CATCH OF THE DAY

A Close Look at a Daily Dataset of Malware Samples

Xabier Ugarte-Pedrero and Mariano Graziano

CARO 2019, Copenhagen

Talos
Cisco Security Research

$whoami

@emd3l

@xabiugarte

Cisco Security Research

Malware Research Team

- Malware analysis
  - Quick analysis (extraction of indicators, coverage)
  - In-depth reversing (manual)

- Automation
  - Signature generation (Bass)
  - Automated analysis tools (FIRST, Pyrebox, ROPMEMU)
  - Clustering

Sharing is caring

- What do we share daily?
- What do we buy/exchange?
- What are the challenges?
- How useful is it?

Clarification

- This presentation describes an academic paper developed in collaboration with Eurecom (France) [1]
- This research was started on the beginning of 2016
- Queries and sample processing were spread through several months by borrowing internal company resources

The dataset and our results should be representative and hold also after 3 years

[1] http://s3.eurecom.fr/docs/tops19_dailymalware.pdf

Catch of the day

Everyday security companies collect millions of samples

Catch of the day

Everyday security companies collect millions of samples

Catch of the day

Everyday security companies collect millions of samples

CERTs

Virustotal

Team Cymru

Universities

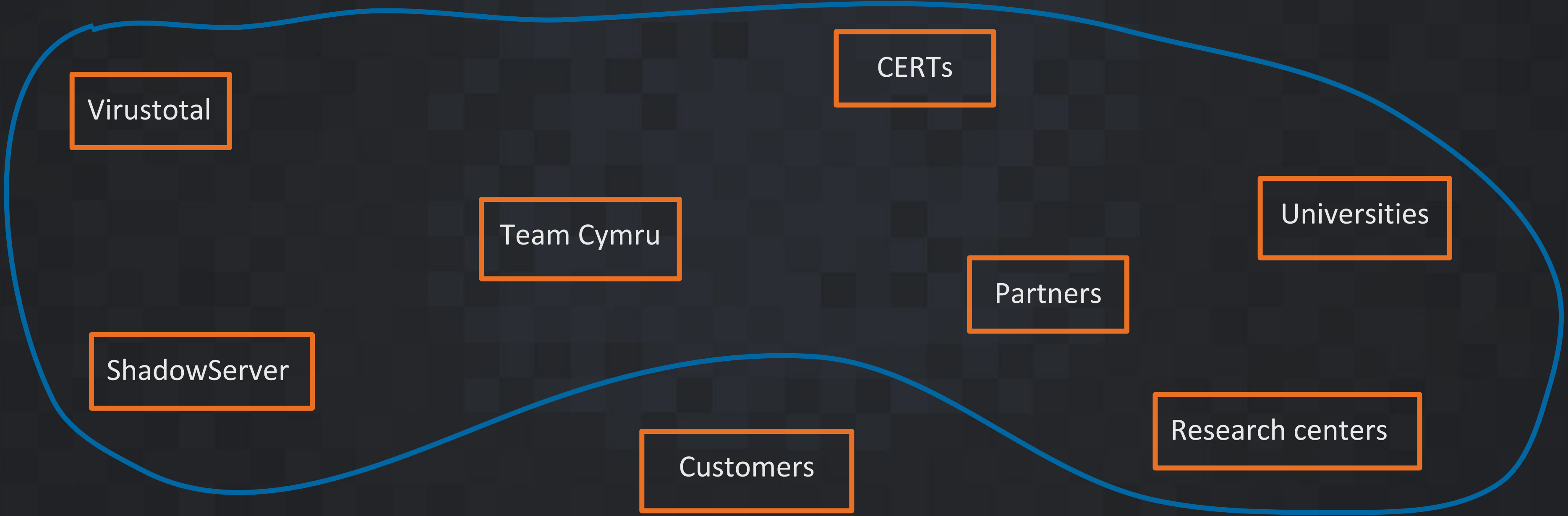
ShadowServer

Research centers

Customers

Catch of the day

Everyday security companies collect millions of samples



Catch of the day

Everyday security companies collect millions of samples

17 different feeds

Open questions

Open questions

- What the dataset contains?
- How many samples belong to known families?
- How much effort to analyze the remaining samples?
- How effective are the state-of-the-art techniques?

but most importantly:

- How much effort would it take?
- How many people? How many VMs? Cores?
- How many resources are wasted?
- What are the challenges?

Find a good day

Find a good day

Day of Week	Total (1 year)	Avg PEs
Monday	39,799,691	298,859
Tuesday	41,374,785	304,719
Wednesday	45,829,468	**344,031**
Thursday	44,725,851	338,893
Friday	43,244,266	324,400
Saturday	40,898,046	327,448
Sunday	38,459,952	320,919

Find a good day

Day of Week	Total (1 year)	Avg PEs
Monday	39,799,691	298,859
Tuesday	41,374,785	304,719
Wednesday	45,829,468	**344,031**
Thursday	44,725,851	338,893
Friday	43,244,266	324,400
Saturday	40,898,046	327,448
Sunday	38,459,952	320,919

+8,35%

Find a good day

Day of Week	Total (1 year)	Avg PEs
Monday	39,799,691	298,859
Tuesday	41,374,785	304,719
Wednesday	45,829,468	**344,031**
Thursday	44,725,851	338,893
Friday	43,244,266	324,400
Saturday	40,898,046	327,448
Sunday	38,459,952	320,919

Most prolific day!

Find a good day

Week	Nb. of samples	Week	Nb. of samples	Week	Nb. of samples
40	4,487,907	41	8,001,208	42	7,561,698
43	7,324,254	44	8,054,180	45	7,584,566
46	7,786,035	47	8,674,714	48	6,145,345
49	6,398,709	50	4,749,192	51	4,874,549
52	5,057,094	53	2,118,189		

Find a good day

Week	Nb. of samples	Week	Nb. of samples	Week	Nb. of samples
40	4,487,907	41	8,001,208	42	7,561,698
43	7,324,254	44	8,054,180	45	7,584,566
46	7,786,035	47	8,674,714	48	6,145,345
49	6,398,709	50	4,749,192	51	4,874,549
52	5,057,094	53	2,118,189		

Worst case:
week 47 from 16
to 22 November

Find a good day

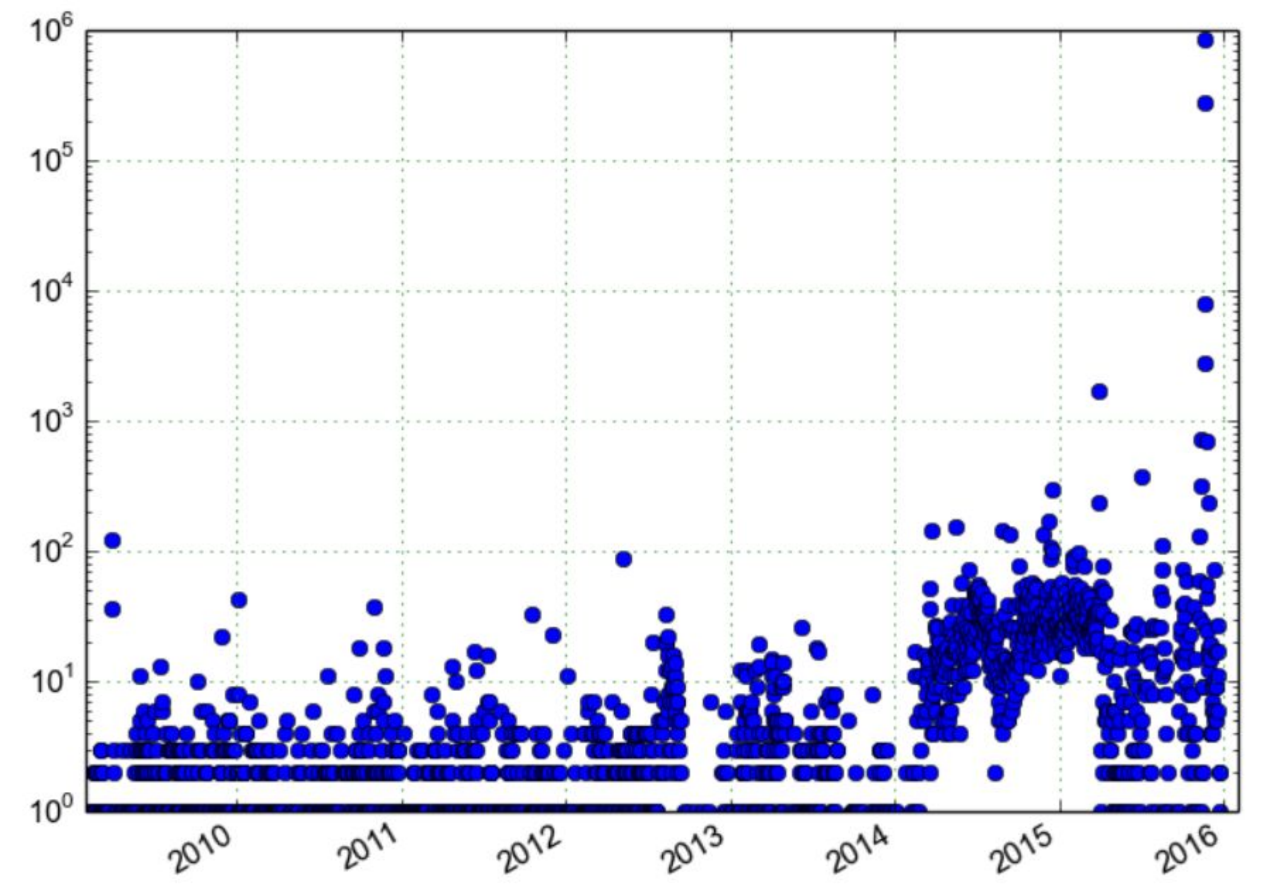
Day: Wednesday, November 18 2015

Number of samples: 1,261,882

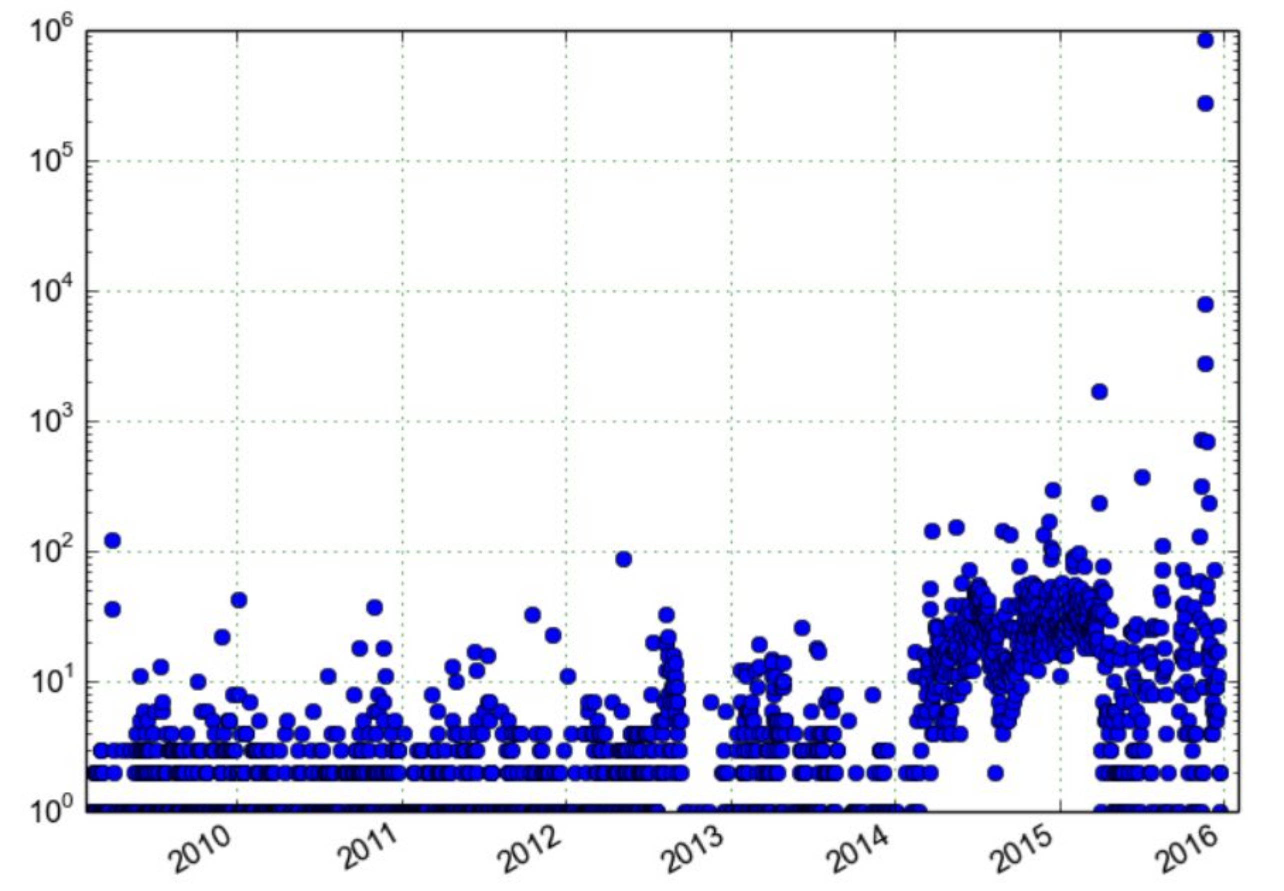
Worst case:
the highest possible number of samples

First look in VT

First look in VT

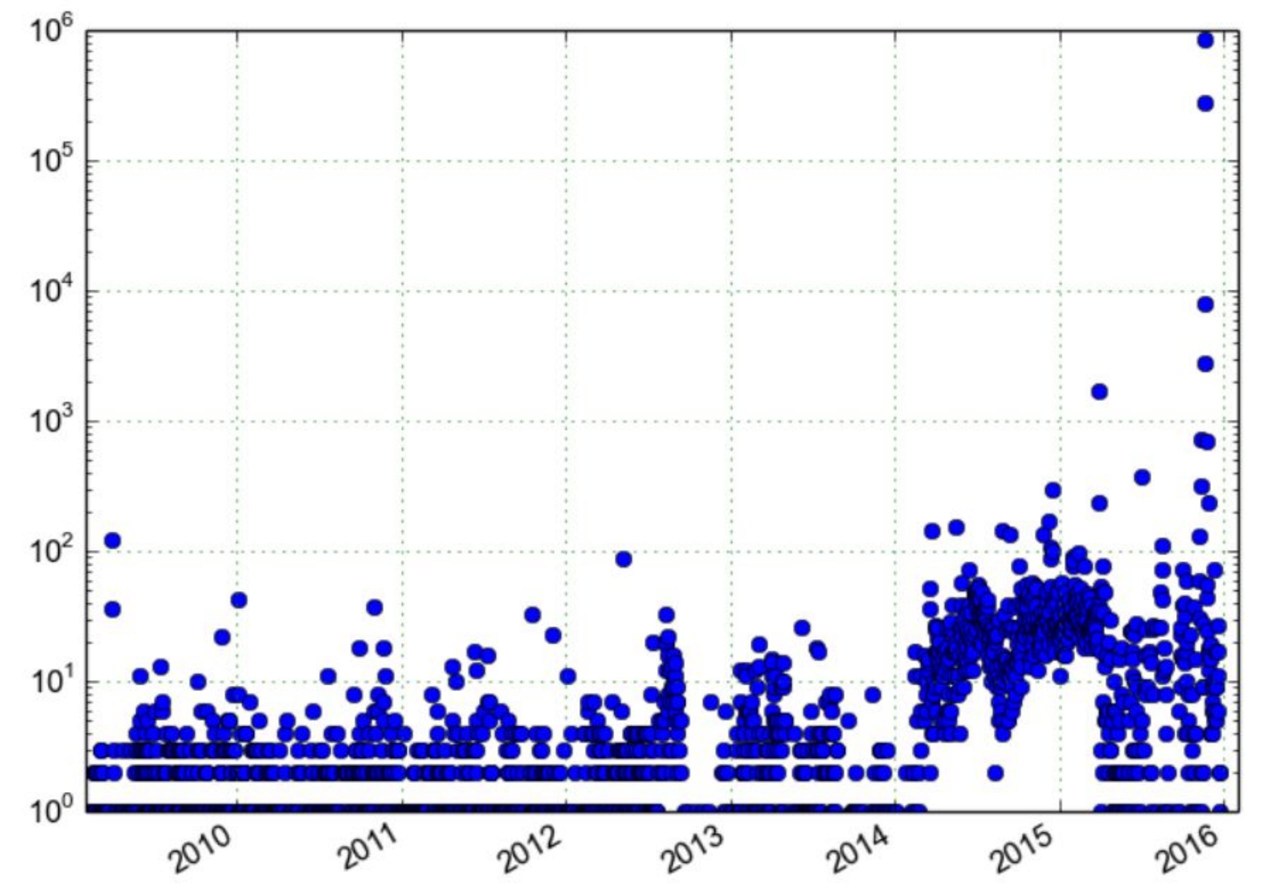


First look in VT



90% in VT

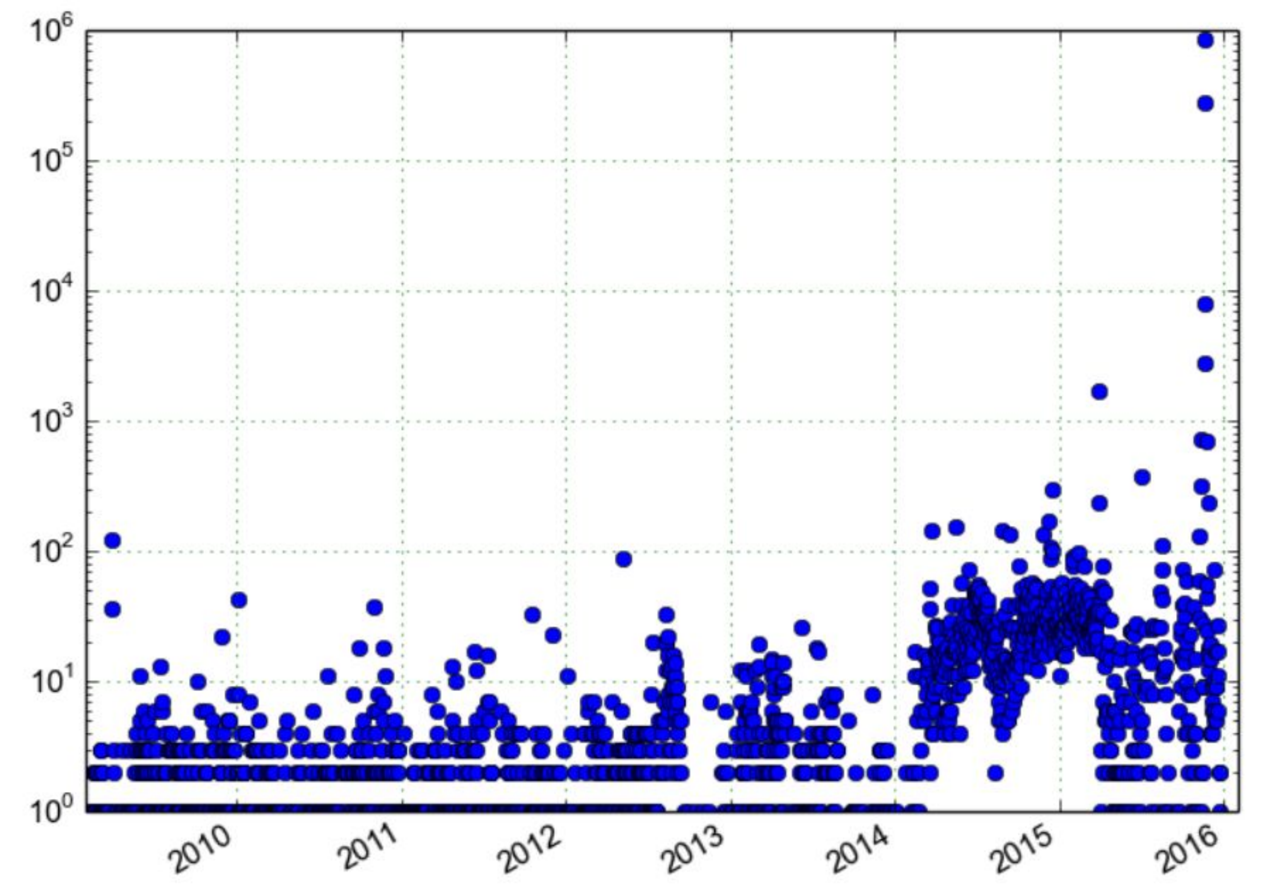
First look in VT



90% in VT

89% same/before day

First look in VT



90% in VT

89% same/before day

1,6% known before

Dataset

Dataset

160k Win32 EXE

440k Win32 DLLs

55k Android

4,3k Mach-O

4,5k ELF

Dataset

- 160k Win32 EXE
- 440k Win32 DLLs
- 55k Android
- 2,5k OOXML Word
- 4,3k Mach-O
- 47k PDF
- 4,5k ELF
- 5k MS Excel
- 904 Hangul
- 18k MS Word

Dataset

160k Win32 EXE

53k HTML

440k Win32 DLLs

1,9k PNG

55k Android

34k MP3

2,5k OOXML Word

4,3k Mach-O

47k PDF

4,5k ELF

5k MS Excel

16,8k RAR

9,4k JPG

25,9k ZIP

904 Hangul

18k MS Word

Dataset

- 2 Symbian
- 160k Win32 EXE
- 62 FF extensions
- 440k Win32 DLLs
- 64 Chrome extensions
- 53k HTML
- 1,9k PNG
- 55k Android
- 34k MP3
- 2,5k OOXML Word
- 4,3k Mach-O
- 47k PDF
- 4,5k ELF
- 5k MS Excel
- 16,8k RAR
- 9,4k JPG
- 904 Hangul
- 2 OO Draw
- 18k MS Word
- 25,9k ZIP

Windows Executables

Windows Executables

Subsystem	DLLs	Executables
WINDOWS_GUI	66.327	162.327
EFI_BOOT_SERVICE_DRIVER	214.887	21.201
WINDOWS_CUI	139.246	10.285
EFI_RUNTIME_DRIVER	24.435	3215
NATIVE	92	888
EFI_APPLICATION	781	400
WINDOWS_CE_GUI	113	59
UNKNOWN	28	36
EFI_ROM	17	0
XBOX	3	0
Total	445.929	198.411

Windows Executables

Subsystem	DLLs	Executables
WINDOWS_GUI	66.327	162.327
EFI_BOOT_SERVICE_DRIVER	214.887	21.201
WINDOWS_CUI	139.246	10.285
EFI_RUNTIME_DRIVER	24.435	3215
NATIVE	92	888
EFI_APPLICATION	781	400
WINDOWS_CE_GUI	113	59
UNKNOWN	28	36
EFI_ROM	17	0
XBOX	3	0
Total	445.929	198.411

Windows Executables

Subsystem	DLLs	Executables
WINDOWS_GUI	66.327	162.327
EFI_BOOT_SERVICE_DRIVER	214.887	21.201
WINDOWS_CUI	139.246	10.285
EFI_RUNTIME_DRIVER	24.435	3215
NATIVE	[illegible]	[illegible]
EFI_APPLICATION	[illegible]	[illegible]
WINDOWS_CE_GUI	[illegible]	[illegible]
UNKNOWN	28	36
EFI_ROM	17	0
XBOX	3	0
Total	445.929	198.411

172,612 executables
subsystem 2 and
subsystem 3
13,7% of the dataset

Windows executables

- 60% of the samples have a size between 100K and 1M
- 98% x86_32, 1,8% x86_64, 0,01% ARM
- 51% of the samples with an entropy higher than 7
- 18,3% binaries are signed (11 with revoked certs)

172k samples are still too many

Sample ingestion pipeline

172k samples are still too many

We design a possible pipeline to process the samples

This pipeline is an instrument:

- Understand the distribution of samples
- Understand the challenges for a company
- Estimate the cost (computational and human)

Sample ingestion pipeline

Pipeline leverages de-facto malware analysis techniques

static analysis
dynamic analysis
manual inspection

Sample ingestion pipeline

VirusTotal

How much can we trust these AVs?

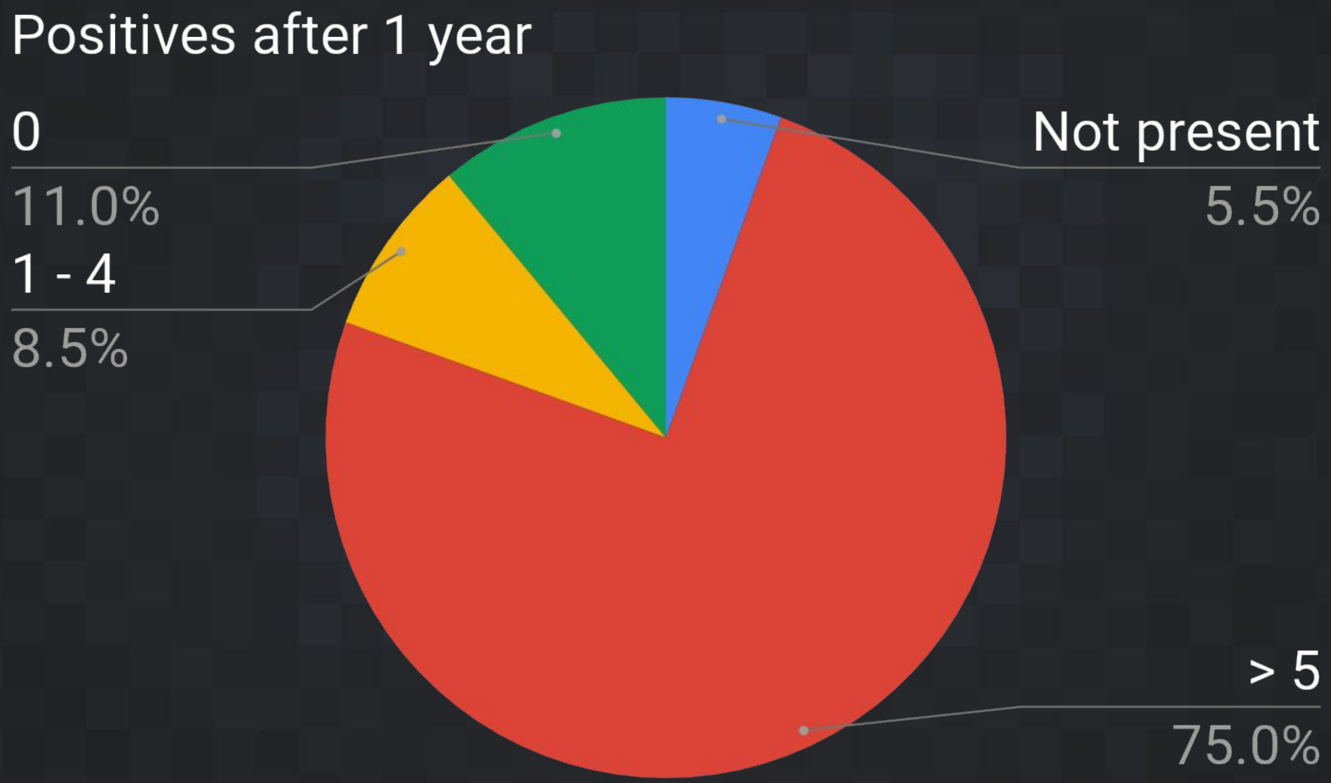
- Time of last scan vs current detection
- AV configuration parameters might be different
- Different types of engines (some are ML, heuristic…)
- FP prone AVs?
- Inaccurate / generic labels

Sample ingestion pipeline

Number of positives over time:

- January 2016
- July 2016
- January 2017 (rescan)

Sample ingestion pipeline



Sample ingestion pipeline

AV results after one year:

- 4,684 samples from 0 positives to 1+
- 2,281 from 1+ positives to 0
- A few samples removed from VT

3.5% of samples changed their disposition

Sample ingestion pipeline

AVClass[2] (state of the art for AV label aggregation)

69% of the samples classified into 1,057 families

allaple	54,097
virut	16,328
browsefox	7,400
outbrowse	4,600
installcore	2,395

49%

[2] https://github.com/malicialab/avclass

Sample ingestion pipeline

- Samples with no AV class
  - 16.5% not present in VT
  - 67.7% had less than 5 positives
- AV class detected 22% as PUP
  - 87.4% of these had an AVClass
    - browsefox, outbrowse, installcore, eorezo, softpulse, loadmoney

Sample ingestion pipeline

Dynamic analysis

- Extract additional information
- We leveraged a state of the art set up
- Internal to the company, we borrowed processing time
- Tuned and maintained: detonation, disarm anti-analysis, etc…

Sample ingestion pipeline

Dynamic analysis

- Part of the samples showed low / no activity
  - We ran those on a second sandbox

Sample ingestion pipeline

A stunning 19% of the samples did not show a meaningful activity

Table 7. Classification of Samples with No/Low Activity

	No activity	Low activity
GUI	599	270
Missing DLLs	3,814	599
Crash	0	723
Corrupted file	9,805	64
Total	14,218	1,656
Still Unexplained	10,159	6,499

Sample ingestion pipeline

This takes (in one single day)

- 17 GiB of space
- 55 VMs (5 minute per sample)

dedicated to samples that have a GUI, crash, missing dependencies, or are corrupted

Sample ingestion pipeline

We expected to have polymorphic variants

We grouped behavioral reports

Sample ingestion pipeline

Clustering tools / algorithms:

- Custom report normalization
- TLSH[3] (Trendmicro) over the normalized report
  - Take report as input produce locality sensitive hash as output
- Single-linkage + distance based flat clustering

[3] https://github.com/trendmicro/tlsh

Sample ingestion pipeline

1,853 clusters, 6,846 outliers

3 types of clusters

- Majority clusters (65%)
- NoClass clusters (23%)
- Mixed clusters (12%)

Sample ingestion pipeline

But these types do not tell us which type of samples are inside

Which kind of samples do we have?

- **Mk** -> Malicious samples we know (family name)
- **B** -> Samples we know are benign.
- **Mu** -> Malicious samples we have not identified

Sample ingestion pipeline

Which kind of samples do we have?

- **Mk** -> Malicious samples we know (family name)
- **B** -> Samples we know are benign.
- **Mu** -> Malicious samples we have not identified
  - **Mv** -> Variants of Mk
  - **Mg** -> Detected by generic engines / PUP
  - **Mn** -> The rest (we think it could be bad, but cannot automatically assign a class).

Sample ingestion pipeline

We applied re-classification rules to identify samples

Mv, B, Mg

Sample ingestion pipeline

Observed phenomena

1. We can propagate labels
2. We have clusters of “generic” malware, that may not deserve same attention as undetected malware
3. Benign samples usually to show low activity
4. Some clusters considered mixed because of naming inconsistency. E.g.: backupmypc & mypcbackup

Sample ingestion pipeline

As a result:

- Tentatively re-classified 4,946 previously unknown files
- Samples remaining...
  - 2,754 singleton
  - 4,177 unknown samples (**Mn**) in clusters
- We can assign priorities:
  - Singleton + **Mn** → High priority
  - **Mg** clusters → Medium priority

Manual analysis experiments

How much manual analysis effort needed?

3 different experiments

- High priority group
- Samples with low / no activity
- 64 bit binaries

These groups sum up to 24k binaries

Sampled files from each of those groups

Manual analysis experiments

Experiment configuration:

- Analysts with 2 to 6 years of experience
- Asked these questions:
  - GW/MW?
  - Class (keylogger, RAT, botnet) and family?
  - How much time did it take?
  - Which approach did you use?
    - Blackbox
    - Manual
  - Would you need a deeper manual analysis?

Manual analysis experiments

High priority group

- Extracted several samples per cluster and singleton files
  - 52% / 43.2% labelled malicious (5% margin of error)
  - ~3% / ~5% required manual analysis
  - Malware type and family, 5% better for clustered samples vs singleton samples.

- Cross-checked verdicts for clusters
  - 86% verdicts were consistent

Manual analysis experiments

64bit files (2,603 samples)

- 82% have 0 positives
  - From 101 selected files only 11 should require further inspection.
- For the rest
  - 67% considered benign

Manual analysis experiments

(Low | No) activity group

- Extracted 349 samples from each group
- Same info (including screenshots / video of execution)
- 2 additional questions:
  - Does it have a GUI?
  - Does it show a crash?
- Overall 81% | 91% either considered benign, GUI or crash.

Manual analysis experiments

- Estimation: ~27k samples either require interaction, crashed, corrupted, missing dependencies
  - 100 VMs per day if ran on a sandbox

- Between 30 sec and 90 min to inspect the info / samples
  - Estimation: 900 hours to take a cursory look at the 24k unknown samples.

Takeaways

Takeaways

1. Complete analysis: 600 machines (5 min/sample)
2. Community info: only 3.5% of changed verdicts
3. Automated pipeline reclassified 16% of samples
4. Manual inspection of remaining 15% would take >100 person-days

Takeaways

6. But only 5% of samples marked as requiring additional manual inspection

   Substitute decision process by ML?

7. Up to 16% of resources consumed by samples that do not run properly

Real world datasets

Real world datasets

- Not balanced
- No clear labels
  - Ground truth does not exist
- No clear way to deal sample corruption
- Files treated individually (dependencies?)
- No info about how / where it was collected
- Almost no metadata
  - No info about how to run (parameters? environment?)

Sample ingestion strategies must deal with uncertainty

Pipelines & prioritization strategies

In our daily operations we need to configure

- Heuristic rules
- Thresholds

Systematic measurement and analysis

We must not make blind assumptions about our data

More info...

Link to the paper:

“A Close Look at a Daily Dataset of Malware Samples”
ACM Transactions on Privacy and Security

http://s3.eurecom.fr/docs/tops19_dailymalware.pdf

Q&A

TALOSINTELLIGENCE.COM

blog.talosintelligence.com

@talossecurity