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)

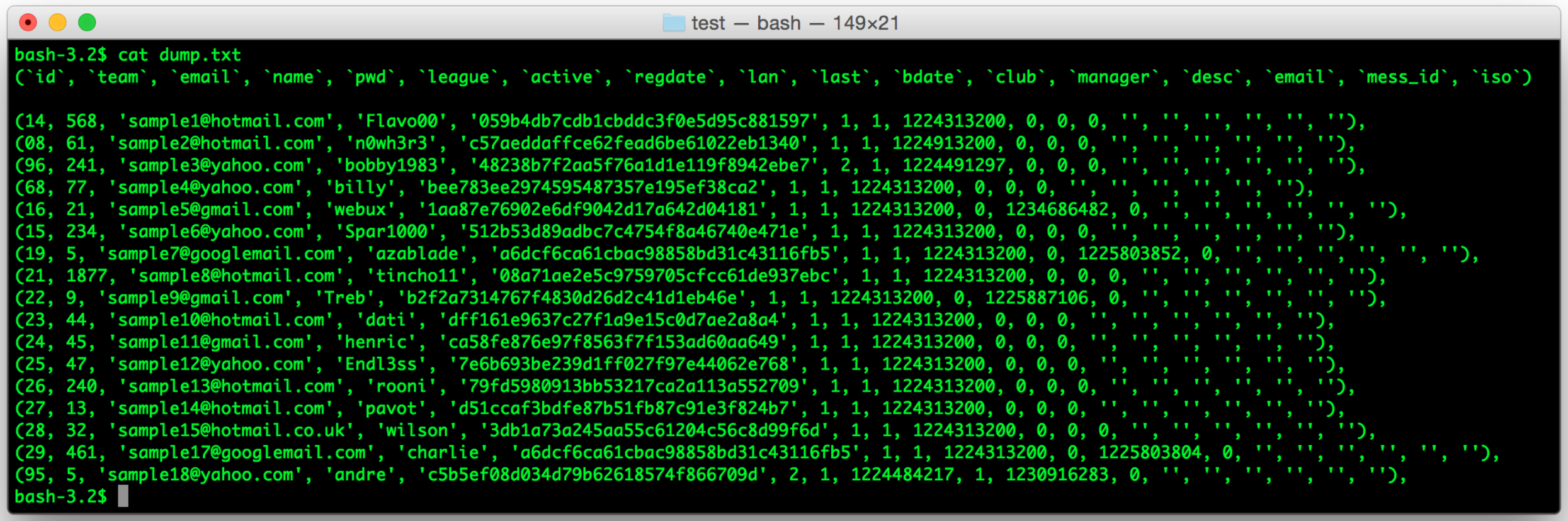
Former USMC intelligence analyst, counterintelligence specialist

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



Exfiltrating Data From Secure Networks

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

test — bash — 149×21

```
bash-3.2$ head -19 dump.cloaked
Iron Range Off-Highway Vehicle State Recreation Area N 47 ° 28.721 W 092 ° 26.319
Minnesota Valley State Recreation Area N 44 ° 39.144 W 093 ° 42.849
Hill Annex Mine State Park N 47 ° 19.585 W 093 ° 16.697
Brown's Creek State Trail N 45 ° 04.531 W 092 ° 49.776
Lake Vermilion-Soudan Underground Mine State Park N 47 ° 49.197 W 092 ° 14.320
Minnesota Valley State Recreation Area N 44 ° 39.144 W 093 ° 42.849
Sakatah Lake State Park N 44 ° 13.504 W 093 ° 31.181
Itasca State Park N 47 ° 11.488 W 095 ° 10.199
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
Hill Annex Mine State Park N 47 ° 19.585 W 093 ° 16.697
Red River State Recreation Area N 47 ° 55.743 W 097 ° 01.723
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
Franz Jevne State Park N 48 ° 38.526 W 094 ° 03.773
Split Rock Creek State Park N 43 ° 53.664 W 096 ° 21.941
Big Stone Lake State Park N 45 ° 23.126 W 096 ° 32.099
Root River State Trail N 43 ° 46.298 W 091 ° 38.153
Split Rock Lighthouse State Park N 47 ° 11.912 W 091 ° 22.479
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

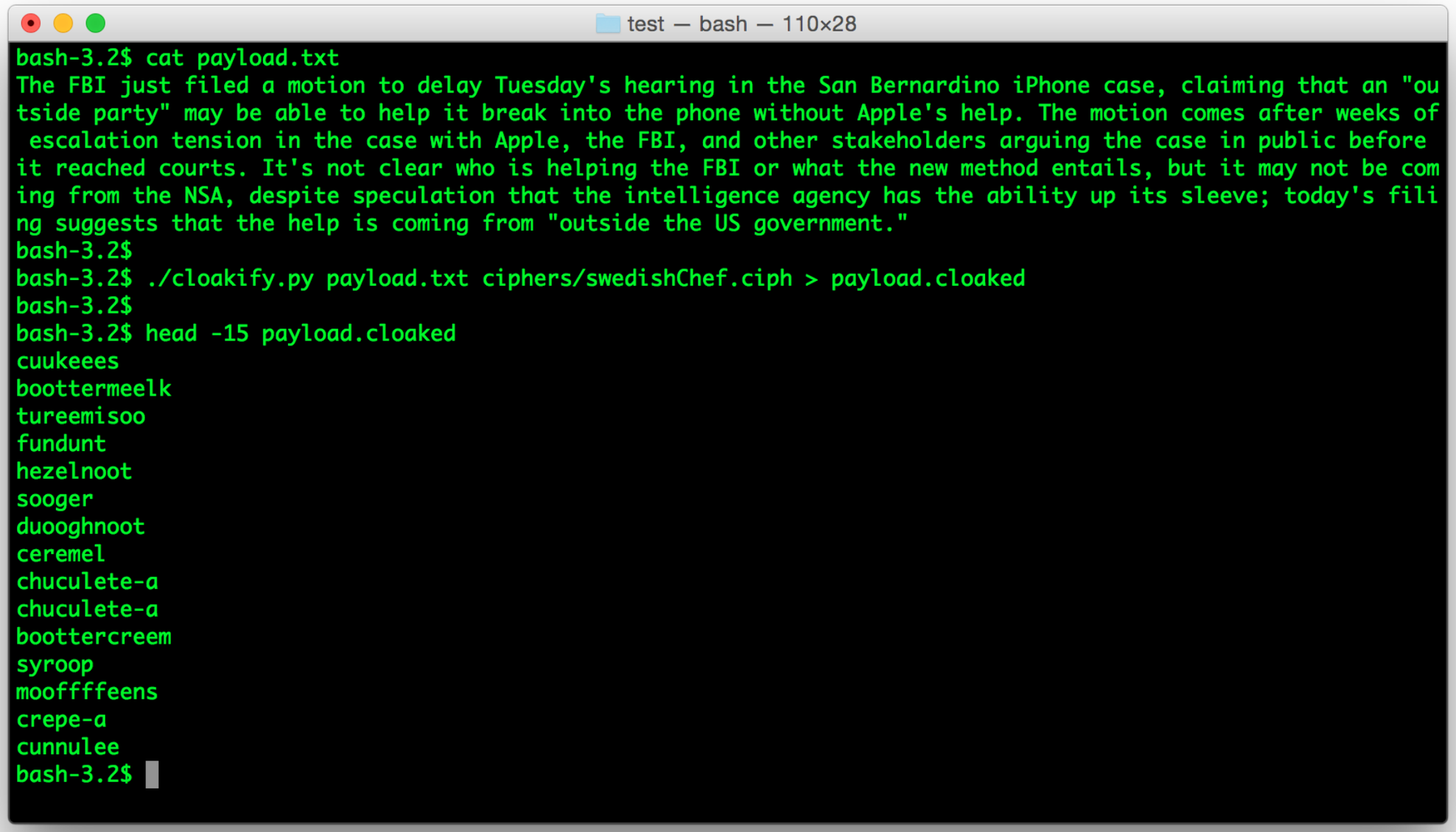
Solutions being put in place are generally addressing the obvious

- 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 file into a list of arbitrary strings

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



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

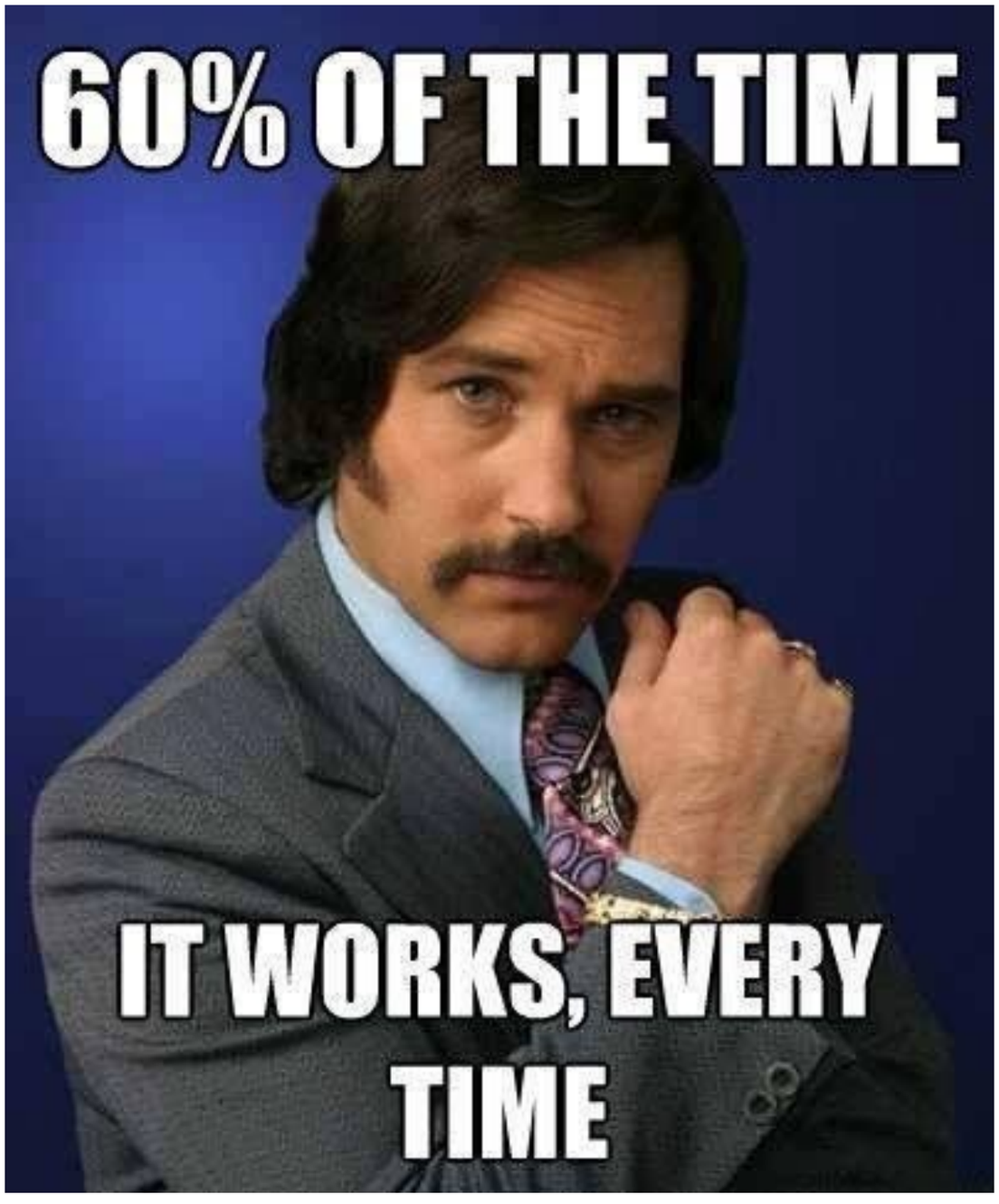
- Increases payload size (directly proportional to size of cipher elements)

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 of any kind 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!

Top 100 IP Addresses

```
Cloakify — bash — 76×22
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$
```

Swedish Chef (Muppet Desserts)

```
Cloakify — bash — 76×22
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$
```

Star Trek Characters

```
Cloakify — bash — 76×22
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$
```

Emoji Faces

```
Cloakify — bash — 76×22
bash-3.2$ ./cloakify.py payload.txt ciphers/emoji.ciph | tail -20
bash-3.2$
```

Amphibians (Latin)

```
Cloakify — bash — 76×22
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$
```

Geocaching Sites

```
Cloakify — bash — 76×22
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!

Cloakify — bash — 79×22

```
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$
```

World Beaches

Cloakify — bash — 74×22

```
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$
```

Belgian Beers

Cloakify — bash — 79×22

```
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$
```

World Cup Teams

Cloakify — bash — 71×22

```
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$
```

Ski Resorts

Exfiltrating Data From Secure Networks

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

Cloakify — bash — 70×22

```
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
ขนมปัง
ผลประกอบการ
ผลไม้แช่อิ่ม
พาย
คัสตาร์
bash-3.2$
```

Thai

Cloakify — bash — 76×22

```
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$
```

Chinese
(Cantonese or Mandarin?)

Cloakify — bash — 69×22

```
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$
```

Russian

Cloakify — bash — 69×22

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

Hindi

Cloakify — bash — 76×22

```
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$
```

Persian

Cloakify — bash — 76×22

```
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
برولیه
فطيرة الجبن
bash-3.2$
```

Arabic

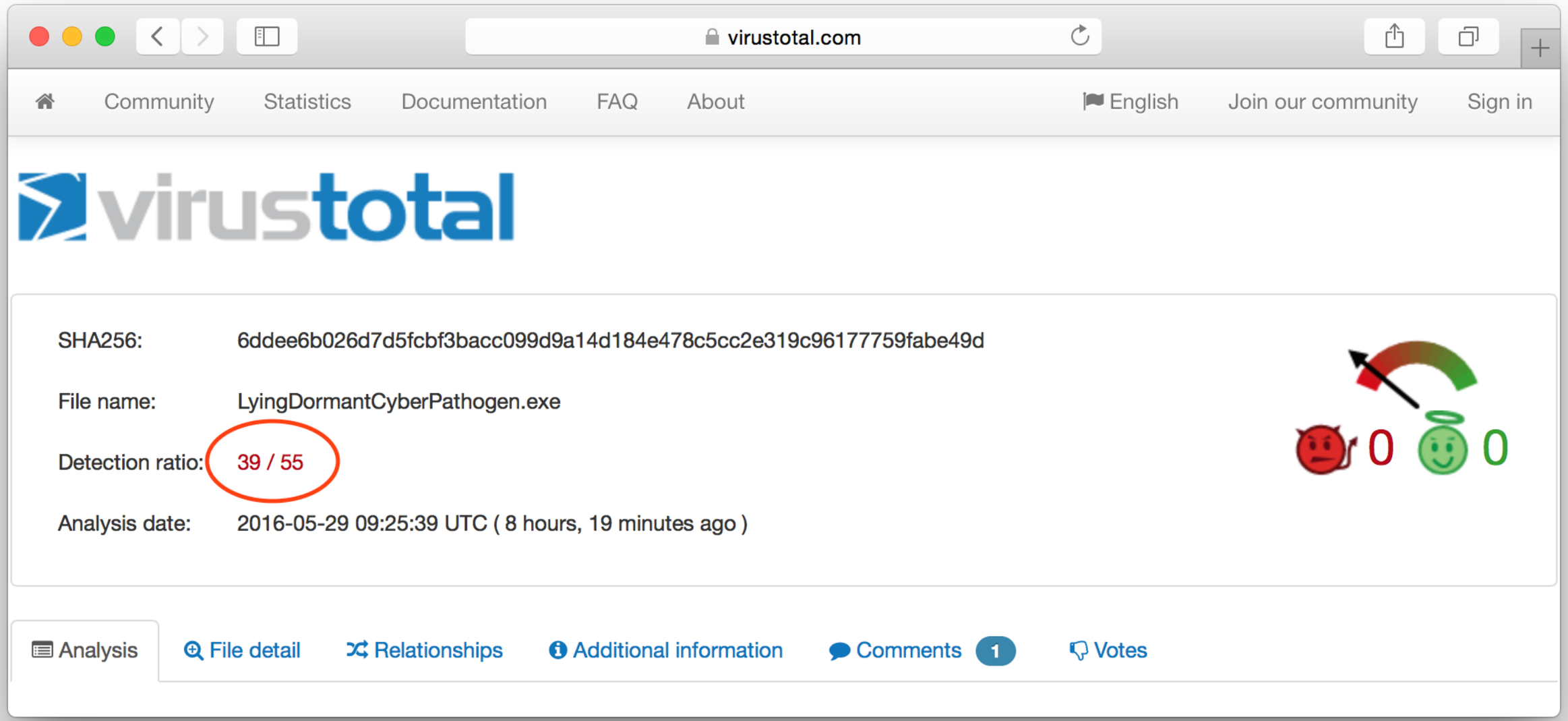
Exfiltrating Data From Secure Networks

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

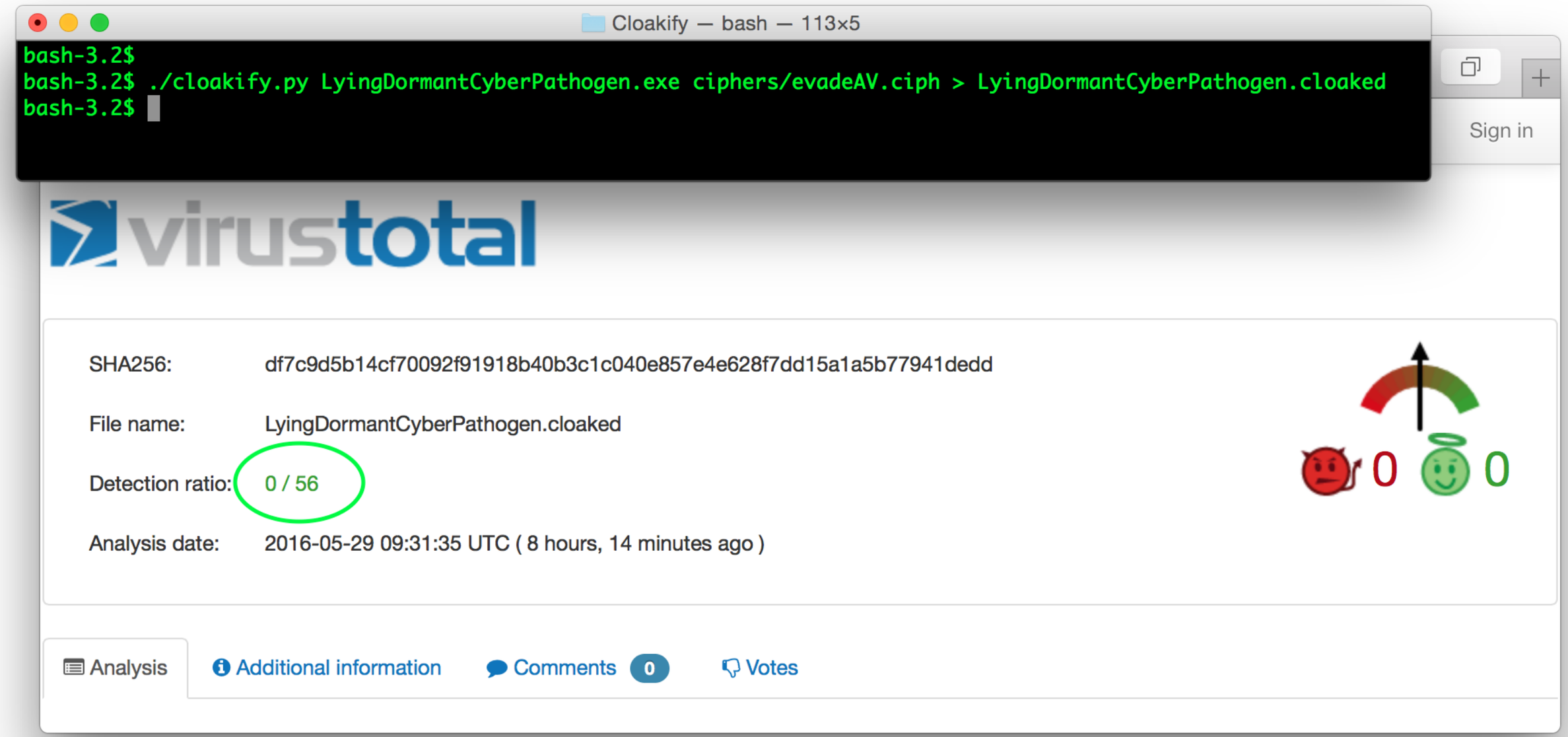


Exfiltrating Data From Secure Networks

AV Evasion (cont.)

Solution: Encode the payload with ‘evadeAV’ cipher

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



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.

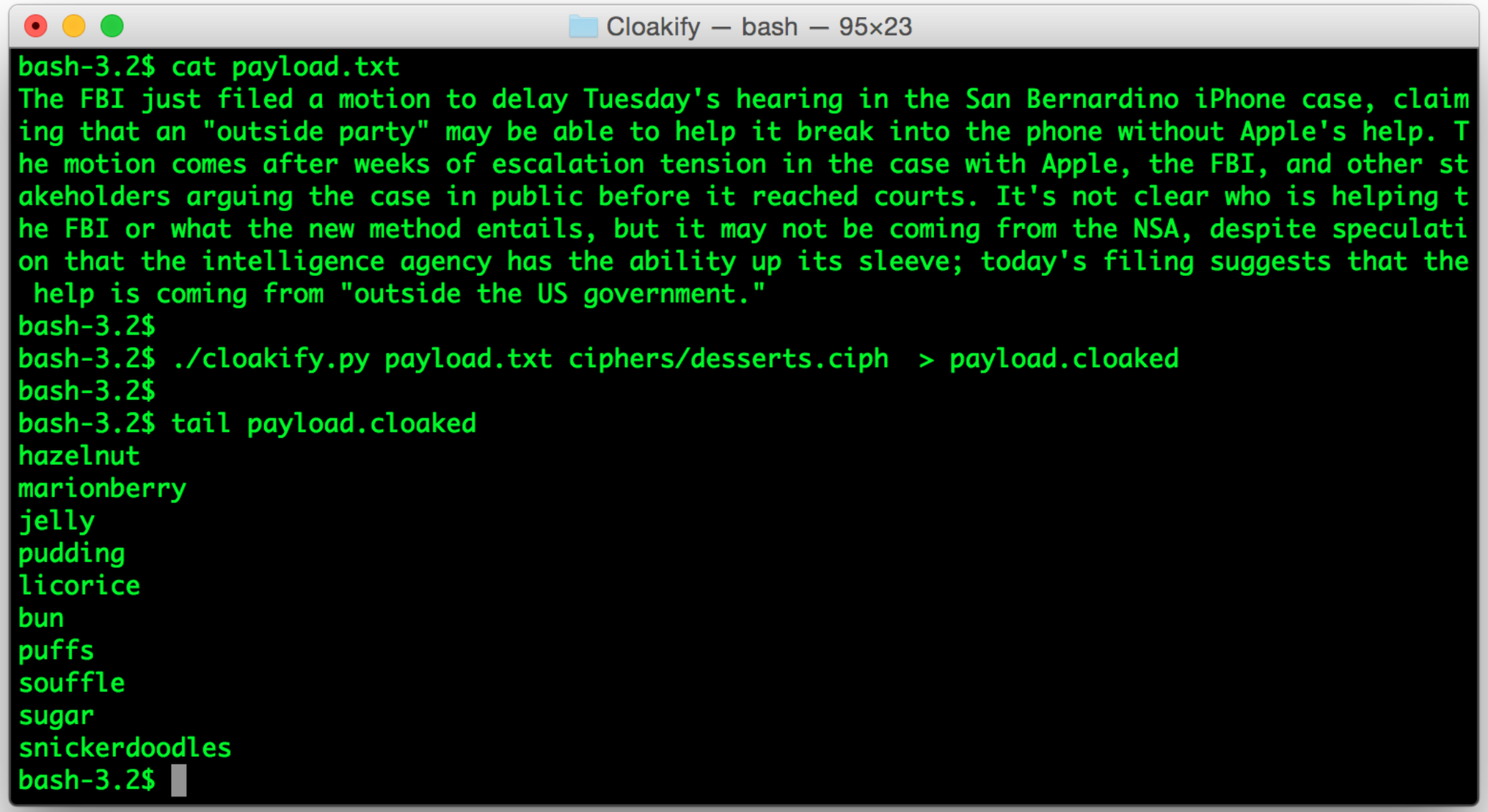
“Ticket closed! Beer now closer! Good job, me!”

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).

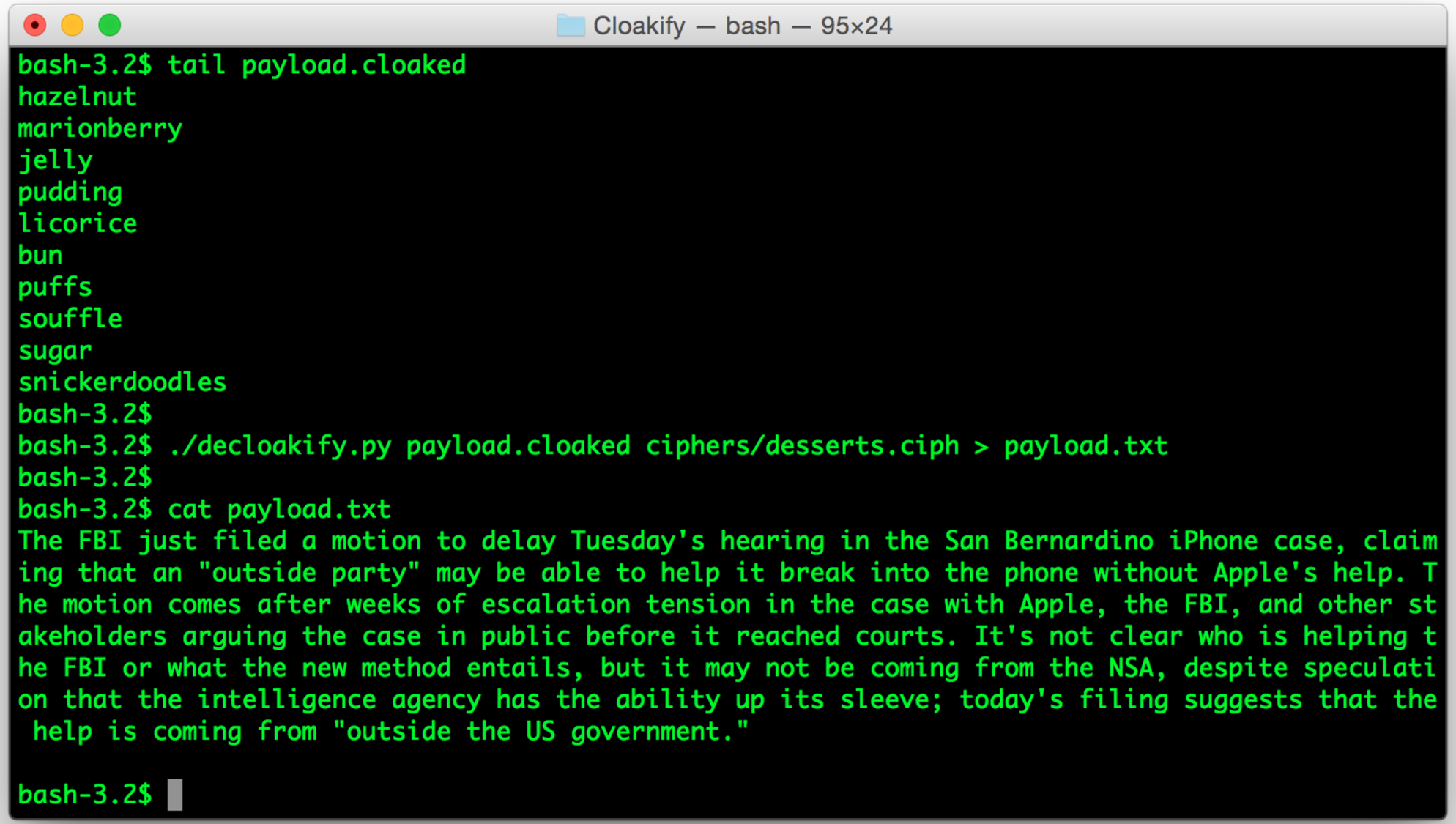
Solution: Encode your payload with any of the benign ciphers, exfiltrate



Exfiltrating Data From Secure Networks

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

“The Snickerdoodles are a lie? Noooooooo!” (But the payload is still delicious.)



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.”

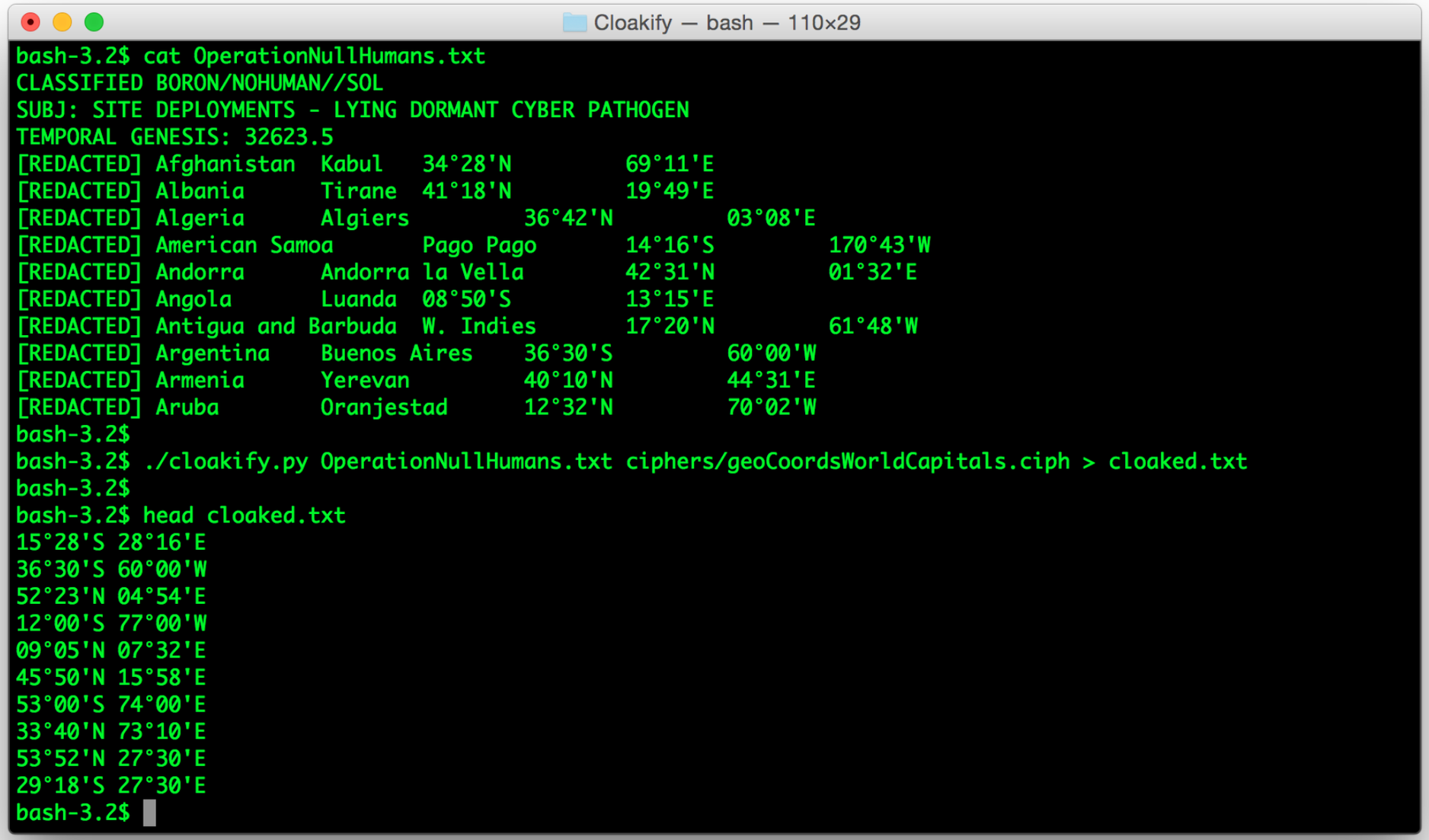
Exfiltrating Data From Secure Networks

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.

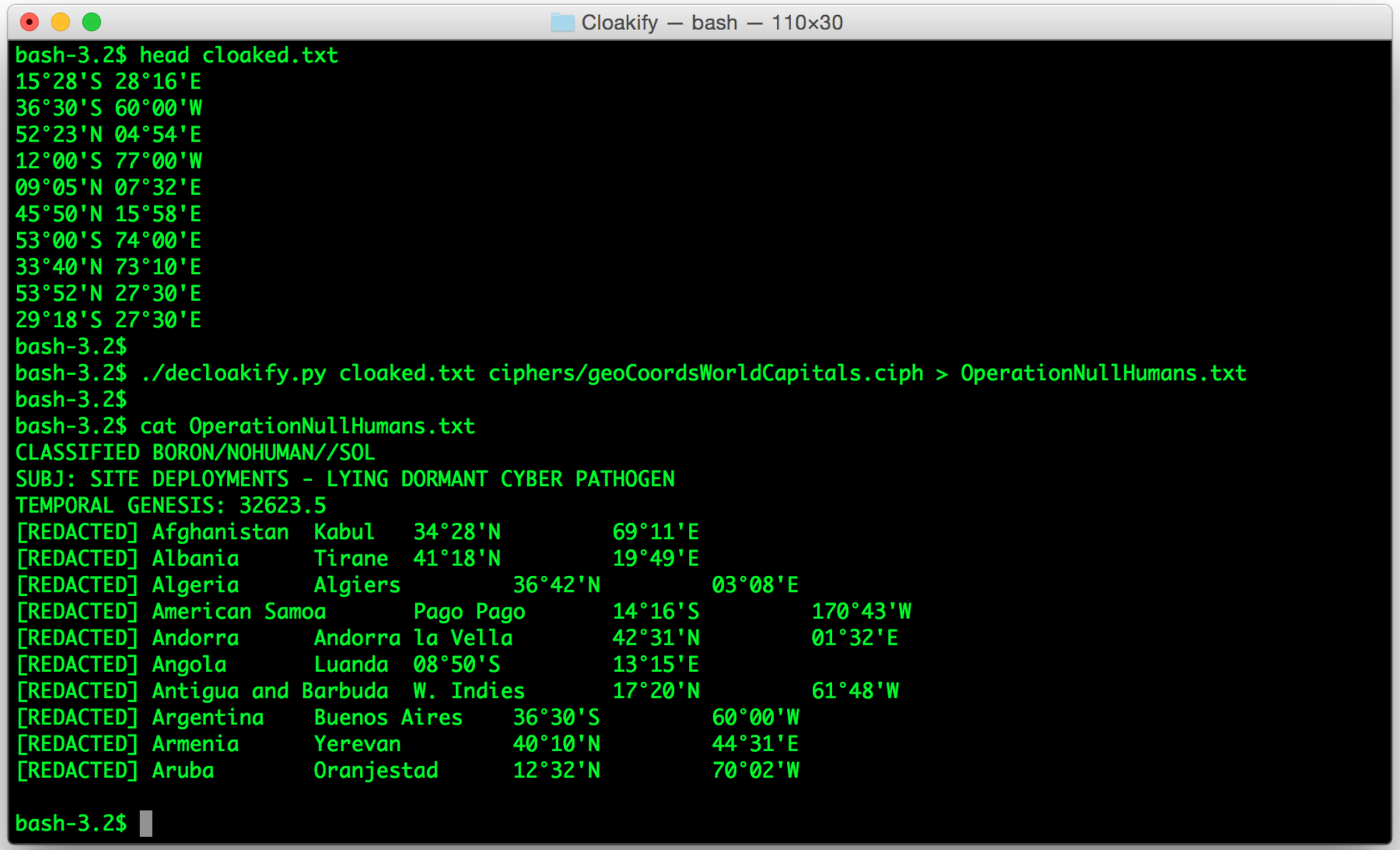


Exfiltrating Data From Secure Networks

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.



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.

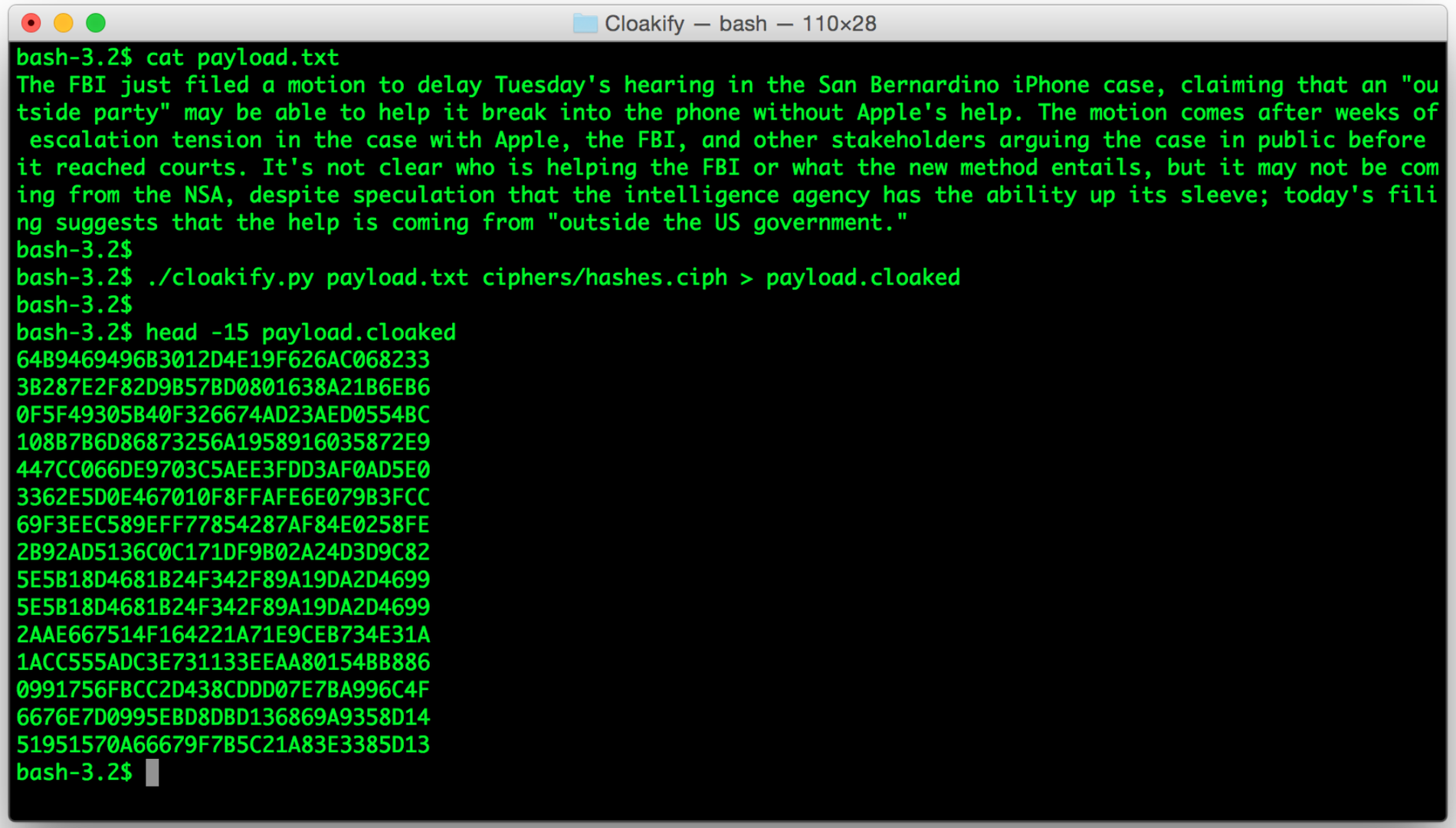
“Phase plasma rifle in the 40 watt range.”

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



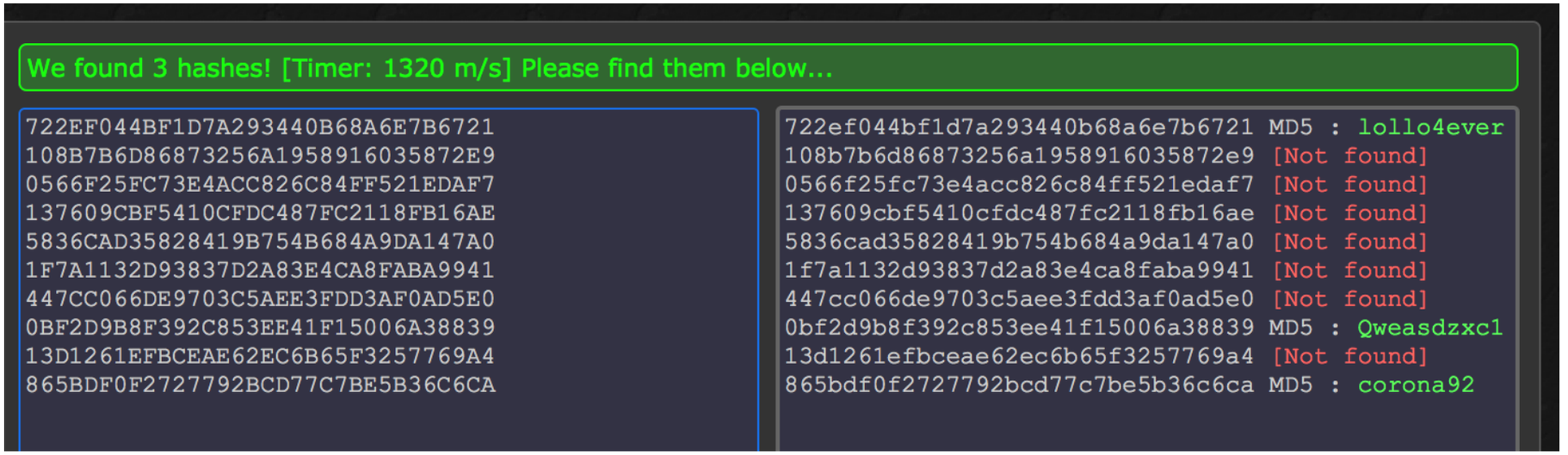
Exfiltrating Data From Secure Networks

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

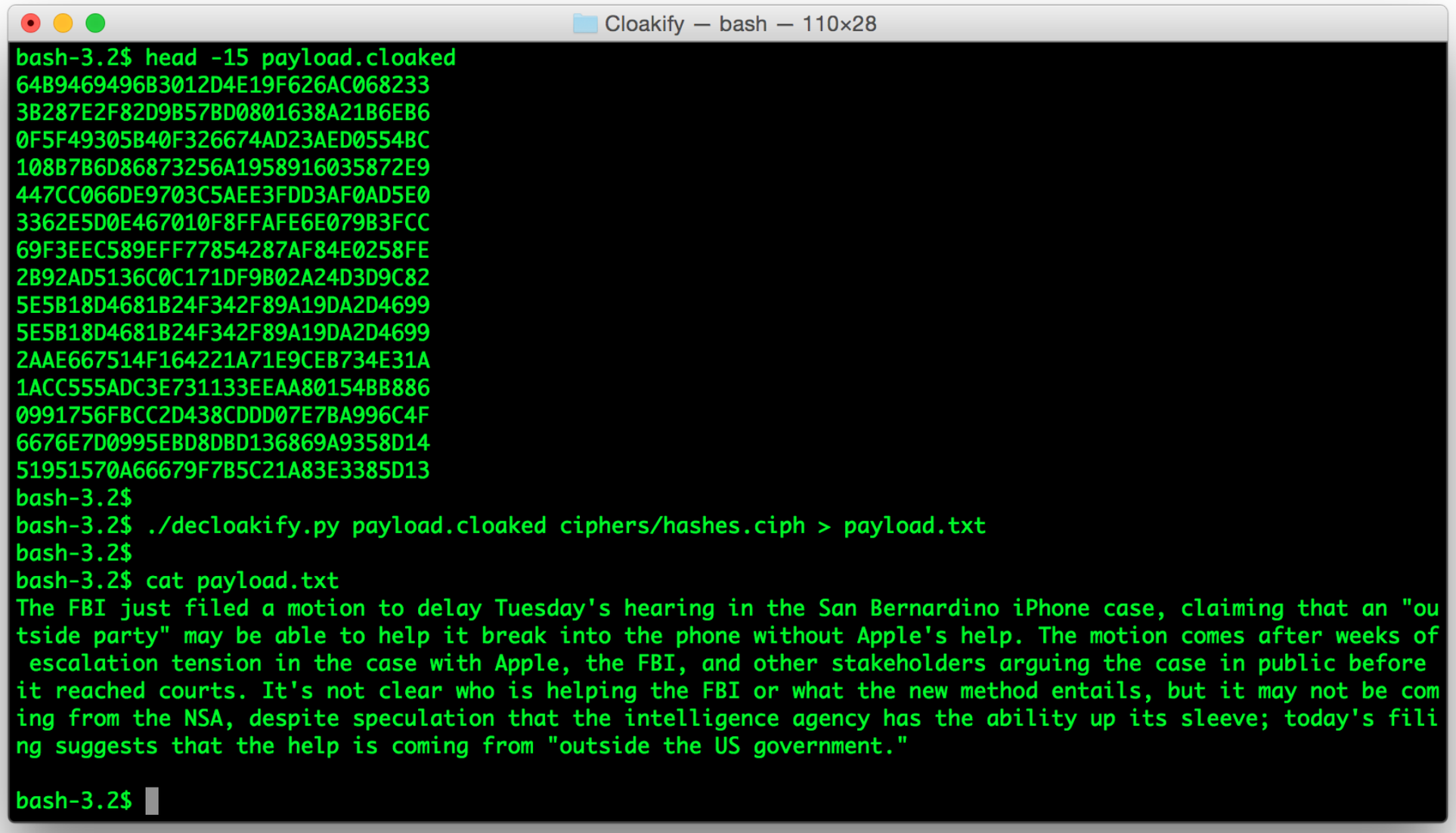


With attention now focused on the exfiltrated “password hashes”, the data is likely to end up stored away for record keeping and final incident report, spared from further examination

Meanwhile…

Exfiltrating Data From Secure Networks

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



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?

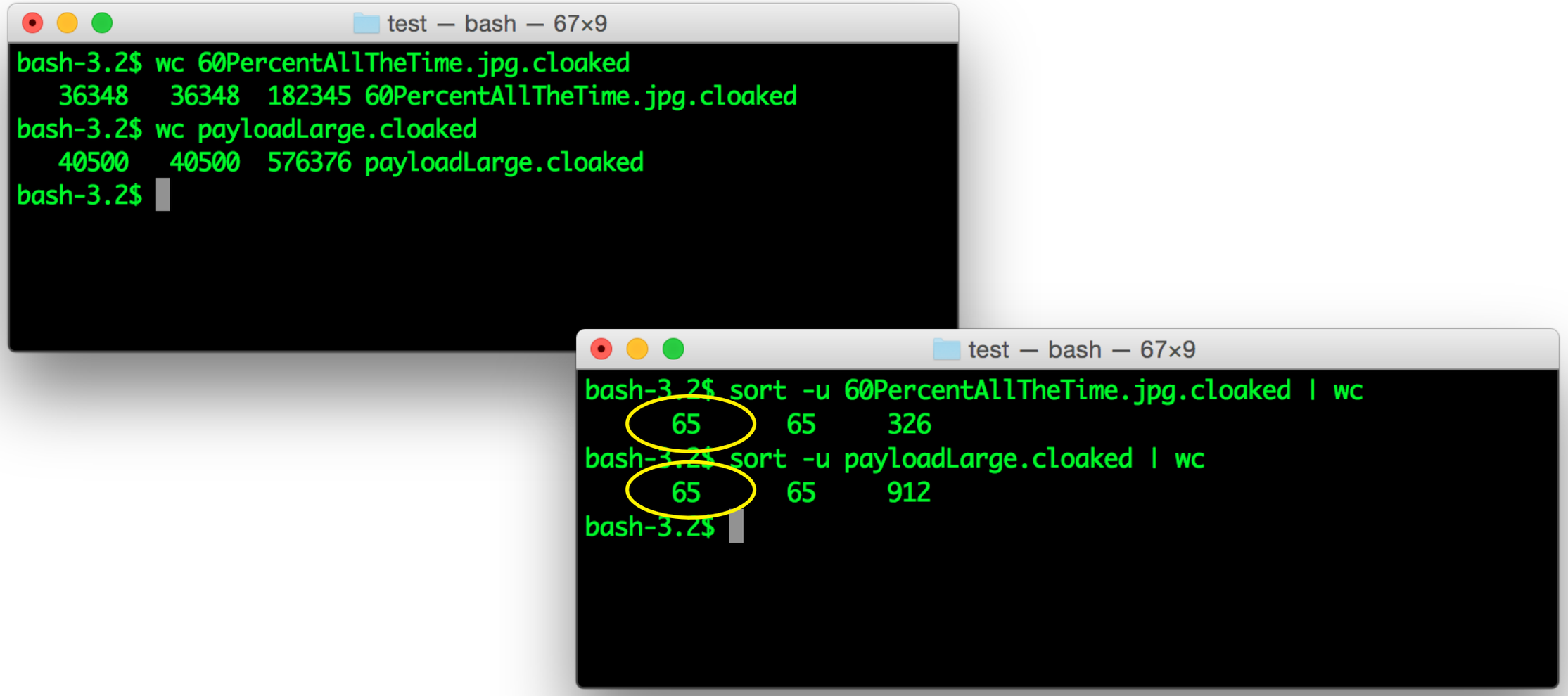
Frequency Analysis Vulnerability

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

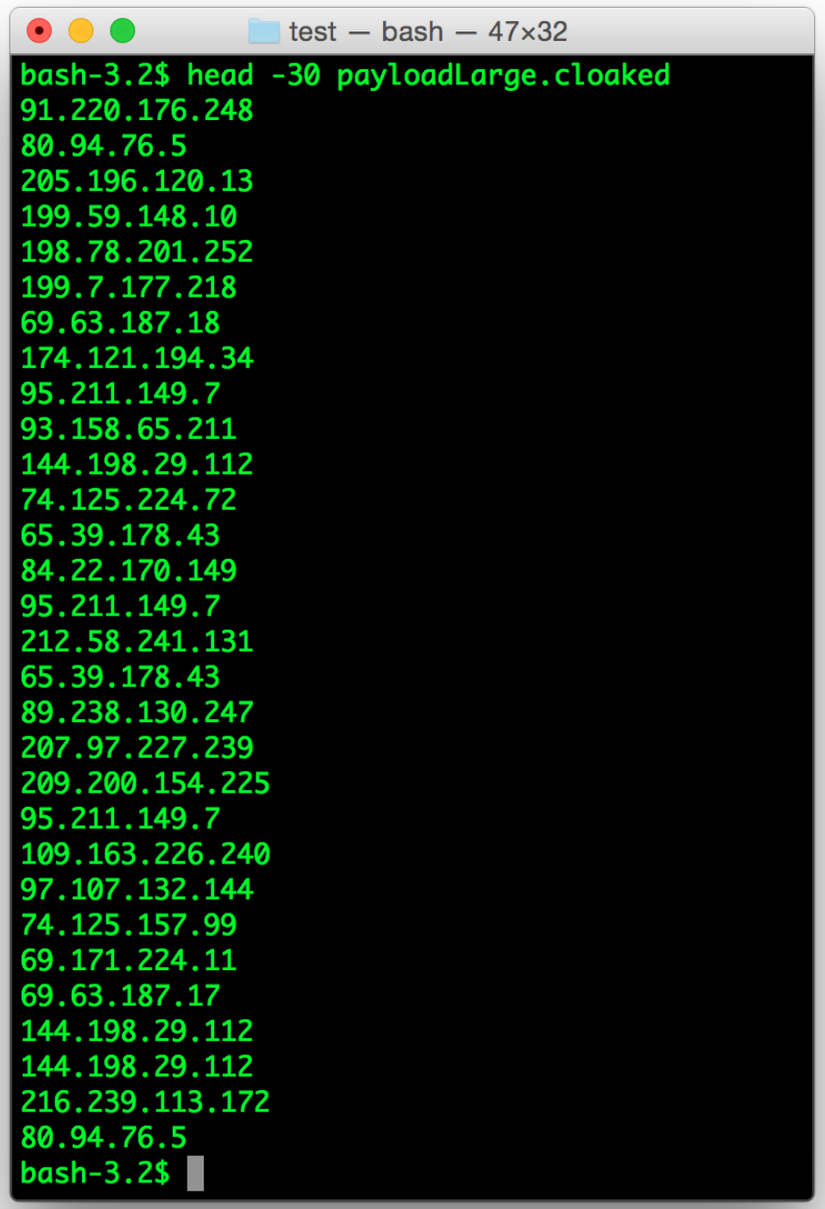
From there it's straightforward to brute-force the cipher



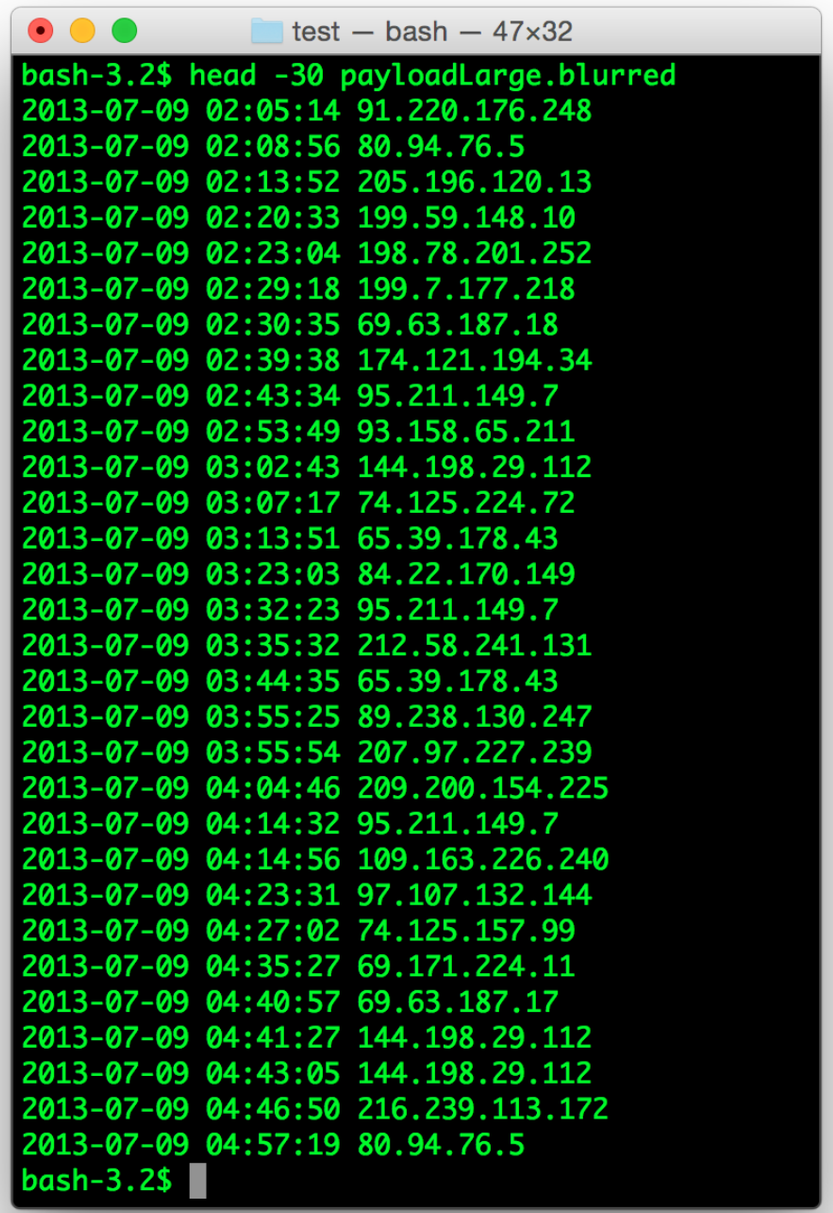
Exfiltrating Data From Secure Networks

Vulnerability - Frequency Analysis Attacks (cont.)

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



Prepend Randomized Sequential Timestamps

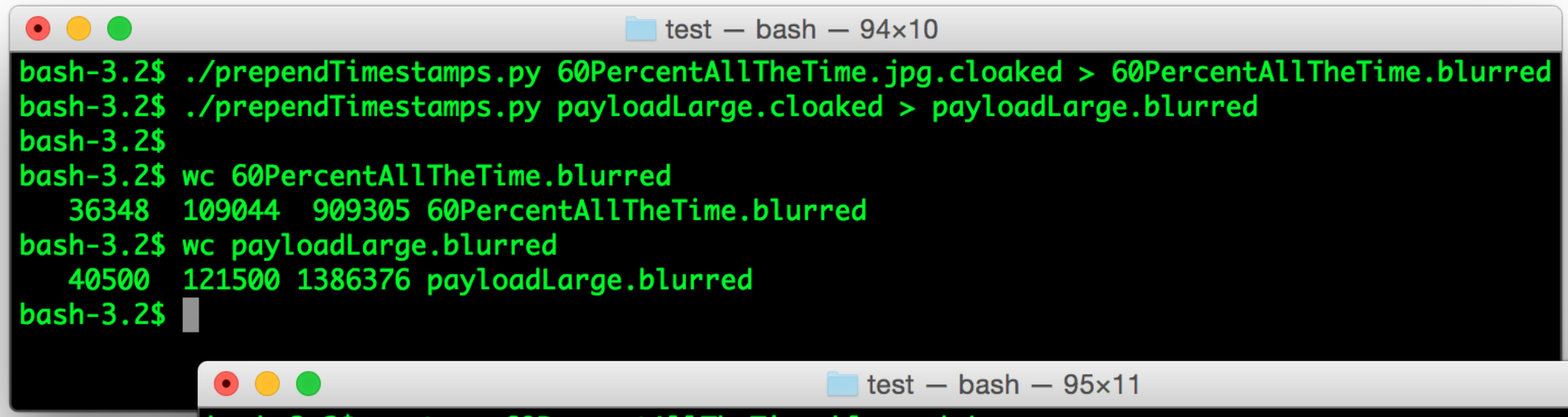


Exfiltrating Data From Secure Networks

Vulnerability - Frequency Analysis Attacks (cont.)

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

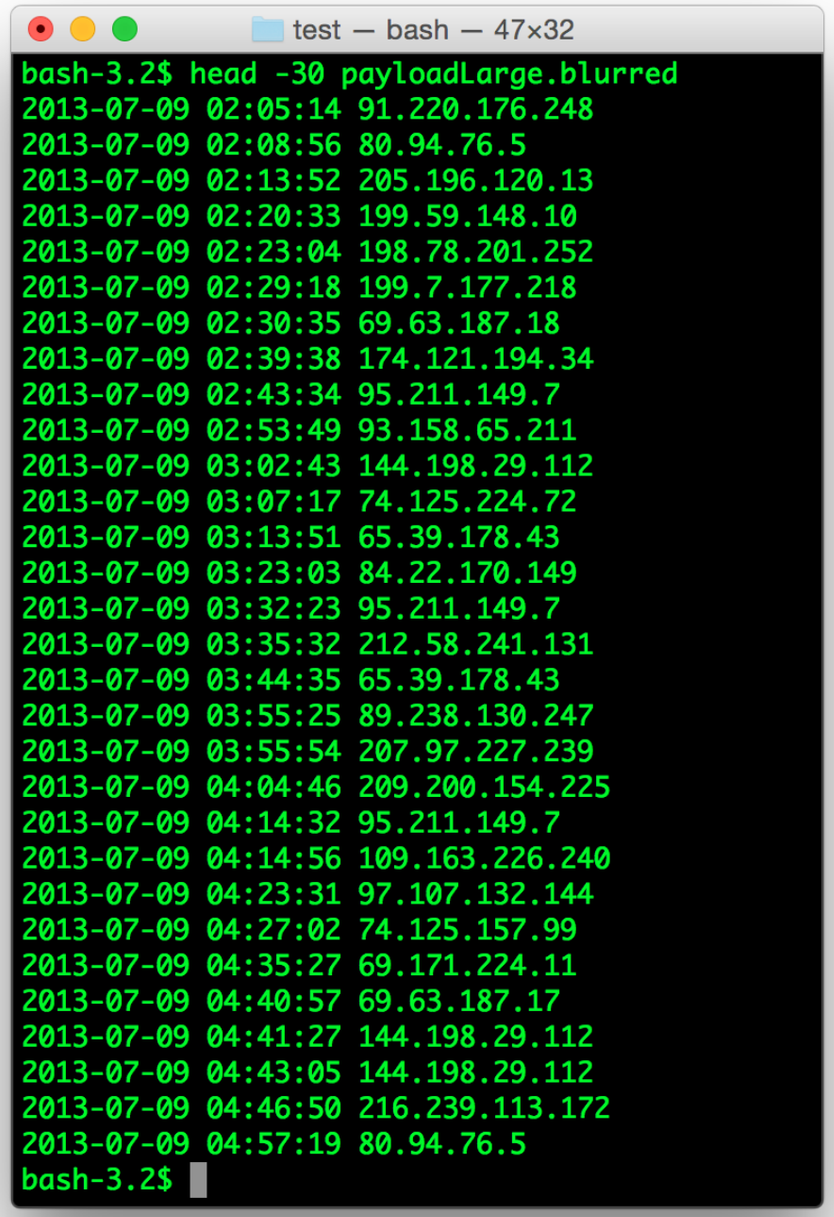
(Script rolls back to initial date when current date is reached - no time travel timestamps from the future regardless of file size)



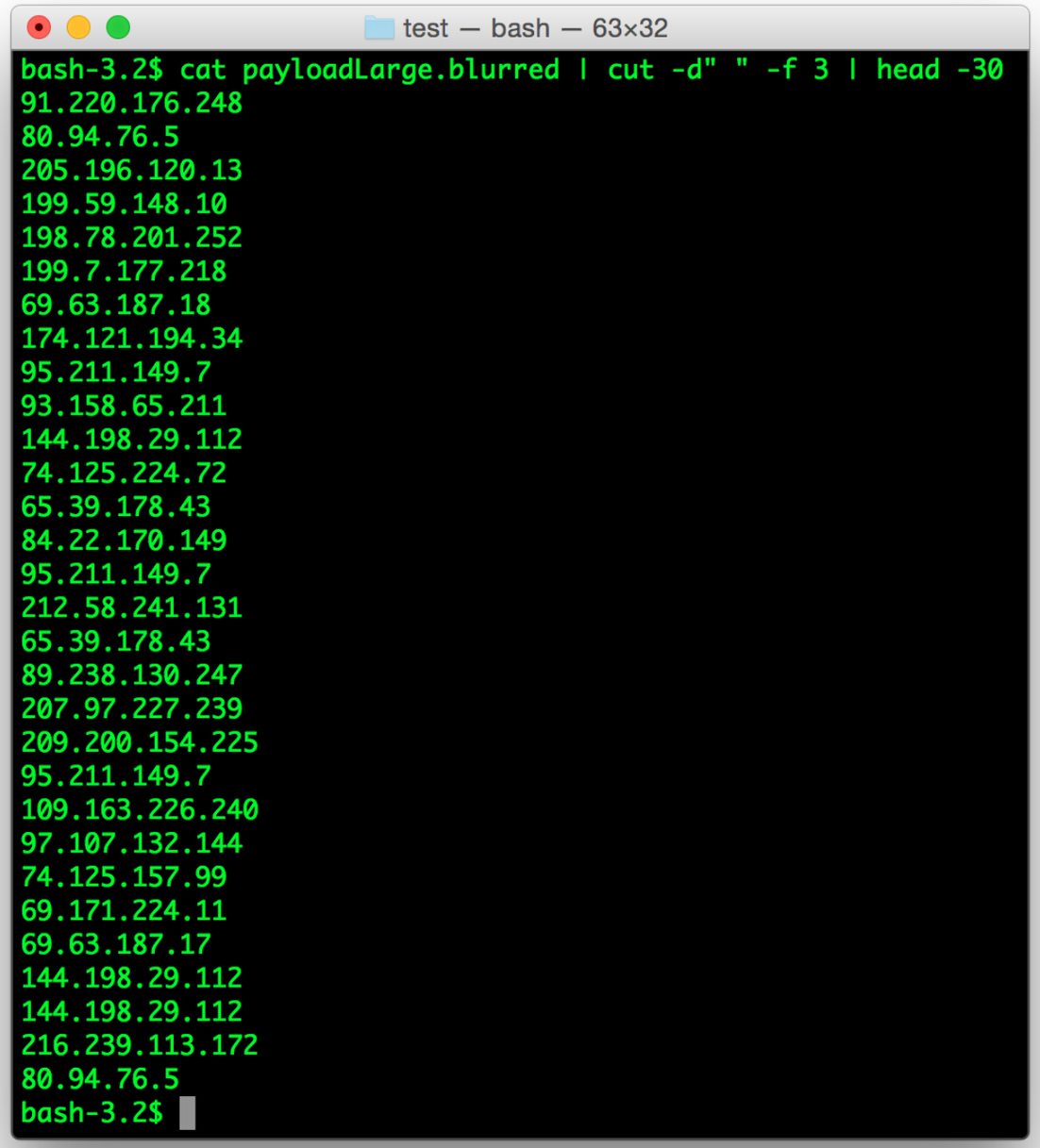
Exfiltrating Data From Secure Networks

Vulnerability - Frequency Analysis Attacks (cont.)

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



Strip Timestamps, Ready for Decloaking!



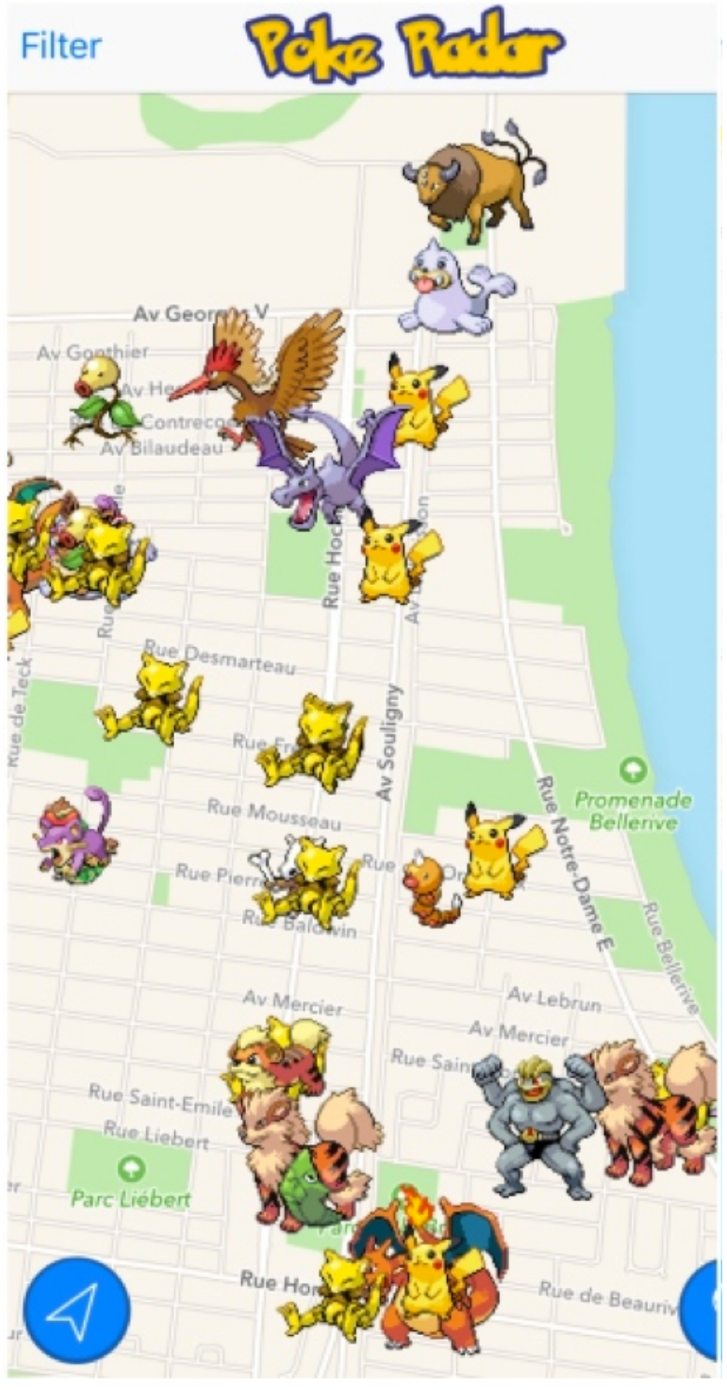
Exfiltrating Data From Secure Networks

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

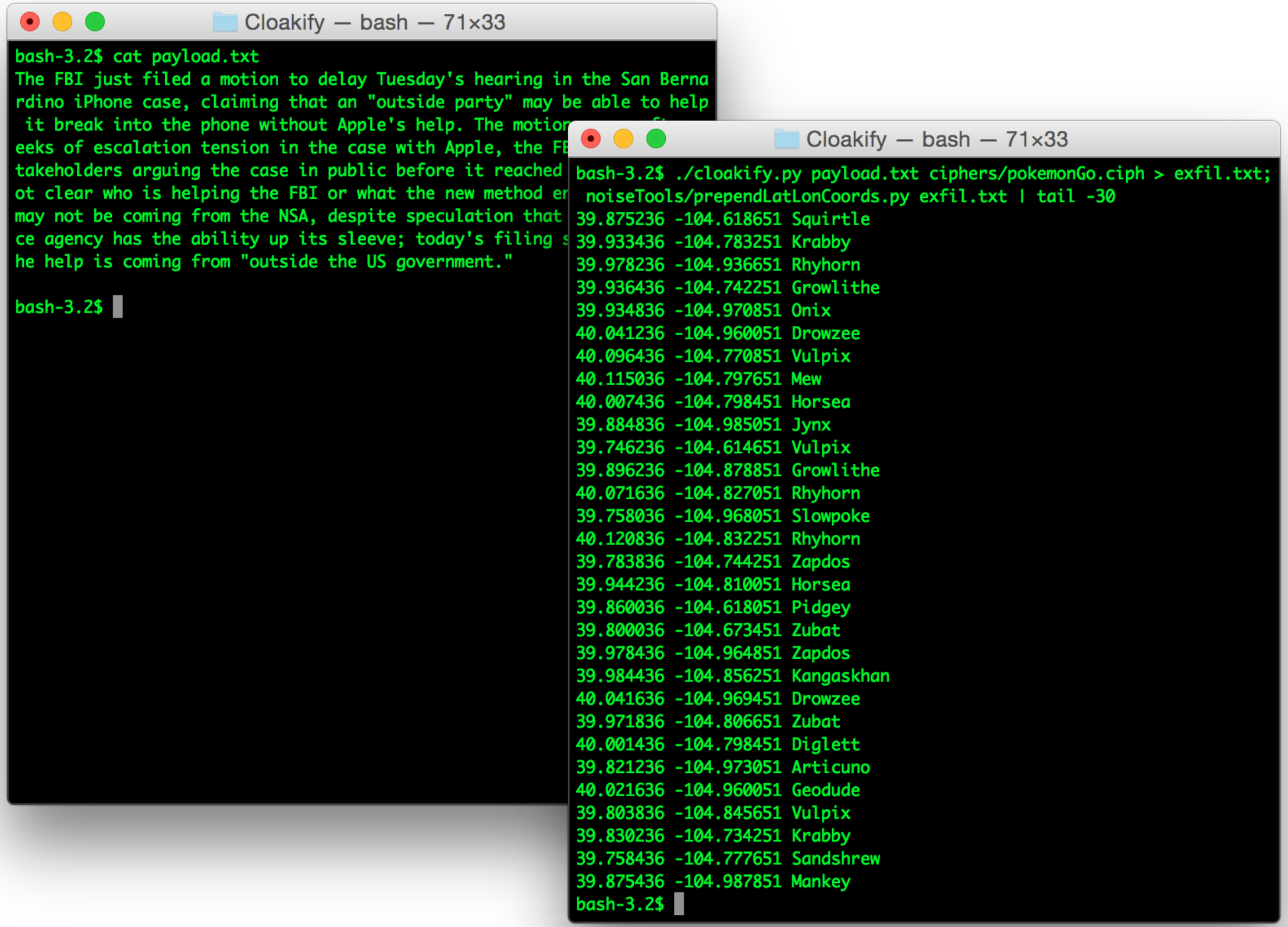


Poke Radar, a crowdsourced Pokemon location hotspot for Abras. Vancouver, on the other hanc (Poke Radar/CBC Screenshot)

Exfiltrating Data From Secure Networks

“soooo much exfil joy”

Exfiltrating Data From Secure Networks



Exfiltrating Data From Secure Networks

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

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 to tailor 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)