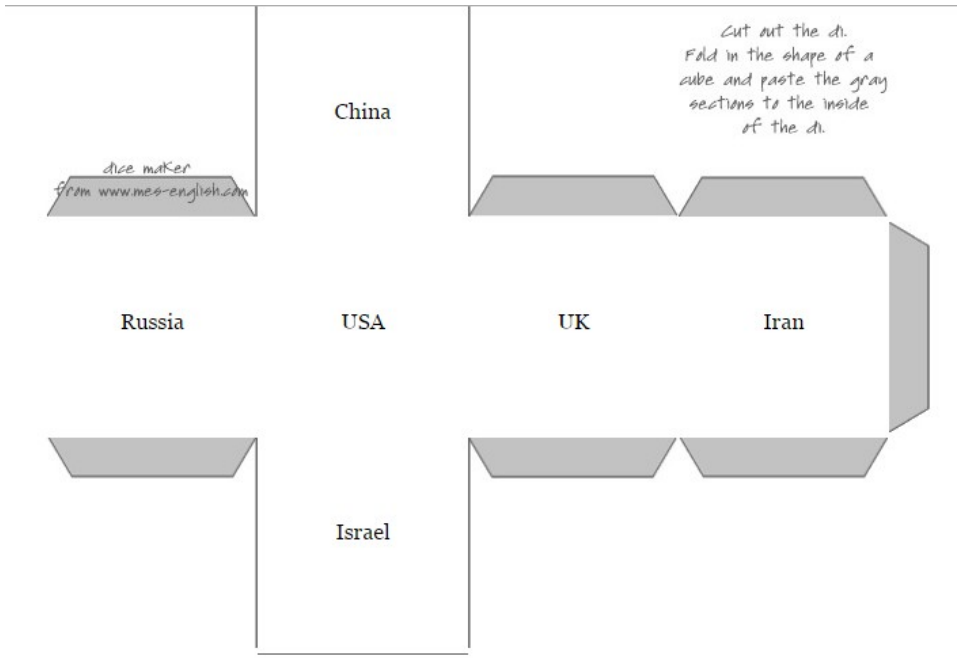
Threat Testing for the Good of the Internet

How hacking anti-virus and other security products makes everyone safer

SE Labs

whoami
Simon Edwards

- **SE Labs founder**
- AMTSO Director
- Ex-Dennis Technology Labs
- Developed the APT Attribution Generator (see above-right…) which inspired the Attribution dice (see below-right)



whoamit
SE Labs

- London-based security testing lab
- Experienced, comparatively large team
- Works with:
  - Global 500 enterprises
  - **Security service/ product vendors***
  - Security teams (e.g BT)
  - Analysts
- *Vendors: traditional and ‘next-gen’
- No-one knows what ‘SE’ stands for…

What we test

- Endpoint security software (detection/ protection/ response)
- Network appliances (security)
- Combined solutions (endpoint and appliance)
- Cloud security services (vs. on-prem)

Why do we test?

- Too much snake oil
- Bad enough before ‘next-gen’
- “AV is dead, AI will save the world!”
- **This stuff costs LOAD$!**

Intelligence-based testing

- Realism
  - NO VT TESTING!
  - Real-life attacks and close copying of techniques
  - Full attack chain
  - **Breach, not just compromise**
- Validation
  - Don’t believe the security products
  - **Forensic ‘incident response’ approach**
- Ethics
  - Reproducible (product improvement)
  - **Transparent (low-level data sharing, clear methodologies)**

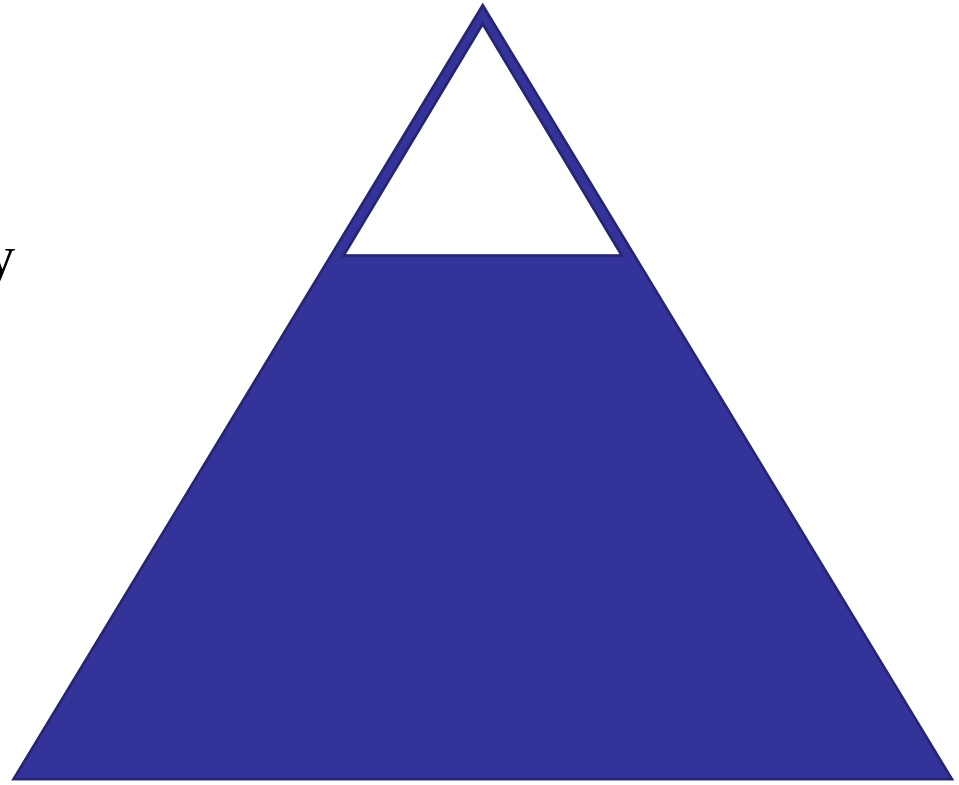
Different strokes for different folks

- Not all products work the same way
- Not all products do the same thing (or claim to – some next-gen)
- Testing needs to pay heed to these differences
- Millions of malware samples vs. series of well-known targeted attacks
- How products react to real attacks provides **valuable information for improvement**

Real threats for better tests

- Locate prevalent threats
- Don’t take feeds from vendors
- Expose products realistically
  - Social engineering (web, email)
  - Automatic attacks (web-based exploits) < a specialty
  - Targeted attacks

- What about APTs?
  - Threat intelligence exists
  - **It’s ‘just hacking’**
  - FireEye’s pyramid of relevance



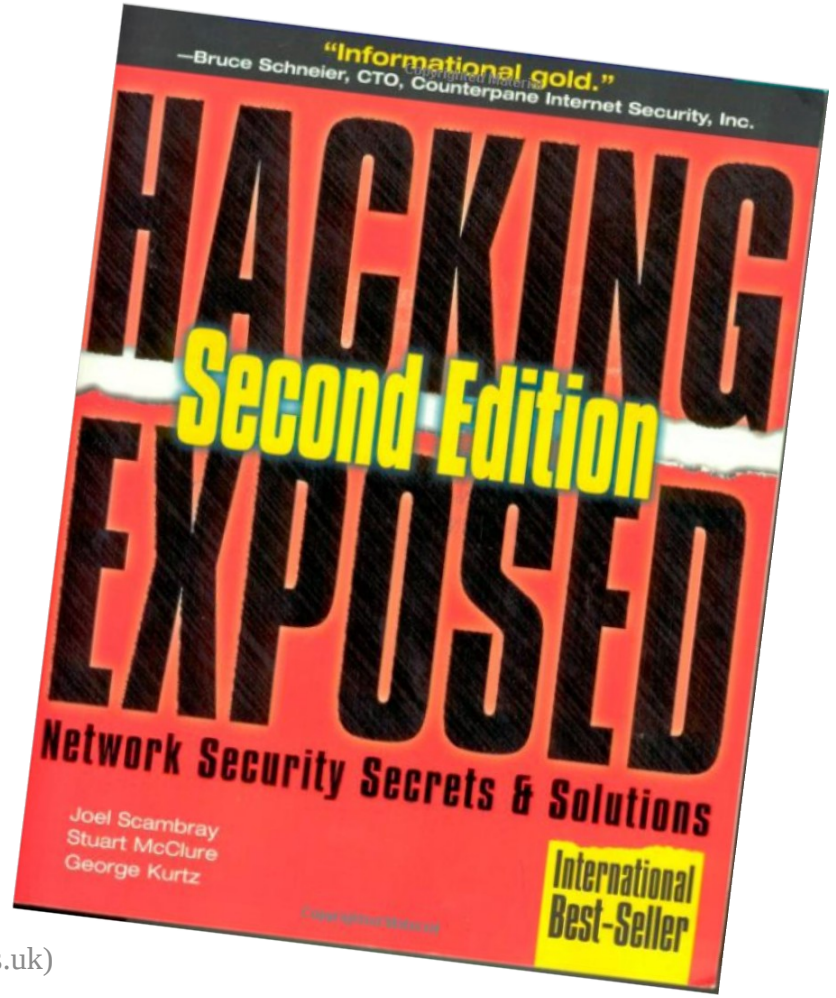
Can we behaving like an ‘APT’?

- Nation states/ criminals are **not equally well-resourced**
  - The Unbearable Lightness of APTing (https://www.virusbulletin.com/uploads/pdf/conference_slides/2015/Balmas-etal-VB2015.pdf) compares tactics used by US and middle-east actors
  - Mossad vs. Hamas
- Nation states have **incentives not to use zero days**
  - Scalable
  - Extra hard to attribute
  - Disposable ‘burner’ technology
- Nation state **targets often lack defences** (activists and their friends/ families)
- Breach Level Index 2016 **– 1% state sponsored**
- Verizon **– 0.4%** Cyber Espionage

Hacking Exposed vs. Hacking Team

- Hacking Exposed 2nd Edition – pub. **2000**
- Hacking Team – compromised **2015**
- Data leak published **2015**
- Phineas Phisher's 'methodology' pub. **2016**

- Compare and contrast his/her methods and those outlined in a 16 year-old manual. (Hint: virtually **identical**)



Zero to Neo

Zero resources

Skilled

Unlimited resources

Basic

Advanced

Why justify ‘APT’ testing

FireEye claimed:

"Any lab test is fundamentally unable to replicate the targeted, advanced attacks launched by sophisticated criminal networks and nation-states.

The best way to evaluate FireEye is for organizations to deploy our technology in their own environment and they will understand why we are the market leader in stopping advanced attacks.”

[
https://www.fireeye.com/blog/executive-perspective/2014/04/real-world-vs-lab-testing-the-fireeye-response-to-nss-labs-breach-detection-systems-report.html
]

Breach = process, not infection

- Products will miss the infection stage sometimes
- Products may not notice post-infection activities – but they might!
- A breach is a **combination of attack stages**
- Many tests stop after the malware is introduced
- A thorough test will make no assumptions about a product's capabilities
  - Test like a real attacker and see what happens
  - Take no short-cuts (e.g. introduce malware realistically, such as via email)
  - Use realistic configurations (seek advice)

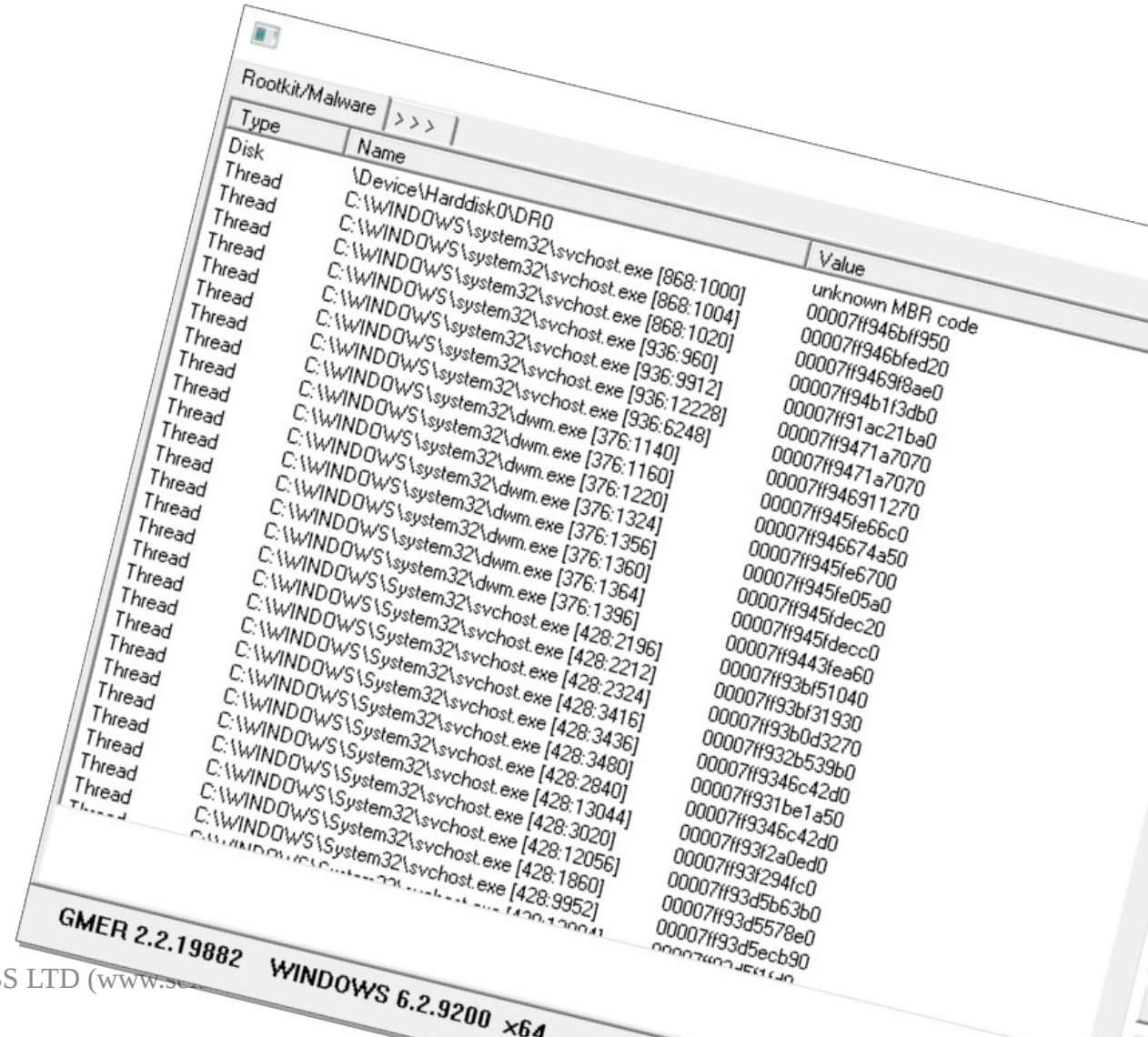
Spot the dog

Testing challenges with evasive malware

- **Rootkits hide** (O RLY?!)
- Mainly about validation of installation and removal…
- How can you tell if anti-malware blocked/ removed a rootkit?
- How can you validate that the rootkit installation succeeded (when pre-infecting systems?)
- What if you want to test specialised anti-rootkit tools?
- What kind of evidence will satisfy challengers of the results?
- Have the seen the size of a modern memory dump?

Using anti-malware to validate

- Just, no.
- **It's never been an acceptable approach** to:
  - Use anti-malware tools to validate malware
    - E.g. "File.exe is malware because Kaspersky/Symantec/Microsoft says so."
  - Use anti-malware tools to validate other tools' results
    - E.g. "System was cleaned because GMER says so."

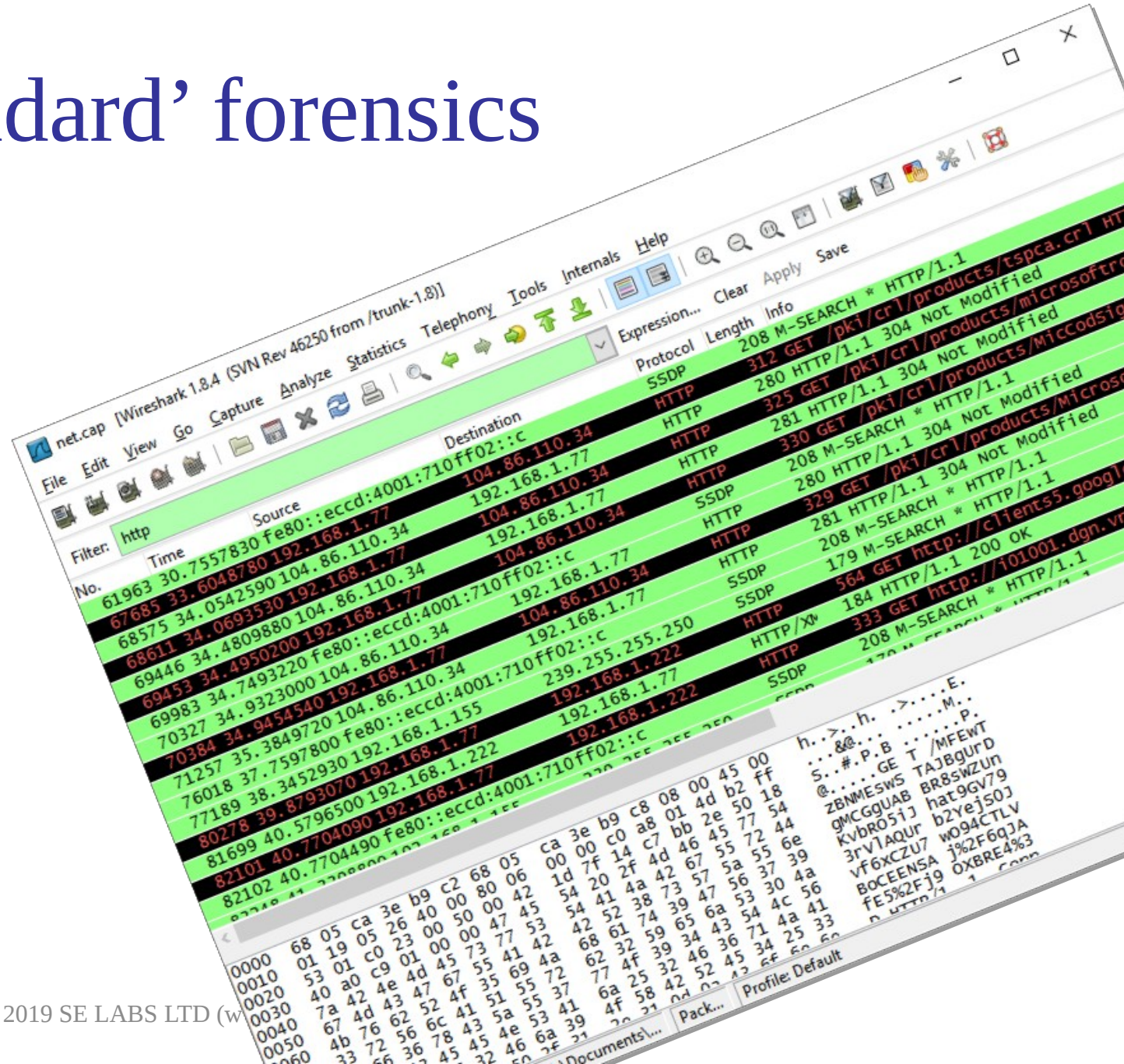


Limitations of ‘standard’ forensics

- Assume the worst – lowest level (although maybe not firmware)

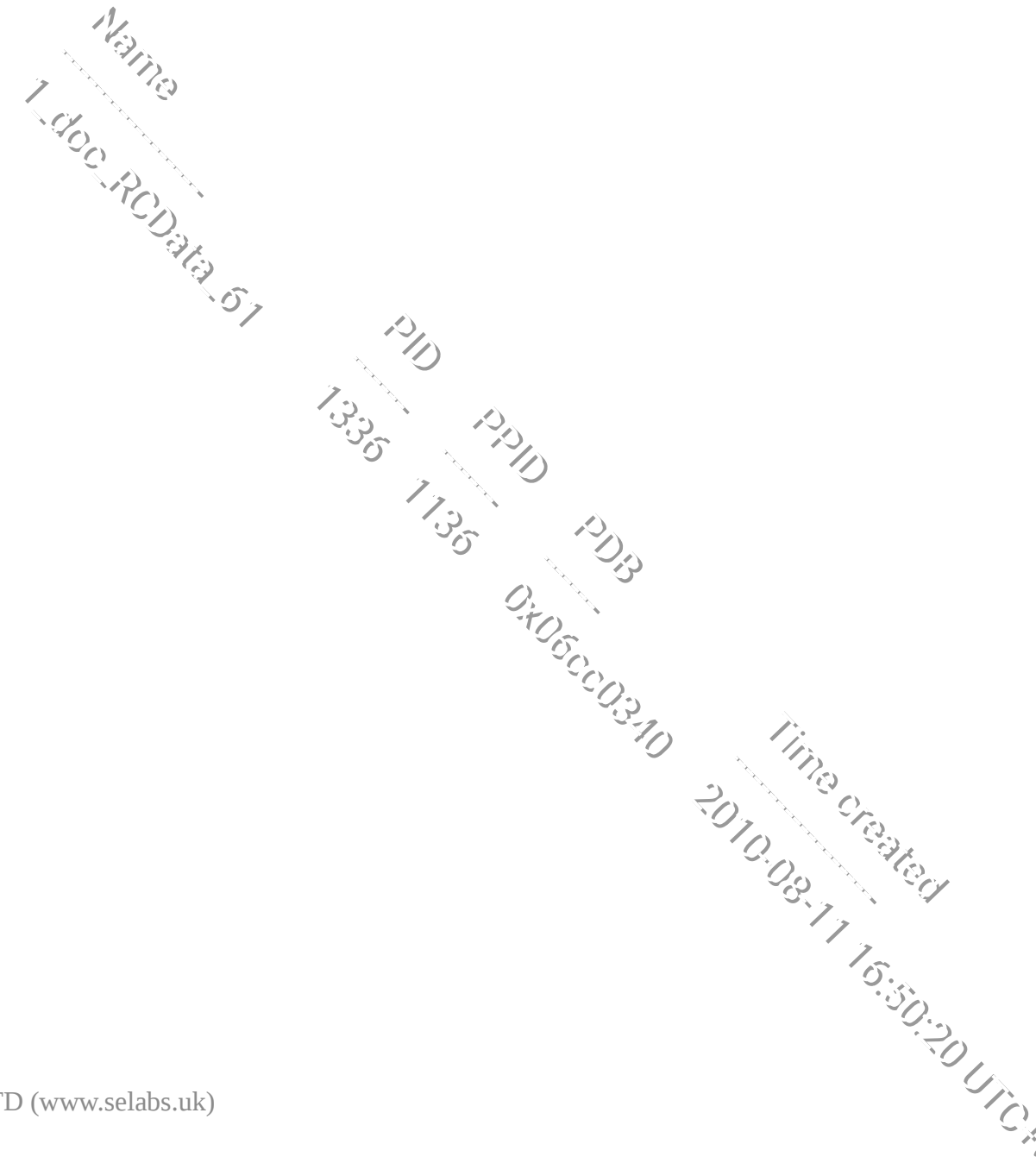
- File system changes
- Registry changes
- Process Monitor files
- PCAP files

- Prefetch files < not always reliable
- MBR changes < this one can work **offline**



Main tasks

- Does the infection work on a clean system? (**Is the sample viable?**)
- What does a successful infection look like?
- Did the security product prevent the infection? Or…
- Did the removal tool clean the system (and to what extent)?



Validating rootkit infection/ removal

- The very expensive way
- The cheap (**free**), scalable way(s)
- Offline memory analysis is useful/ essential

1. Acquire memory
2. Analyse using tools
3. Spot rootkit (validating installation)
4. Determine rootkit missing (validating removal)

Expensive memory dumping

WindowsSCOPE CaptureGUARD Physical Memory Acquisition Hardware – PCIe Add-on

$9,599 ea. (March 2019)

http://www.windowsscope.com/product/captureguard-physical-memory-acquisition-hardware-pcie-add-on/

Free memory dumping

- **DumpIt**
  (Was by MoonSols, now Comae Technologies)
  - Direct download: http://www.moonsols.com/wp-content/plugins/download-monitor/download.php?id=7
  - 'Legit' download: https://comae.typeform.com/to/XIvMa7 (annoying survey > email link to download)

- Mdd
  (Apparently from ManTech, but good luck finding the link on the corporate site…)
  - Download: https://sourceforge.net/projects/mdd/

- Why >1? Sometimes one will crash on infected systems.

Analysis: Malware infection (not rootkit)

From a recent SE Labs test, in which the result of ‘compromised’ was disputed by the vendor…

- File system changes:
  - + C:\Users\x\AppData\Local\Temp**server.exe**
- Registry changes:
  - + HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion**Run**\x: ""C:\Users\x\AppData\Local\Temp**server.exe**" ..“
  - + HKLM\SYSTEM\CurrentControlSet\services\SharedAccess\Parameters\FirewallPolicy**FirewallRules**\x: "v2.10|Action=**Allow**|Active=TRUE|**Dir=In**|Protocol=17|Profile=Public|App=C:\Users\x\AppData\Local\Temp**server.exe**|Name=server.exe|

Summary: ‘Regular’ infection

```
si@SEL:~/$ volatility -f dump.raw --profile=Win7SP1x64 pslist

Name                 PID  PPID   Thds     Hnds   Sess  Wow64 Start
-------------------- ---- ------ -------- ------ ----- ------ ----------
System               4    0      112      504    ------ 0      2017-01-18 10:04:15
smss.exe             288  4      2        33     ------ 0      2017-01-18 10:04:15
svchost.exe          716  540    12       369    0      0      2017-01-18 10:04:20
Agent.exe            780  540    42       978    0      0      2017-01-18 10:04:21
taskhost.exe         1500 540    11       197    1      0      2017-01-18 10:05:45
dwm.exe              1544 796    6        126    1      0      2017-01-18 10:05:45
explorer.exe         1552 1536   35       820    1      0      2017-01-18 10:05:45
AgentUI.exe          2256 1552   17       379    1      0      2017-01-18 10:05:47
server.exe           2688 2152   11       218    1      1      2017-01-18 10:05:48
Dumpit.exe           3372 1552   6        53     1      1      2017-01-18 10:07:06
```

Memory analysis for visual proof

Direct Kernel Object Mode (DKOM)

- Common technique: unlink a process' entry from the doubly-linked list
- Malicious process won't appear in the process list (pslist)
- Run psscan and compare outputs
- Entries in psscan output that are **missing** from pslist are suspect

- Bit onerous, though…

psxview FTW!

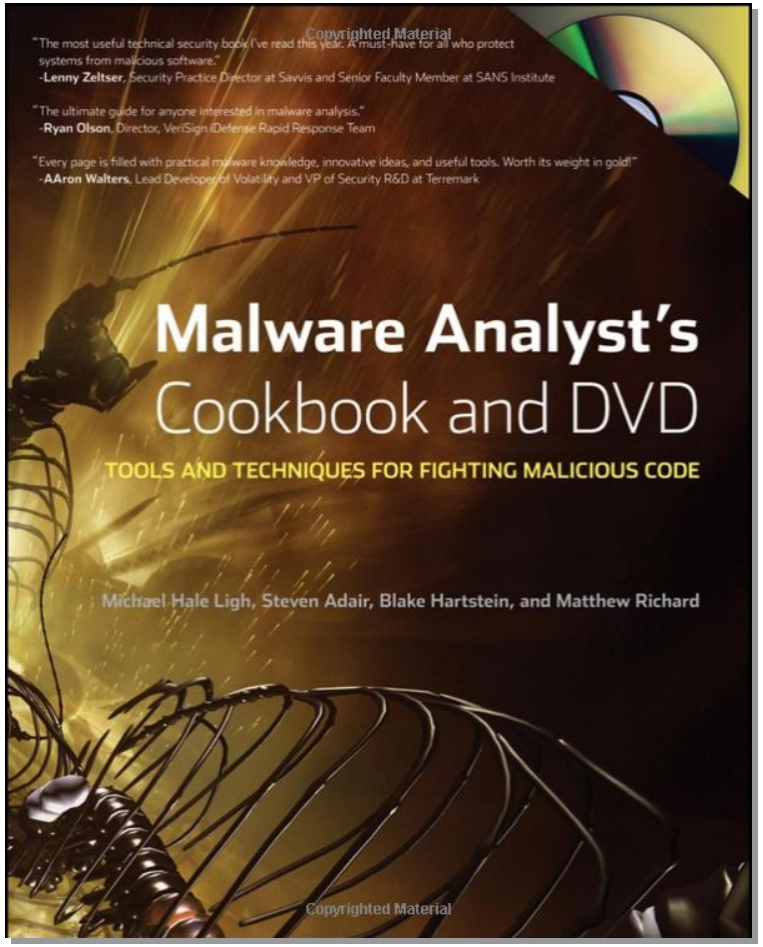
Name	pslist	psscan	thrdproc	pspcid	csrss	session	deskthrd
svchost.exe	True	True	True	True	True	True	True
1_doc_RCData_61	False	True	True	True	True	True	True
explorer.exe	True	True	True	True	True	True	True
winlogon.exe	True	True	True	True	True	True	True
svchost.exe	True	True	True	True	True	True	True
msiexec.exe	False	True	False	False	False	False	False
rundll32.exe	False	True	False	False	False	False	False

Stuxnet: A series of unfortunate events

- Stuxnet is a rootkit for industrial systems
- First identified (publicly) in 2010
- Supposedly to sabotage Iran's nuclear programme (and make it look like an accident)
- Windows XP was popular at the time

Analysing Stuxnet

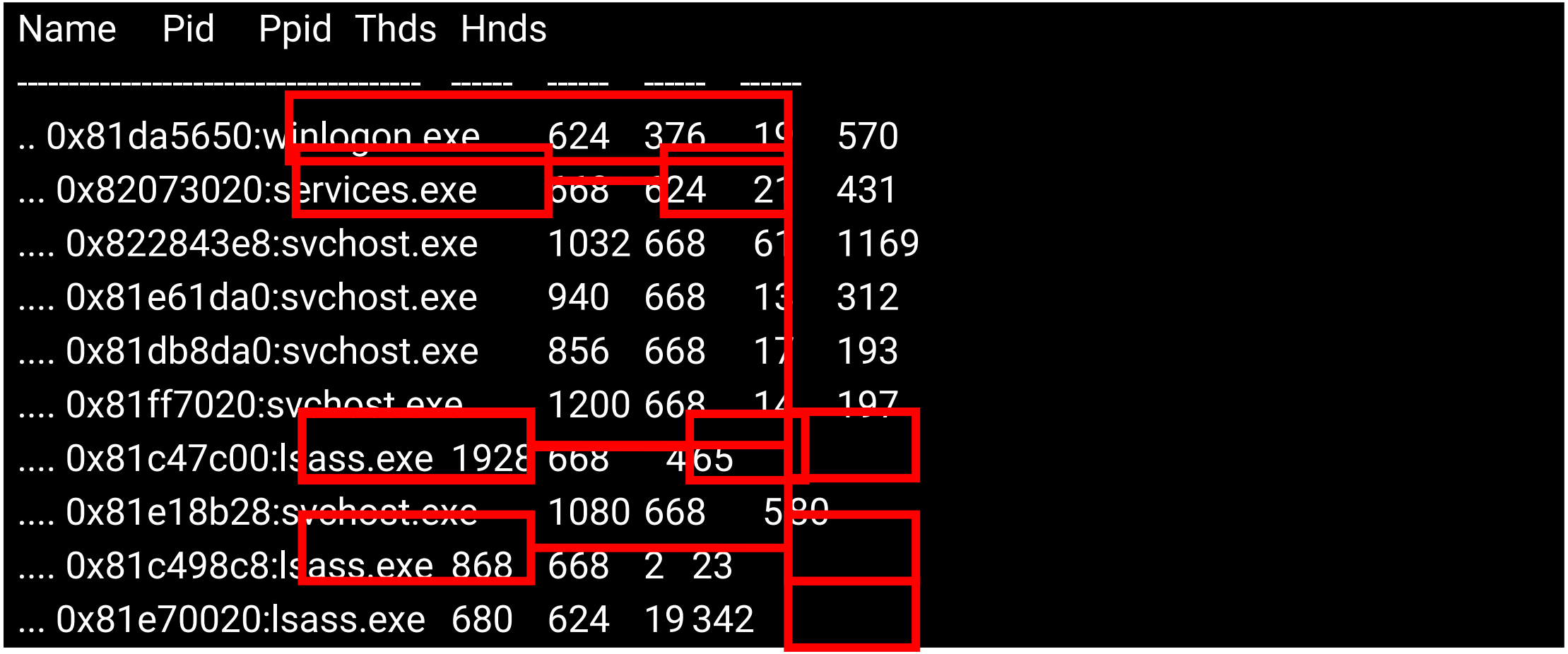
- Infected memory dumps are available
- Malware Analyst's Cookbook and DVD
  - Good introduction to Volatility (and loads of other good stuff)
  - Website used to link to the sample (but the link is dead now)
  - Google for "stuxnet.vmem.zip" (or ask me later!)



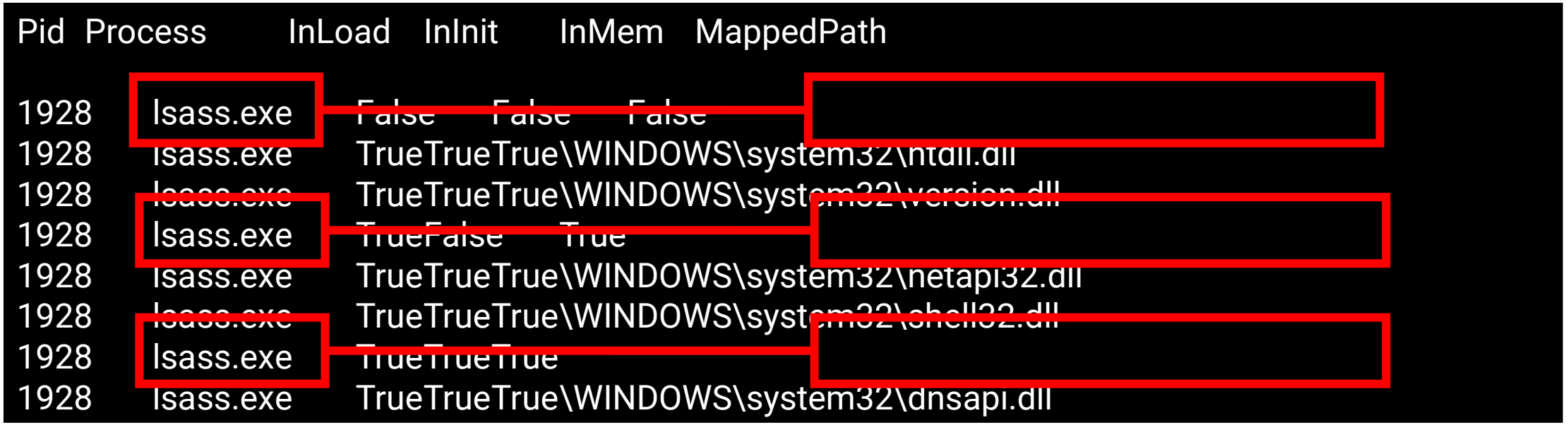
volatility -f stuxnet.vmem --profile=WinXPSP3x86 pstree

How many lsass.exe processes?!

Stuxnet processes



Unlinked DLLs



Next steps

- Dump processes (procexedump; procmemdump)
- Compare with legitimate lsass.exe
- Example:

!This program cannot be run in DOS mode.
Rich
.text
`.data
.rsrc
ADVAPI32.dll
KERNEL32.dll
NTDLL.DLL
LSASRV.dll
SAMSRV.dll

!This program cannot be run in DOS mode.
Rich
.verif
.text
.bin
.reloc
ZwMapViewOfSection
ZwCreateSection
ZwOpenFile
ZwClose
ZwQueryAttributesFile
ZwQuerySection

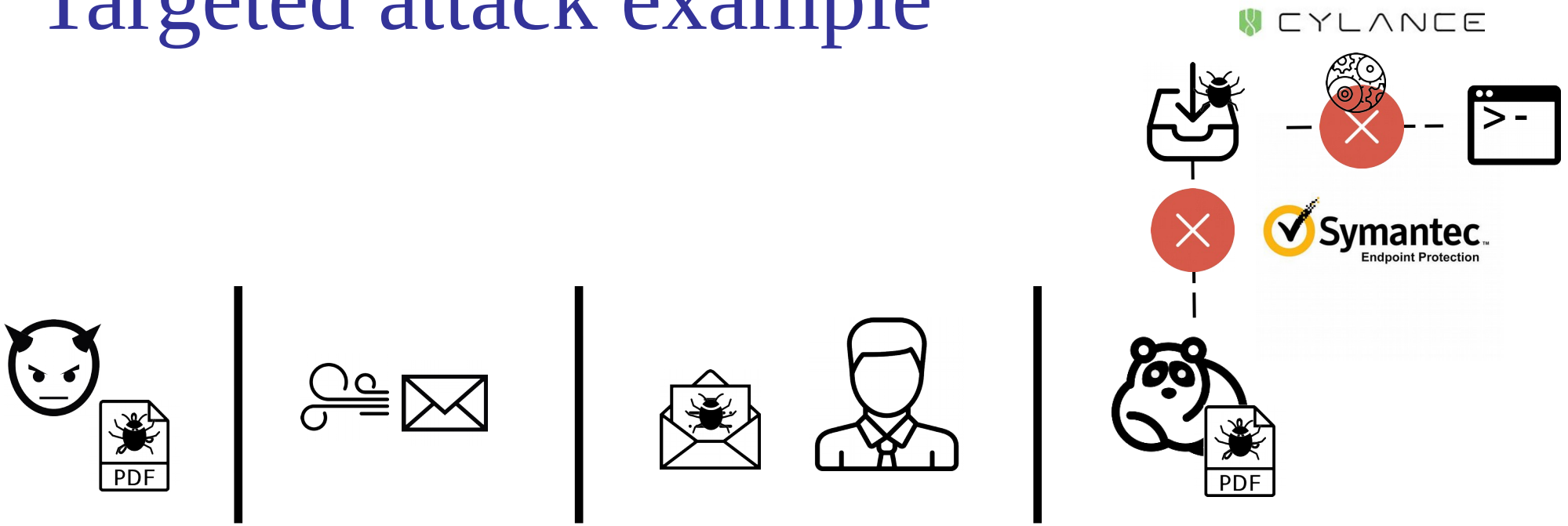
Code injection/ DLL hiding

- Stuxnet uses three types of code injection and DLL hiding
- We just saw an example of process hollowing
  - There is a brief description of this method in the WikiLeaks ‘Vault 7: CIA Hacking Tools Revealed’ documentation
  - General link: https://wikileaks.org/ciav7p1/
  - Process Hollowing: https://wikileaks.org/ciav7p1/cms/page_3375167.html

Important tips

- Use these techniques in all tests, not just anti-rootkit tests
- Hygiene is important. Clean your MBR between test cases.
- Have more than one memory dumping tool to hand.
- Analyse offline.
- For a reasonable test, have a lot of storage for memory dumps.
- Share output of analysis, not full memory dumps (KBs vs. GBs).

Targeted attack example



Demo

- Fully-updated Windows 10 PC, Windows Defender, UAC enabled
- Four year-old PDF exploit
- Default Metasploit installation (and no other tools)
- Minimal social engineering
- Full 0wnership in < 7 minutes
- What does anti-malware see?
- What does ‘next-gen’ see?
- EXAMPLE THREAT WAS NOT CHERRY PICKED FOR DRAMA!

Anti-virus vs. next-gen detection

Show All | EOP | Injection | Password Dump | Download-and-Run

Binary Summ

Charac
Commo
Detect

Recent processes

Hostname			
DESKTOP-P602			

MD5 **005bd0de5bc936de41a5ba632f3d6116**

(Click here to Download)

File Type	Executable
File Size	383,490
Characteristics	Available for download, Not signed
Common Names	pijljkx.exe, dlnuvec.exe, dollar_time.exe
Detected by AV	0 engines detect this file as malicious

DESKTOP-P6021OS	9/30/16 8:34:56pm	pijljkx.exe (PID 5688) opened process memory of lsass.exe (PID 804). This could be an exploit dumping credentials from memory.	Detail [Hide]

Next steps?

Remember: Breach = process, not infection

- Increase tools used?
  - How much noise is needed to trigger detection?
- Increase realism/ scope?
  - Perform initial reconnaissance (could be detected)
  - Pivot to other systems/ services (e.g. AD, Dropbox)
  - Attempt persistence? (not always needed/ realistic)
  - C2 hosting
- Scoring method?
  - Time to detect
  - Detect vs. detect + protect
  - What about products designed only to detect? Penalise? (hint: no!)
  - What if data cannot be exfiltrated?

Reasons to run full breach testing

- Testing can indicate:
  - How useful are established and new security solutions?
  - Where are the **strengths and limits** of their capabilities?
  - Do they do what they claim?
  - Do they have **other benefits**?
- The above information can help businesses consider:
  - Are they good **value for money**?
  - How much **training** will staff need to use them effectively?
  - How much **overlap** is there with currently deployed measures?

Without good, thorough test results…

Constructed by Yonathan Klijnsma, Fox-IT

Questions?

@SELabsUK

@SPGEdwards