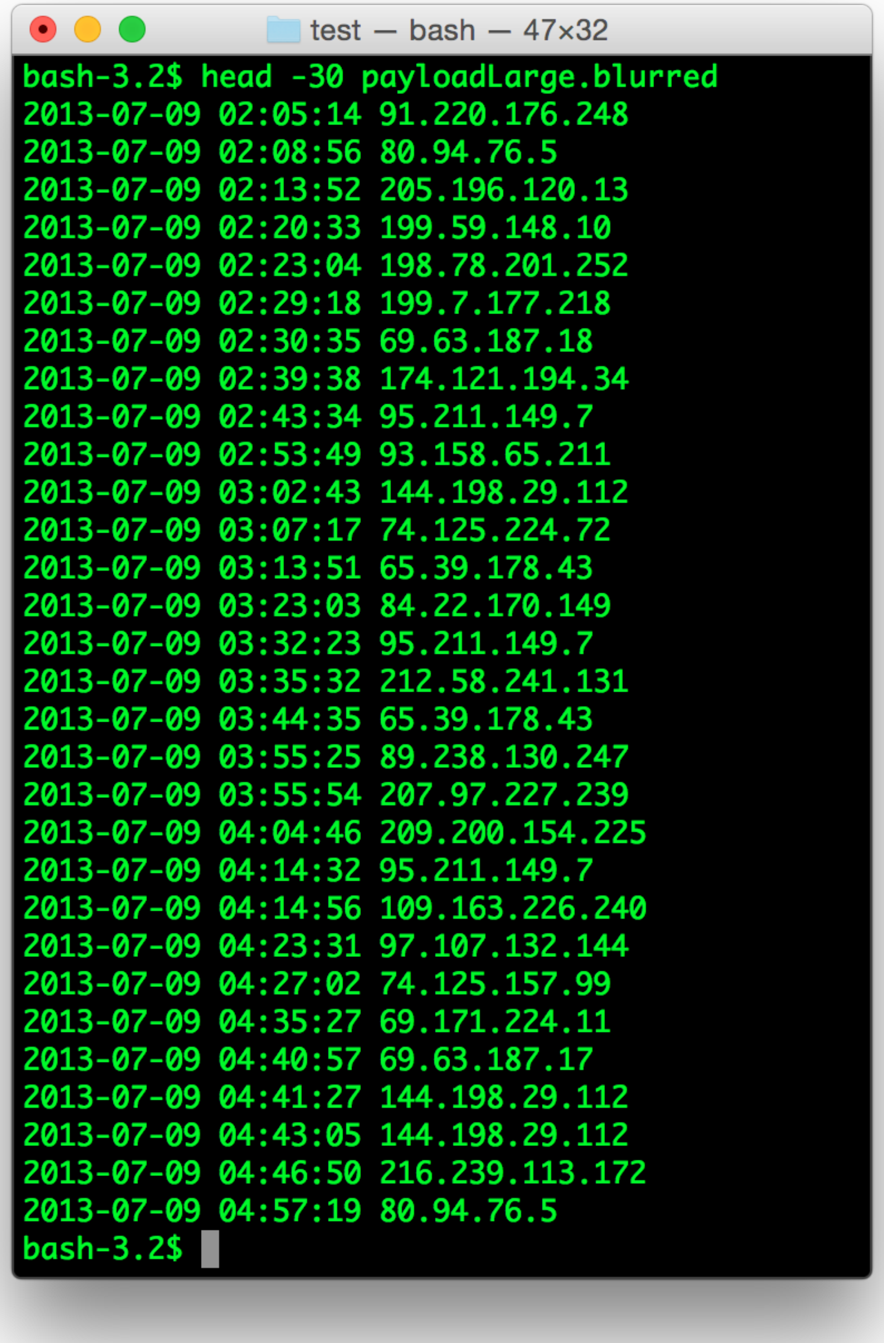
```
bash-3.2$ ./prependTimestamps.py 60PercentAllTheTime.jpg.cloaked > 60PercentAllTheTime.blurred
bash-3.2$ ./prependTimestamps.py payloadLarge.cloaked > payloadLarge.blurred
bash-3.2$
bash-3.2$ wc 60PercentAllTheTime.blurred
   36348  109044  909305 60PercentAllTheTime.blurred
bash-3.2$ wc payloadLarge.blurred
   40500  121500 1386376 payloadLarge.blurred
bash-3.2$ 
```

```
bash-3.2$ sort -u 60PercentAllTheTime.blurred | wc
   36347  109041  909280
bash-3.2$ sort -u payloadLarge.blurred | wc
   40500  121500 1386376
bash-3.2$ 
```

Exfiltrating Data From Secure Networks

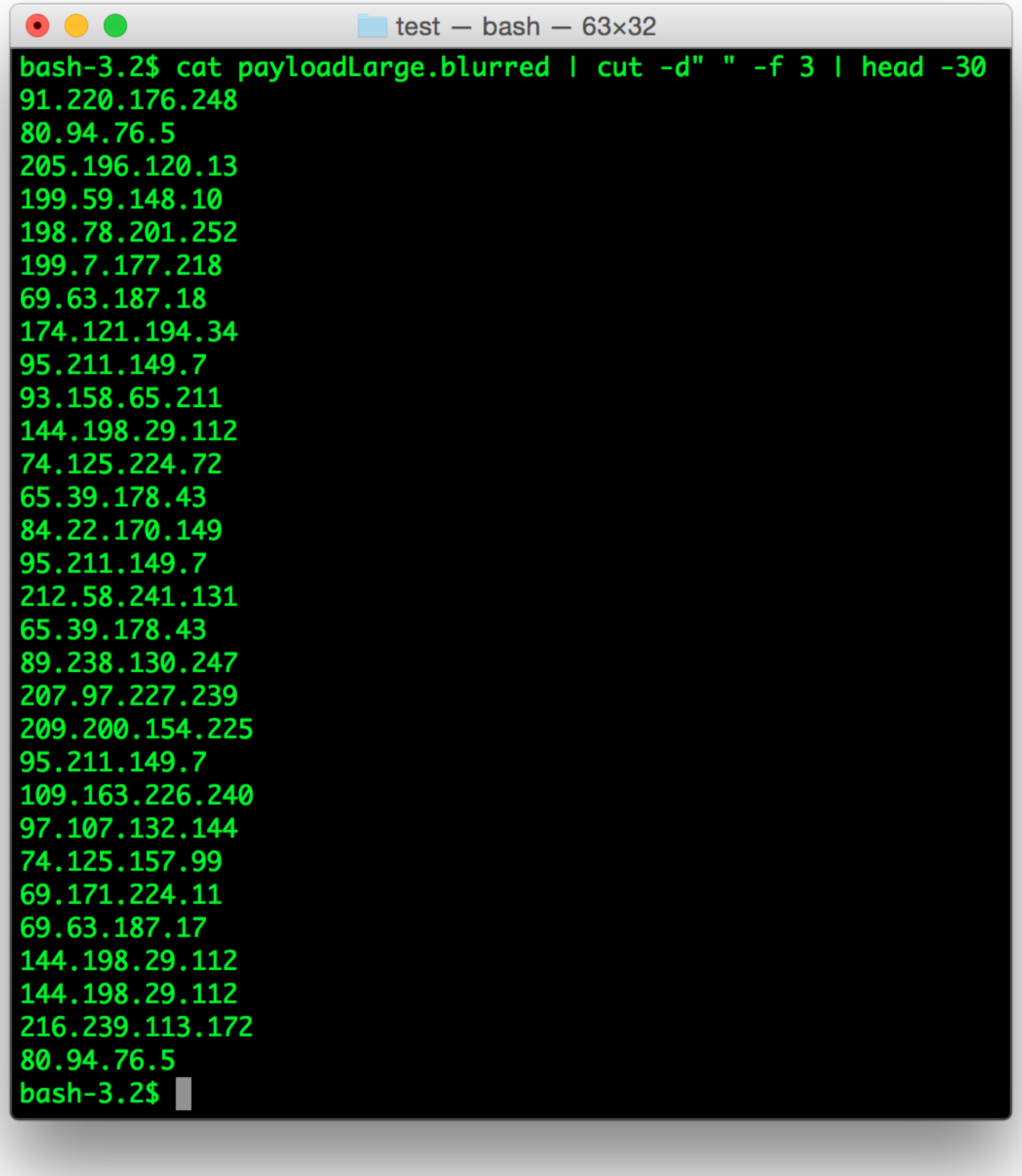
Vulnerability - Frequency Analysis Attacks (cont.)

After exfiltration, strip away the noise and decode the cloaked file



```
bash-3.2$ head -30 payloadLarge.blurred
2013-07-09 02:05:14 91.220.176.248
2013-07-09 02:08:56 80.94.76.5
2013-07-09 02:13:52 205.196.120.13
2013-07-09 02:20:33 199.59.148.10
2013-07-09 02:23:04 198.78.201.252
2013-07-09 02:29:18 199.7.177.218
2013-07-09 02:30:35 69.63.187.18
2013-07-09 02:39:38 174.121.194.34
2013-07-09 02:43:34 95.211.149.7
2013-07-09 02:53:49 93.158.65.211
2013-07-09 03:02:43 144.198.29.112
2013-07-09 03:07:17 74.125.224.72
2013-07-09 03:13:51 65.39.178.43
2013-07-09 03:23:03 84.22.170.149
2013-07-09 03:32:23 95.211.149.7
2013-07-09 03:35:32 212.58.241.131
2013-07-09 03:44:35 65.39.178.43
2013-07-09 03:55:25 89.238.130.247
2013-07-09 03:55:54 207.97.227.239
2013-07-09 04:04:46 209.200.154.225
2013-07-09 04:14:32 95.211.149.7
2013-07-09 04:14:56 109.163.226.240
2013-07-09 04:23:31 97.107.132.144
2013-07-09 04:27:02 74.125.157.99
2013-07-09 04:35:27 69.171.224.11
2013-07-09 04:40:57 69.63.187.17
2013-07-09 04:41:27 144.198.29.112
2013-07-09 04:43:05 144.198.29.112
2013-07-09 04:46:50 216.239.113.172
2013-07-09 04:57:19 80.94.76.5
bash-3.2$ 
```

Strip Timestamps, Ready for Decloaking!



```
bash-3.2$ cat payloadLarge.blurred | cut -d" " -f 3 | head -30
91.220.176.248
80.94.76.5
205.196.120.13
199.59.148.10
198.78.201.252
199.7.177.218
69.63.187.18
174.121.194.34
95.211.149.7
93.158.65.211
144.198.29.112
74.125.224.72
65.39.178.43
84.22.170.149
95.211.149.7
212.58.241.131
65.39.178.43
89.238.130.247
207.97.227.239
209.200.154.225
95.211.149.7
109.163.226.240
97.107.132.144
74.125.157.99
69.171.224.11
69.63.187.17
144.198.29.112
144.198.29.112
216.239.113.172
80.94.76.5
bash-3.2$ 
```

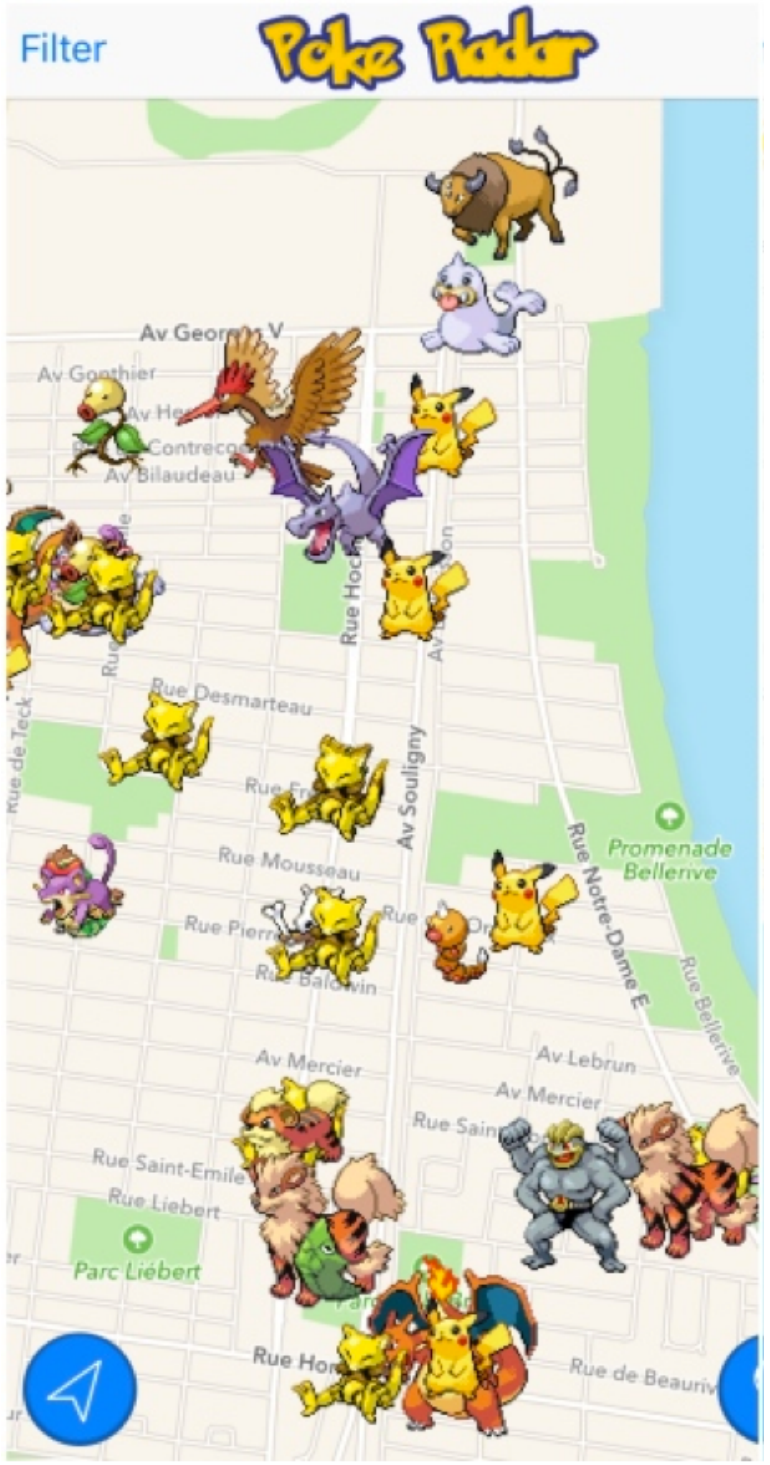
Exfiltrating Data From Secure Networks

The Ideal Cloakify Use Case

- Cipher is socially / culturally relevant
- Cipher is readily identifiable and dismissible
- Blends well with one of the ‘noiseTools’ scripts to increase entropy

If only there existed such a thing…
Something in front of everyone…
Everywhere…
Make it stop…

Exfiltrating Data From Secure Networks

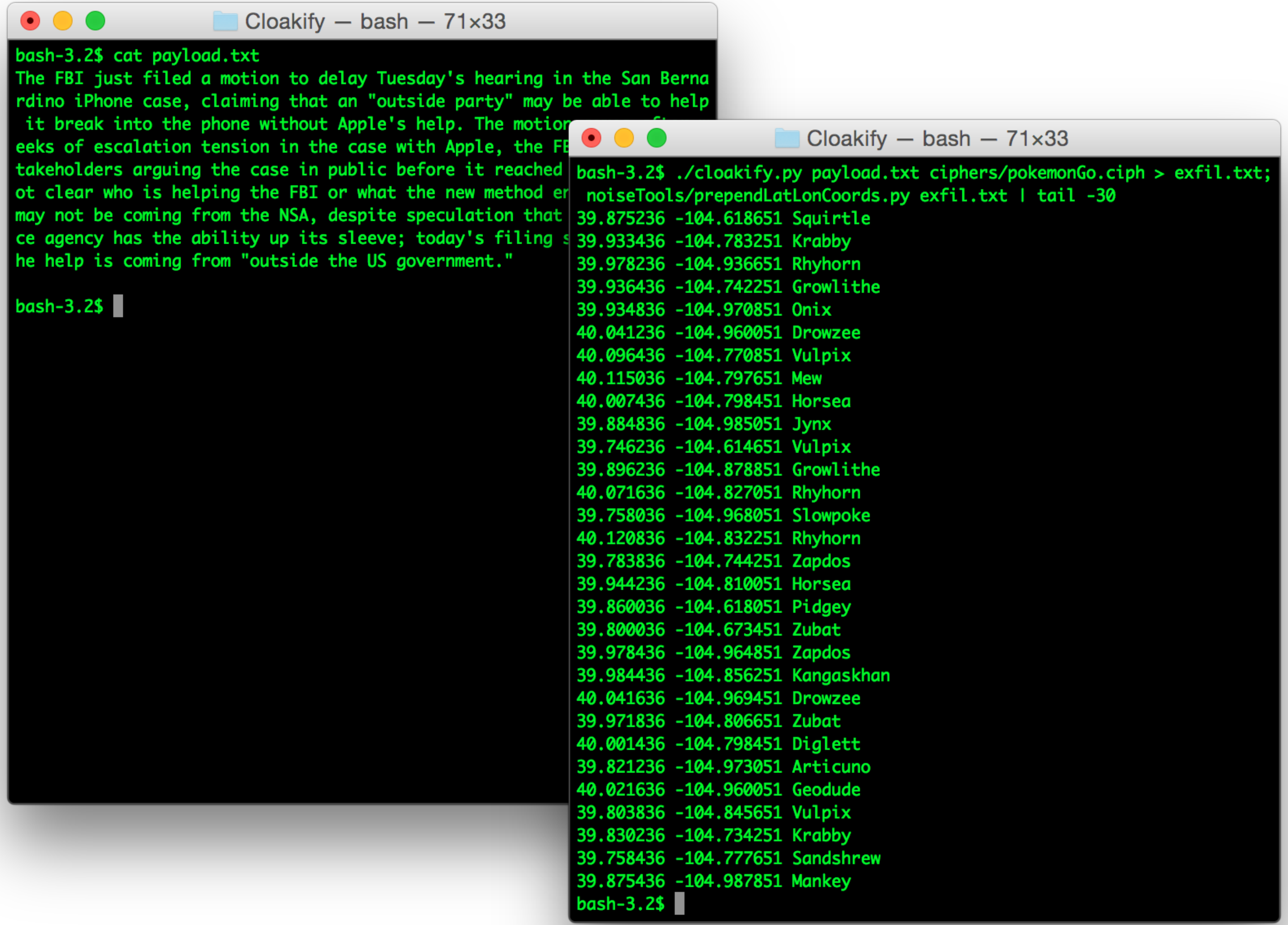


Poke Radar, a crowdsourced Pokemon location
hotspot for Abras. Vancouver, on the other hanc
(Poke Radar/CBC Screenshot)

Exfiltrating Data From Secure Networks

“soooo much exfil joy”

Exfiltrating Data From Secure Networks



```
bash-3.2$ cat payload.txt
The FBI just filed a motion to delay Tuesday's hearing in the San Berna
rdino iPhone case, claiming that an "outside party" may be able to help
 it break into the phone without Apple's help. The motion
eeks of escalation tension in the case with Apple, the FE
takeholders arguing the case in public before it reached
ot clear who is helping the FBI or what the new method er
may not be coming from the NSA, despite speculation that
ce agency has the ability up its sleeve; today's filing s
he help is coming from "outside the US government."

bash-3.2$ 
```

```
bash-3.2$ ./cloakify.py payload.txt ciphers/pokemonGo.ciph > exfil.txt;
 noiseTools/prependLatLonCoords.py exfil.txt | tail -30
39.875236 -104.618651 Squirtle
39.933436 -104.783251 Krabby
39.978236 -104.936651 Rhyhorn
39.936436 -104.742251 Growlithe
39.934836 -104.970851 Onix
40.041236 -104.960051 Drowzee
40.096436 -104.770851 Vulpix
40.115036 -104.797651 Mew
40.007436 -104.798451 Horsea
39.884836 -104.985051 Jynx
39.746236 -104.614651 Vulpix
39.896236 -104.878851 Growlithe
40.071636 -104.827051 Rhyhorn
39.758036 -104.968051 Slowpoke
40.120836 -104.832251 Rhyhorn
39.783836 -104.744251 Zapdos
39.944236 -104.810051 Horsea
39.860036 -104.618051 Pidgey
39.800036 -104.673451 Zubat
39.978436 -104.964851 Zapdos
39.984436 -104.856251 Kangaskhan
40.041636 -104.969451 Drowzee
39.971836 -104.806651 Zubat
40.001436 -104.798451 Diglett
39.821236 -104.973051 Articuno
40.021636 -104.960051 Geodude
39.803836 -104.845651 Vulpix
39.830236 -104.734251 Krabby
39.758436 -104.777651 Sandshrew
39.875436 -104.987851 Mankey
bash-3.2$ 
```

Exfiltrating Data From Secure Networks

Cloakify Toolset available via GitHub: https://github.com/TryCatchHCF/Cloakify/

Prepackaged ciphers include lists of:

Desserts in English, Arabic, Thai, Russian, Hindi, Chinese, Persian, and Muppet (Swedish Chef)	
GeoCoords World Capitols (Lat/Lon)	**GeoCaching Coordinates (w/ Site Names)**
MD5 Password Hashes	**PokemonGo Monsters**
Star Trek characters	**Emoji**
IPv4 Addresses of Popular Websites	**World Football Teams (‘Soccer’ to us Yanks)**
Ski Resorts	**Top World Beaches**
Belgian Beers	**(The Rest Is Up To Your Imagination)**

Prepackaged scripts for adding noise / entropy to your cloaked payloads:

prependID.py - Adds a randomized ID tag to front of each line
prependLatLonCoords.py - Adds randomized LatLong coordinates to front of each line
prependTimestamps.py - Adds timestamps (log file style) to front of each line

See script comments for details on how to tailor the output for your own needs

Exfiltrating Data From Secure Networks

Creating your own Cloakify cipher

- Generate a list of at least 66 unique words / phrases / symbols (Unicode accepted)
- Randomize the list order (see “randomizeCommandExample.txt” in project directory)
- Remove all duplicate entries and all blank lines
- Pass the new file as the cipher argument to cloakify / decloakify
- Randomize ciphers between engagements

If you don’t want to (or can’t) pull the cipher across network boundaries, generate the cipher in place at the point of data compromise, then record the cipher for later decoding.

Less incriminating if your notebook contains a list of desserts rather than a list of MD5 hashes, so choose ciphers wisely.

Exfiltrating Data From Secure Networks

The Takeaway

"Insanity is repeating the same mistakes and expecting different results"

We need better DLP solutions and related controls

Use Cloakify in your Red Team engagements, your Blue Team drills, your vendor assessments, wherever that you can demonstrate its impact

The more visibility on the problem, the better we can develop solutions to solve the actual issues

Let’s move DLP a few generations ahead of the problem instead of repeatably fixing a barn door

Exfiltrating Data From Secure Networks

Cloakify Toolset

Signature-based AV + DLP + Data Whitelisting
All Bypassed By
A Small Python Script and a Muppet

(Seriously, We Can Fix This)