Immunity from antimalware automation attacks

Dennis Batchelder
Hong Jia
October 2013

Call to action

A new bad guy is weaponizing our antimalware products

We’re getting thousands of incoming “crafted” files and suspect telemetry every month

- Probing our automation strategies and signature weak points
- Poisoning our data sources
- Exploiting how we share samples between ourselves

Our industry inadvertently assists the attackers

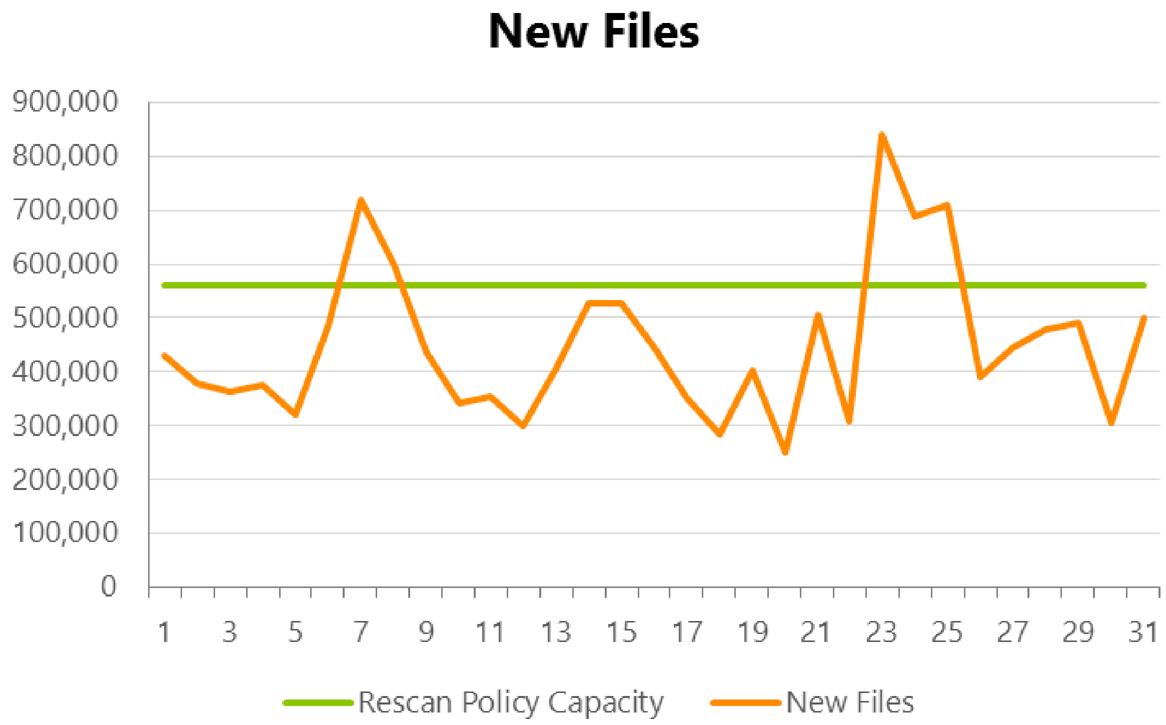
Let’s work together to fix things before we have a catastrophe

AGENDA

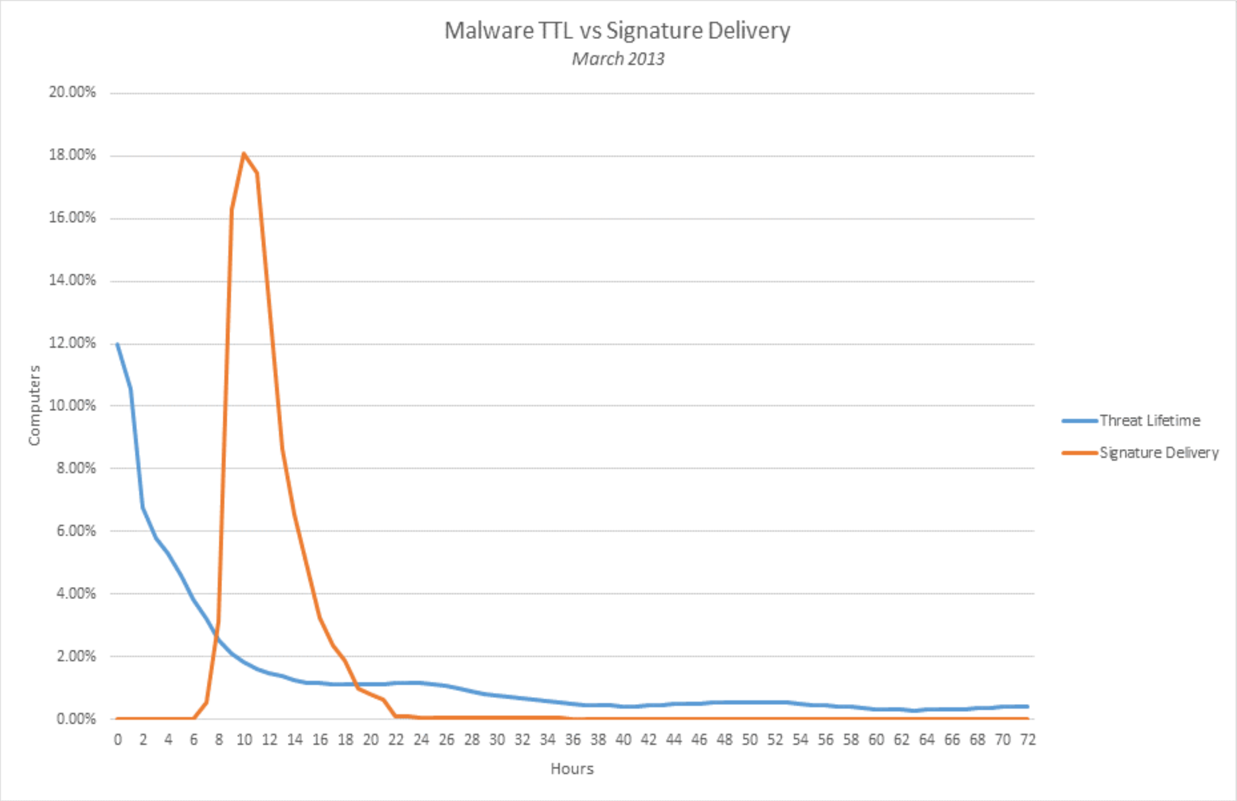
How we got here
The new attacks
The aftermath
Recommendations

How we got here

We automate for good reasons

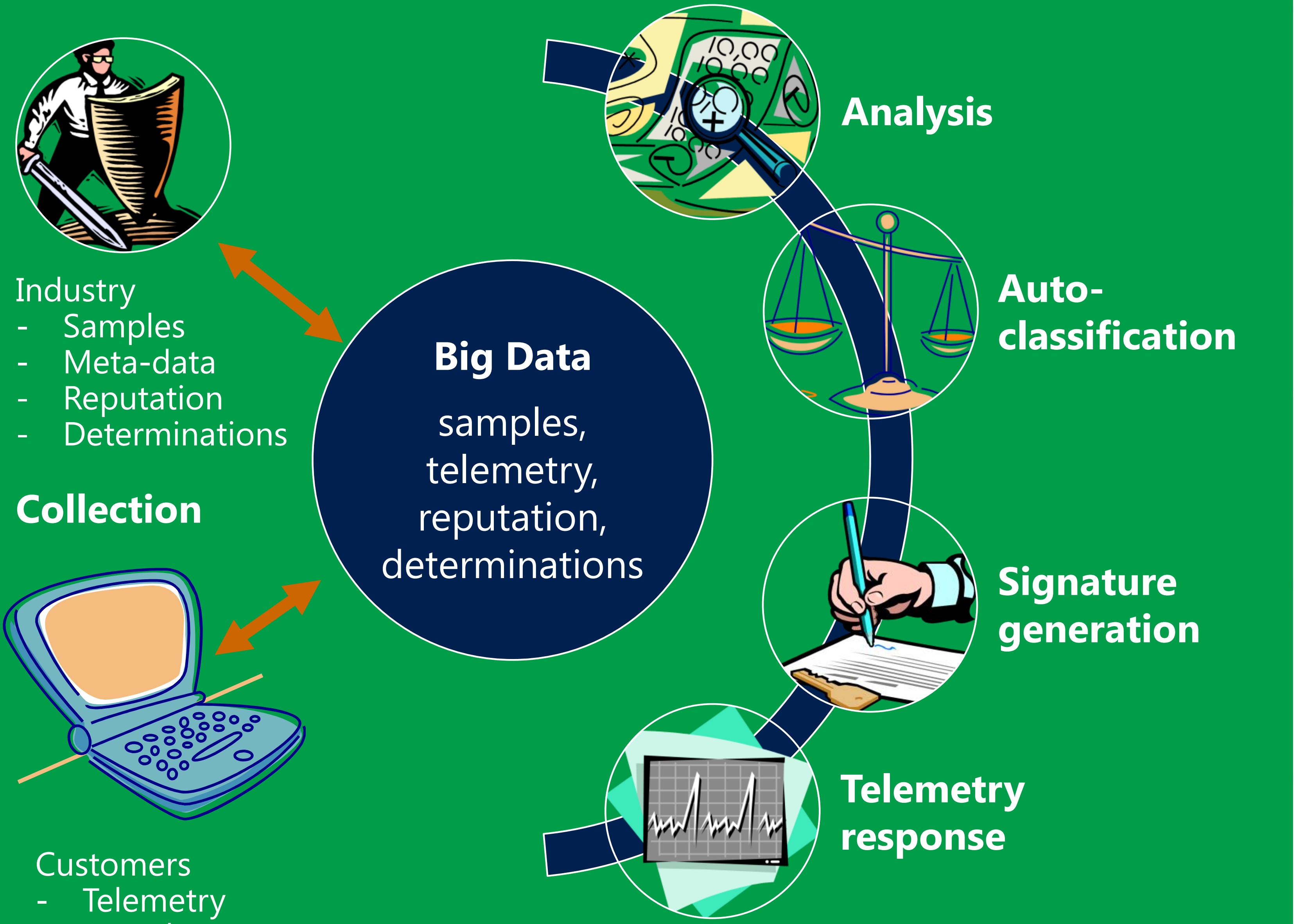


High malware volume



Short malware lifecycle

Antimalware automation



Industry
- Samples
- Meta-data
- Reputation
- Determinations

Collection

Customers
- Telemetry
- Samples

Big Data

samples, telemetry, reputation, determinations

Analysis

Auto-classification

Signature generation

Telemetry response

Collection
- Industry and customers
- Automatic and on demand

Big Data
- Samples
- Map reduce
- Processed/Workflow

Analysis
- Dynamic and Static
- Vendor rescans/determinations
- Human-supplied patterns

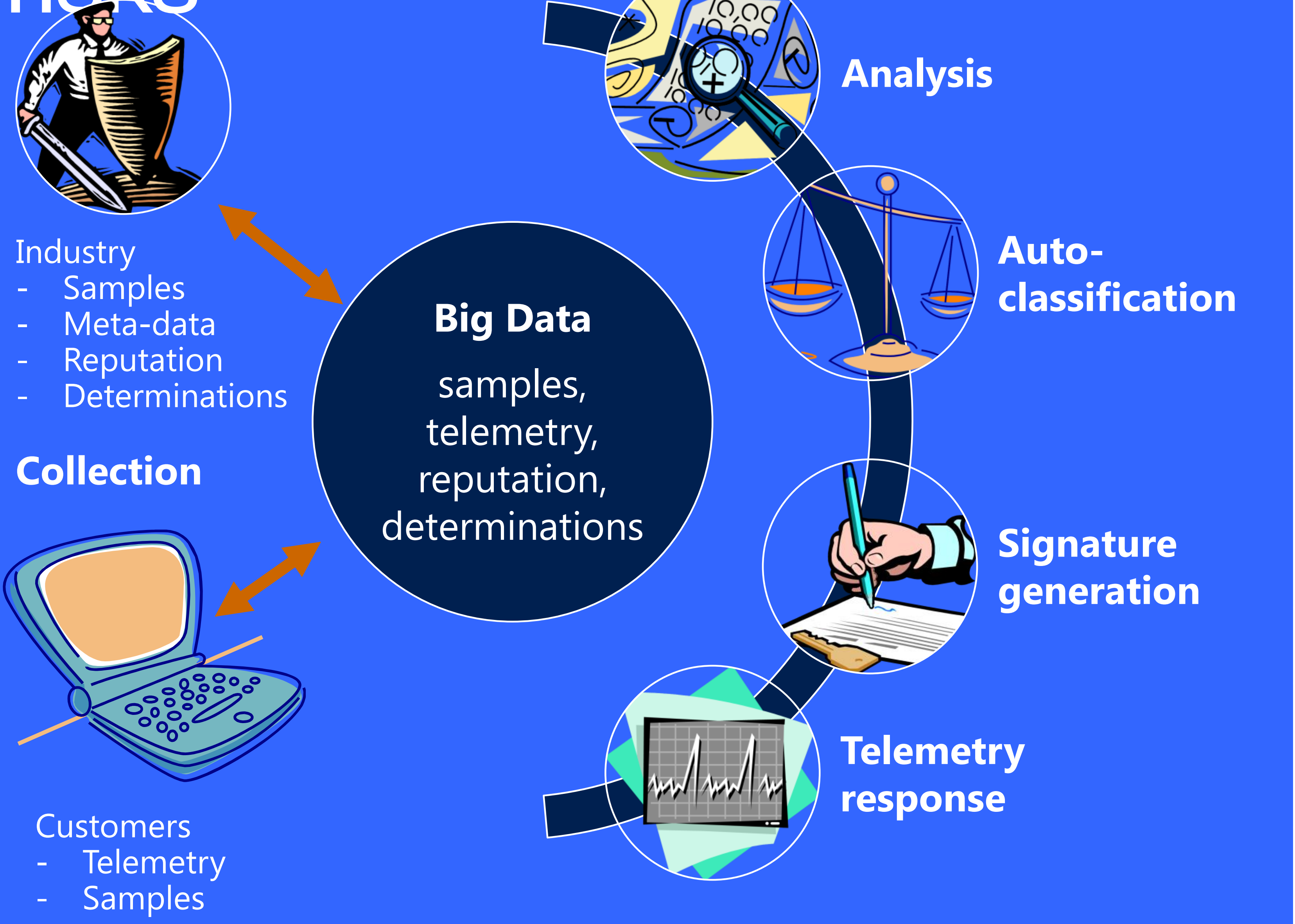
Auto-classification
- Combine analysis with reputation
- Assign determination, family
- Feeds sig-gen and cloud protection

Signature Generation
- Best-fit signature
- Static and proactive
- Signature release pipeline

Telemetry Monitoring
- FP detection
- Never unknowns
- Sample requests

We know how to handle risks of infrastructure attacks…

Infrastructure automation risks



Industry
- Samples
- Meta-data
- Reputation
- Determinations

Collection

Customers
- Telemetry
- Samples

Big Data

samples, telemetry, reputation, determinations

Analysis

Auto-classification

Signature generation

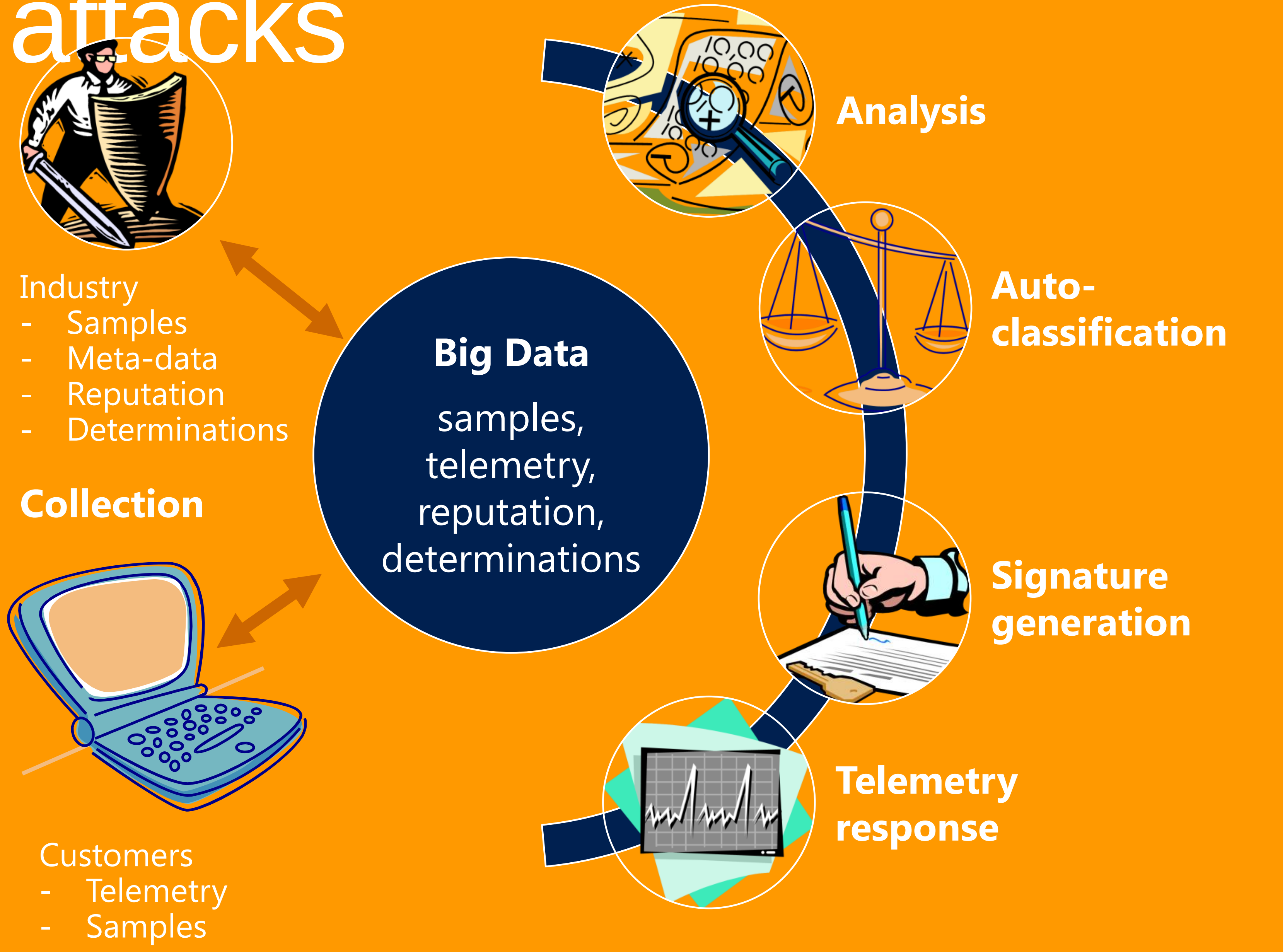
Telemetry response

Risk	Mitigation
Denial of Service blocking samples and telemetry	Collection network protection
Overload causing slow time to protect	Scale-out architectures
Analysis exploits taking down/infecting backend systems	Sandboxing, quotas
Staleness reducing effectiveness	Recency weighting, Curated samples
Outage	Georedundancy
FPs	Signature validation pipeline, large clean lists, live monitoring
Malware infections	Isolation, monitoring
Malware leakage	Sharing agreements, air gaps, physical security
PII disclosure	Data cleansing and auditing

But what if data itself is the attack vector?

-what if the sample isn’t sourced from the wild?
-what if incoming telemetry is lying?
-what if the sample is crafted to exploit us?

Risks of data vector attacks



Industry
- Samples
- Meta-data
- Reputation
- Determinations

Collection

Customers
- Telemetry
- Samples

Big Data

samples, telemetry, reputation, determinations

Analysis

Auto-classification

Signature generation

Telemetry response

Attack	Risk
Fake, probe samples	**Signature bloat, inefficiency**
	Automation strategy leakage
	Signature weakness leakage
	Determination trust leakage
Fake telemetry	**Poisoning file reputation**
	Signing trigger leakage
Crafted samples	**Wide-spread or targeted FPs**
	Financial and brand damages

Evil recipe for weaponizing AV products

Learn system weaknesses

- ❑What causes us to accept samples
- ❑How samples spread around the industry
- ❑Which vendor determinations we trust
- ❑What triggers us to use different kinds of signatures
- ❑Holes in our signatures
- ❑Holes in our automation

Launch the attack

- ❑Craft a sample that:
  - ❑Encourages target vendor to sign it
  - ❑Exploits target vendors signature weakness
- ❑Inject sample and telemetry into the system
- ❑Wait, then watch the mess

And why should we care?

Nobody should be able to exploit our systems…

- It hurts our customers
- And damages our reputation

…no matter the motive

- No having fun at our expense
- No embarrassing the security industry
- No preventing us from working together
- No attacks without our knowledge

So, has it happened?

We’ve seen…

Attack sophistication

- Crafted files moving from clean to junk to malicious files
- Use of TOR for sample and telemetry submission

Microsoft-specific targeting

- Discovered an automation strategy weakness and a weak signature type

Broad industry targeting

- Crafted files targeting other AV vendors
- Embedding our (and other) signature fragments as triggers
- Exposing weaknesses in how we exchange samples between ourselves/testers

6 March – 12 April

Assumed goal: automation holes

Method to craft

- Insert signature fragments into clean files' resource sections
- Submit to VirusTotal via TOR

Results

- ~300 crafted clean files (never seen in wild)
- Many vendors re-sharing and signing
- Our automation treated it as obfuscated sample
- FP with proactive signature on clean code
- Partner FP on copied signature

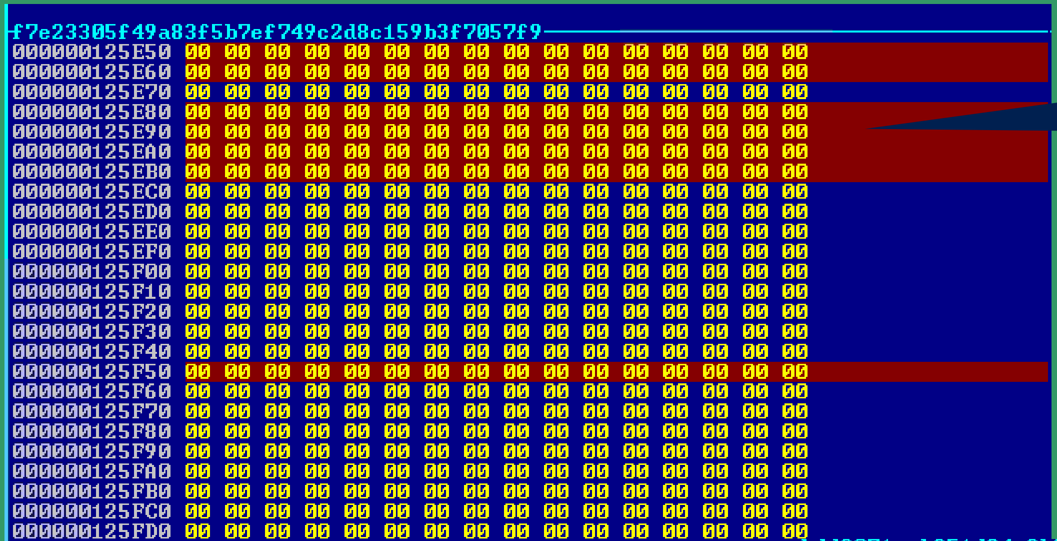
Learn system weaknesses

- ☑ What causes us to accept samples
- ☑ How samples spread around the industry
- ☑ Which vendor determinations we trust
- ☑ What triggers us to use different kinds of signatures
- □ Holes in our automation
- □ Holes in our signatures

Launch the attack

- □ Craft a sample that:
  - □ Encourages target vendor to sign it
  - □ Exploits target vendors signature weakness
- □ Inject sample and telemetry into the system
- □ Wait, the e mess

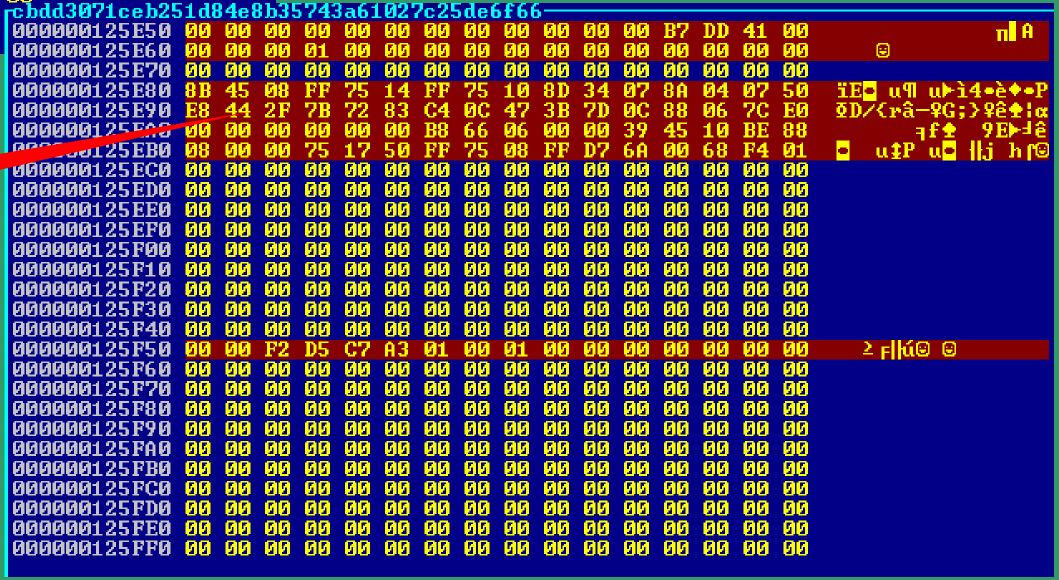
Crafted clean files



```
f7e23305f49a83f5b7ef749c2d8c159b3f7057f9
000000125E50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125E60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125E70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125E80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125E90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125EA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125EB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125EC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125ED0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125EE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125EF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125FA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125FB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125FC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125FD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125FE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125FF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

NULLs in .rsrc

f7e23305f49a83f5b7ef749c2d8c159b3f7057f9
(Epson Brother file)



```
cbdd3071ceb251d84e8b35743a61027c25de6f66
000000125E50 00 00 00 00 00 00 00 00 00 00 00 00 B7 DD 41 00               ╖▌A
000000125E60 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00      ☺
000000125E70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125E80 8B 45 08 FF 75 14 FF 75 10 8D 34 07 8A 04 07 50   ïE◘ u¶ u►ì4•è♦•P
000000125E90 E8 44 2F 7B 72 83 C4 0C 47 3B 7D 0C 88 06 7C E0   ΦD/{râ─♀G;}♀ê♠|α
000000125EA0 00 00 00 00 00 00 B8 66 06 00 00 39 45 10 BE 88         ╕f♠  9E►╛ê
000000125EB0 08 00 00 75 17 50 FF 75 08 FF D7 6A 00 68 F4 01   ◘  u↨P u◘ ╫j h⌠☺
000000125EC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125ED0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125EE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125EF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F50 00 00 F2 D5 C7 A3 01 00 01 00 00 00 00 00 00 00     ≥╒╟ú☺ ☺
000000125F60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125F90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125FA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125FB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125FC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125FD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125FE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000125FF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Signature Fragment in .rsrc

CBDD3071CEB251D84E8B35743A61027C25DE6F66

29 April – present

Assumed goal: signature holes

Method to craft

- Build junk files attempting to cause signature hash collisions
- Insert sig fragment strings/heads to cause “trusted” vendor detections
- Submit to VirusTotal via TOR

Results

- ~2000 crafted junk files (never seen in wild)
- Many vendors re-sharing and signing
- Some vendors sharing with external testers

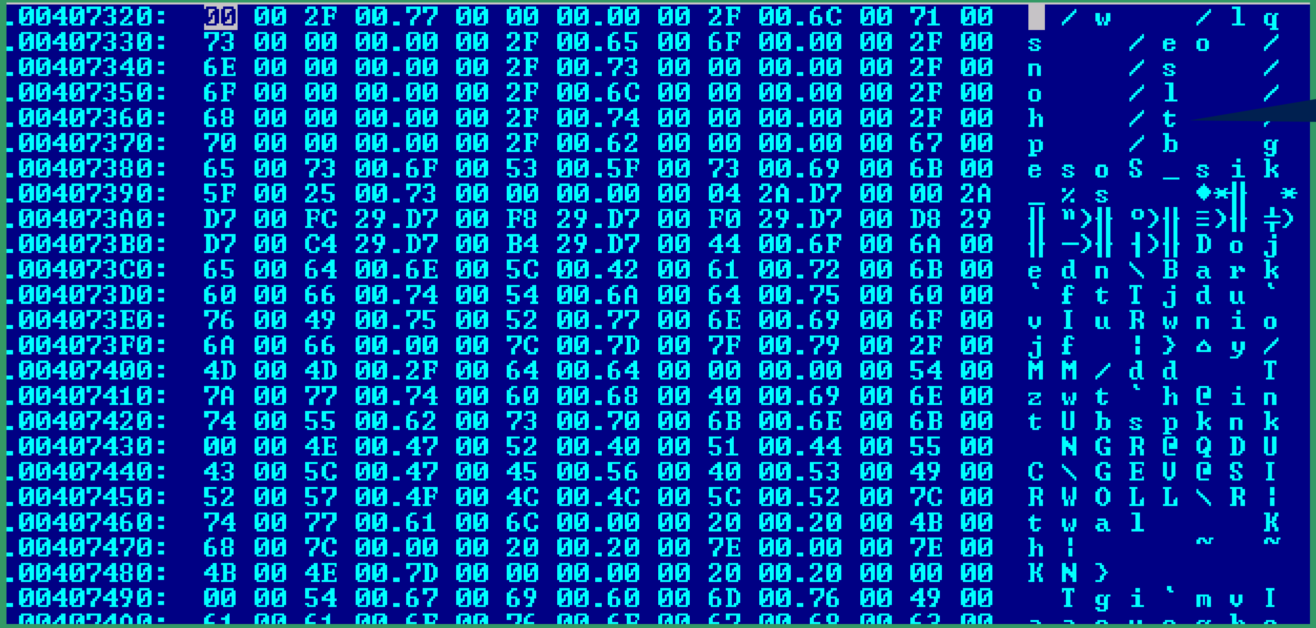
Learn system weaknesses

- ☑ What causes us to accept samples
- ☑ How samples spread around the industry
- ☑ Which vendor determinations we trust
- ☑ What triggers us to use different kinds of signatures
- ☑ Holes in our automation
- ☑ Holes in our signatures

Launch the attack

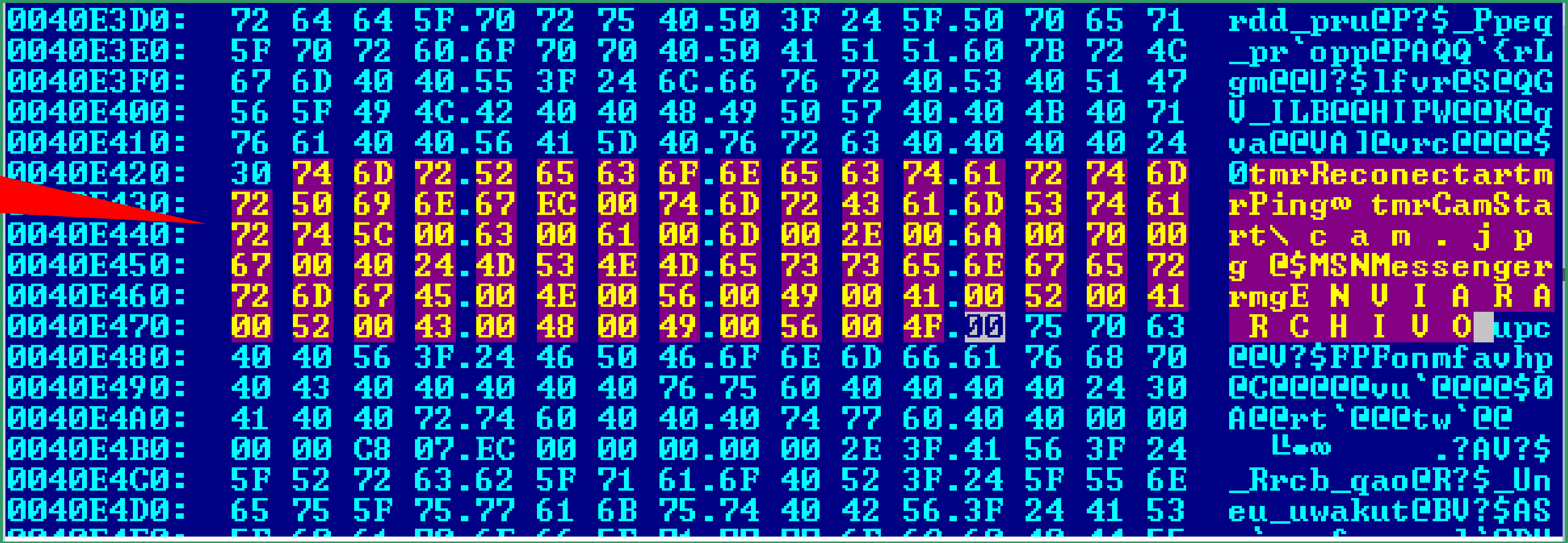
- ❑ Craft a sample that:
  - ❑ Encourages target vendor to sign it
  - ❑ Exploits target vendors signature weakness
- ❑ Inject sample and telemetry into the system
- ❑ Wait, the [illegible]e mess

Crafted junk file 0x361d9b1375bf5f49f4b9f2f9fc4398d5ffdb3553



```
.00407320:  00 00 2F 00.77 00 00 00.00 00 2F 00.6C 00 71 00    / w    / l q
.00407330:  73 00 00 00.00 00 2F 00.65 00 6F 00.00 00 2F 00   s     / e o   /
.00407340:  6E 00 00 00.00 00 2F 00.73 00 00 00.00 00 2F 00   n     / s     /
.00407350:  6F 00 00 00.00 00 2F 00.6C 00 00 00.00 00 2F 00   o     / l     /
.00407360:  68 00 00 00.00 00 2F 00.74 00 00 00.00 00 2F 00   h     / t     /
.00407370:  70 00 00 00.00 00 2F 00.62 00 00 00.00 00 67 00   p     / b     g
.00407380:  65 00 73 00.6F 00 53 00.5F 00 73 00.69 00 6B 00   e s o S _ s i k
.00407390:  5F 00 25 00.73 00 00 00.00 00 04 2A.D7 00 00 2A   _ % s     ♦*╫  *
.004073A0:  D7 00 FC 29.D7 00 F8 29.D7 00 F0 29.D7 00 D8 29   ╫ ⁿ)╫ °)╫ ≡)╫ ╪)
.004073B0:  D7 00 C4 29.D7 00 B4 29.D7 00 44 00.6F 00 6A 00   ╫ ─)╫ ┤)╫ D o j
.004073C0:  65 00 64 00.6E 00 5C 00.42 00 61 00.72 00 6B 00   e d n \ B a r k
.004073D0:  60 00 66 00.74 00 54 00.6A 00 64 00.75 00 60 00   ` f t T j d u `
.004073E0:  76 00 49 00.75 00 52 00.77 00 6E 00.69 00 6F 00   v I u R w n i o
.004073F0:  6A 00 66 00.00 00 7C 00.7D 00 7F 00.79 00 2F 00   j f   | } ⌂ y /
.00407400:  4D 00 4D 00.2F 00 64 00.64 00 00 00.00 00 54 00   M M / d d     T
.00407410:  7A 00 77 00.74 00 60 00.68 00 40 00.69 00 6E 00   z w t ` h @ i n
.00407420:  74 00 55 00.62 00 73 00.70 00 6B 00.6E 00 6B 00   t U b s p k n k
.00407430:  00 00 4E 00.47 00 52 00.40 00 51 00.44 00 55 00     N G R @ Q D U
.00407440:  43 00 5C 00.47 00 45 00.56 00 40 00.53 00 49 00   C \ G E V @ S I
.00407450:  52 00 57 00.4F 00 4C 00.4C 00 5C 00.52 00 7C 00   R W O L L \ R |
.00407460:  74 00 77 00.61 00 6C 00.00 00 20 00.20 00 4B 00   t w a l       K
.00407470:  68 00 7C 00.00 00 20 00.20 00 7E 00.00 00 7E 00   h |       ~   ~
.00407480:  4B 00 4E 00.7D 00 00 00.00 00 20 00.20 00 00 00   K N }
.00407490:  00 00 54 00.67 00 69 00.60 00 6D 00.76 00 49 00     T g i ` m v I
```

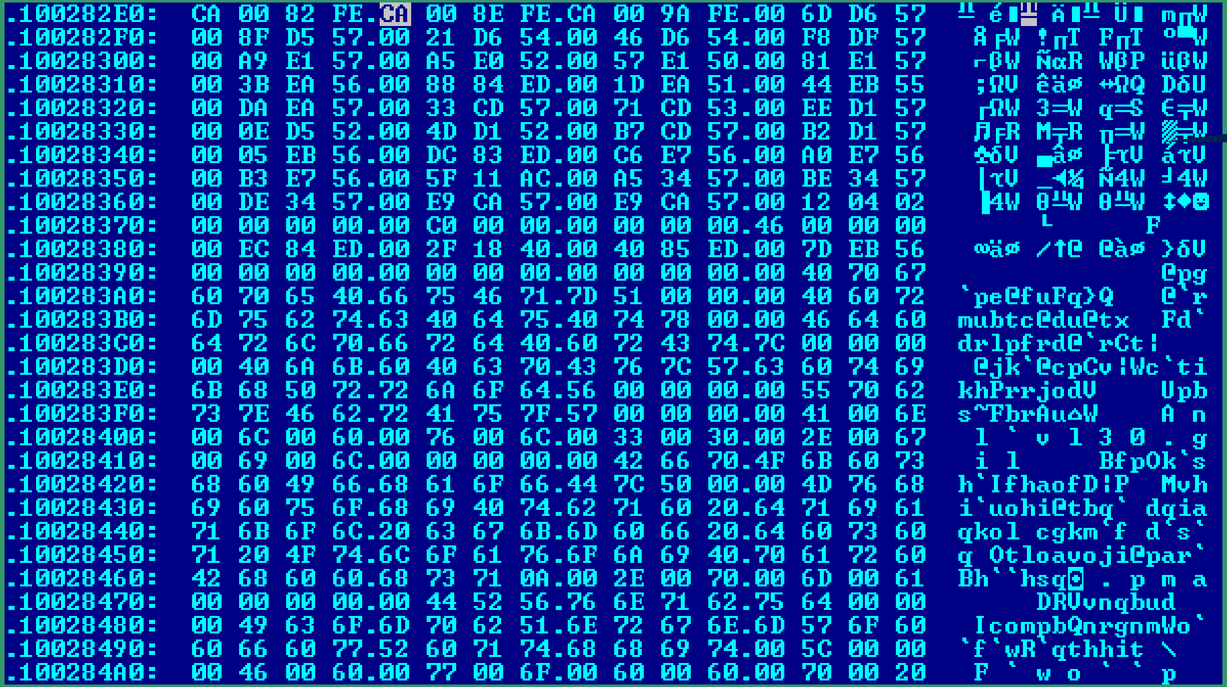
Junk import table



```
0040E3D0:  72 64 64 5F.70 72 75 40.50 3F 24 5F.50 70 65 71  rdd_pru@P?$_Ppeq
0040E3E0:  5F 70 72 60.6F 70 70 40.50 41 51 51.60 7B 72 4C  _pr`opp@PAQQ`{rL
0040E3F0:  67 6D 40 40.55 3F 24 6C.66 76 72 40.53 40 51 47  gm@@U?$lfvr@S@QG
0040E400:  56 5F 49 4C.42 40 40 48.49 50 57 40.40 4B 40 71  V_ILB@@HIPW@@K@q
0040E410:  76 61 40 40.56 41 5D 40.76 72 63 40.40 40 40 24  va@@VA]@vrc@@@@$
0040E420:  30 74 6D 72.52 65 63 6F.6E 65 63 74.61 72 74 6D  0tmrReconectartm
0040E430:  72 50 69 6E.67 EC 00 74.6D 72 43 61.6D 53 74 61  rPing∞ tmrCamSta
0040E440:  72 74 5C 00.63 00 61 00.6D 00 2E 00.6A 00 70 00  rt\ c a m . j p
0040E450:  67 00 40 24.4D 53 4E 4D.65 73 73 65.6E 67 65 72  g @$MSNMessenger
0040E460:  72 6D 67 45.00 4E 00 56.00 49 00 41.00 52 00 41  rmgE N V I A R A
0040E470:  00 52 00 43.00 48 00 49.00 56 00 4F.00 75 70 63   R C H I V O upc
0040E480:  40 40 56 3F.24 46 50 46.6F 6E 6D 66.61 76 68 70  @@V?$FPFonmfavhp
0040E490:  40 43 40 40.40 40 40 76.75 60 40 40.40 40 24 30  @C@@@@@vu`@@@@$0
0040E4A0:  41 40 40 72.74 60 40 40.40 74 77 60.40 40 00 00  A@@rt`@@@tw`@@
0040E4B0:  00 00 C8 07.EC 00 00 00.00 00 2E 3F.41 56 3F 24    ╚•∞     .?AV?$
0040E4C0:  5F 52 72 63.62 5F 71 61.6F 40 52 3F.24 5F 55 6E  _Rrcb_qao@R?$_Un
0040E4D0:  65 75 5F 75.77 61 6B 75.74 40 42 56.3F 24 41 53  eu_uwakut@BV?$AS
```

Embedded signature fragments

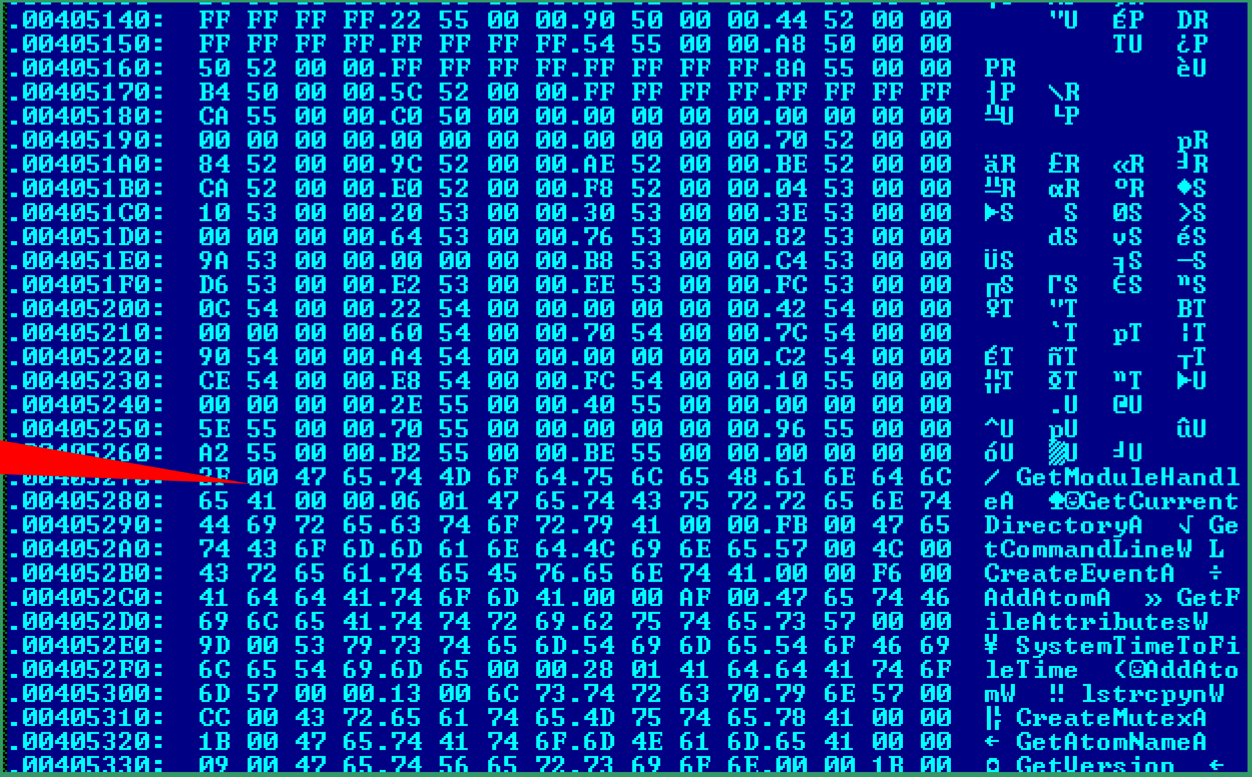
Crafted junk file, signature collision with malware



```
.100282E0:  CA 00 82 FE.CA 00 8E FE.CA 00 9A FE.00 6D D6 57  ╩ é■╩ Ä■╩ Ü■ m╓W
.100282F0:  00 8F D5 57.00 21 D6 54.00 46 D6 54.00 F8 DF 57   Å╒W !╓T F╓T °▀W
.10028300:  00 A9 E1 57.00 A5 E0 52.00 57 E1 50.00 81 E1 57   ⌐ßW ÑαR WßP üßW
.10028310:  00 3B EA 56.00 88 84 ED.00 1D EA 51.00 44 EB 55   ;ΩV êäφ ↔ΩQ DδU
.10028320:  00 DA EA 57.00 33 CD 57.00 71 CD 53.00 EE D1 57   ┌ΩW 3═W q═S ε╤W
.10028330:  00 0E D5 52.00 4D D1 52.00 B7 CD 57.00 B2 D1 57   ♫╒R M╤R ╖═W ▓╤W
.10028340:  00 05 EB 56.00 DC 83 ED.00 C6 E7 56.00 A0 E7 56   ♣δV ▄âφ ╞τV áτV
.10028350:  00 B3 E7 56.00 5F 11 AC.00 A5 34 57.00 BE 34 57   │τV _◄¼ Ñ4W ╛4W
.10028360:  00 DE 34 57.00 E9 CA 57.00 E9 CA 57.00 12 04 02   ▐4W Θ╩W Θ╩W ↕♦☻
.10028370:  00 00 00 00.00 C0 00 00.00 00 00 00.46 00 00 00       └       F
.10028380:  00 EC 84 ED.00 2F 18 40.00 40 85 ED.00 7D EB 56   ∞äφ /↑@ @àφ }δV
.10028390:  00 00 00 00.00 00 00 00.00 00 00 00.00 40 70 67              @pg
.100283A0:  60 70 65 40.66 75 46 71.7D 51 00 00.00 40 60 72  `pe@fuFq}Q   @`r
.100283B0:  6D 75 62 74.63 40 64 75.40 74 78 00.00 46 64 60  mubtc@du@tx  Fd`
.100283C0:  64 72 6C 70.66 72 64 40.60 72 43 74.7C 00 00 00  drlpfrd@`rCt|
.100283D0:  00 40 6A 6B.60 40 63 70.43 76 7C 57.63 60 74 69   @jk`@cpCv|Wc`ti
.100283E0:  6B 68 50 72.72 6A 6F 64.56 00 00 00.00 55 70 62  khPrrjodV    Upb
.100283F0:  73 7E 46 62.72 41 75 7F.57 00 00 00.00 41 00 6E  s~FbrAu⌂W    A n
.10028400:  00 6C 00 60.00 76 00 6C.00 33 00 30.00 2E 00 67   l ` v l 3 0 . g
.10028410:  00 69 00 6C.00 00 00 00.00 42 66 70.4F 6B 60 73   i l     BfpOk`s
.10028420:  68 60 49 66.68 61 6F 66.44 7C 50 00.00 4D 76 68  h`IfhaofD|P  Mvh
.10028430:  69 60 75 6F.68 69 40 74.62 71 60 20.64 71 69 61  i`uohi@tbq` dqia
.10028440:  71 6B 6F 6C.20 63 67 6B.6D 60 66 20.64 60 73 60  qkol cgkm`f d`s`
.10028450:  71 20 4F 74.6C 6F 61 76.6F 6A 69 40.70 61 72 60  q Otloavoji@par`
.10028460:  42 68 60 60.68 73 71 0A.00 2E 00 70.00 6D 00 61  Bh``hsq◙ . p m a
.10028470:  00 00 00 00.00 44 52 56.76 6E 71 62.75 64 00 00       DRVvnqbud
.10028480:  00 49 63 6F.6D 70 62 51.6E 72 67 6E.6D 57 6F 60   IcompbQnrgnmWo`
.10028490:  60 66 60 77.52 60 71 74.68 68 69 74.00 5C 00 00  `f`wR`qthhit \
.100284A0:  00 46 00 60.00 77 00 6F.00 60 00 60.00 70 00 20   F ` w o ` ` p
```

Junk file

a622b580ac5748e0cca17879a303178b118862c0

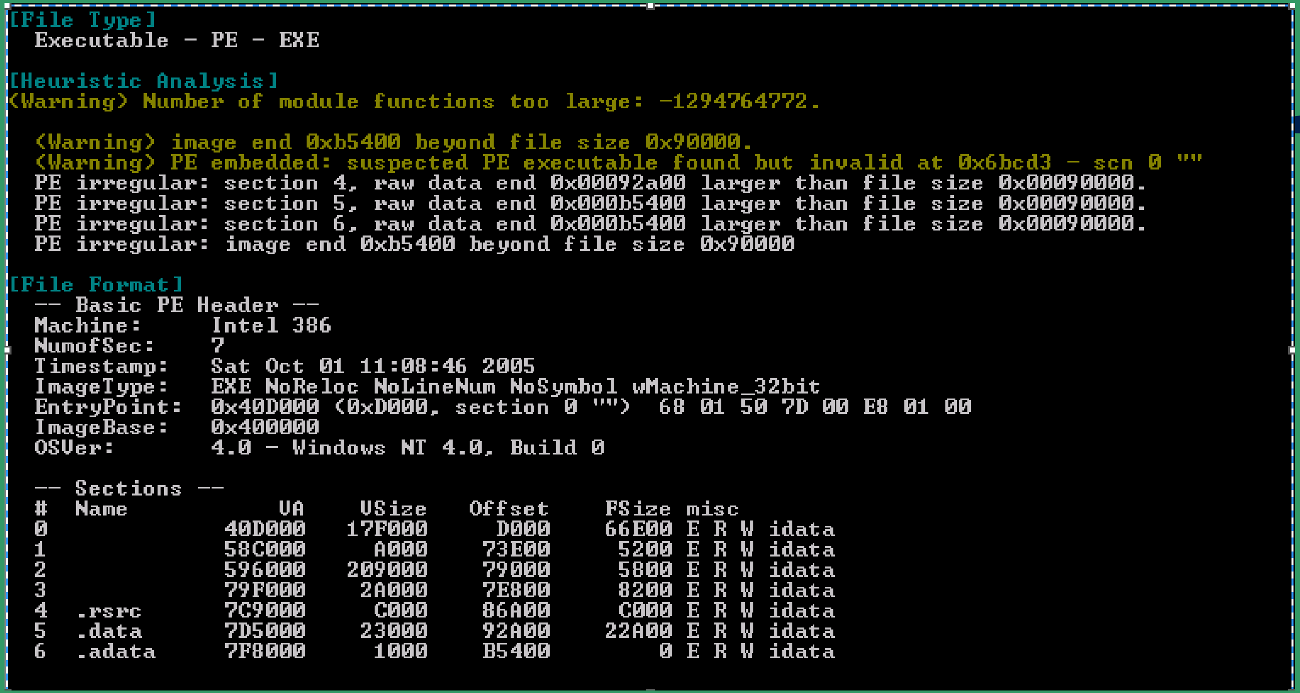


```
.00405140:  FF FF FF FF.22 55 00 00.90 50 00 00.44 52 00 00      "U  ÉP  DR
.00405150:  FF FF FF FF.FF FF FF FF.54 55 00 00.A8 50 00 00          TU  ¿P
.00405160:  50 52 00 00.FF FF FF FF.FF FF FF FF.8A 55 00 00  PR          èU
.00405170:  B4 50 00 00.5C 52 00 00.FF FF FF FF.FF FF FF FF  ┤P  \R
.00405180:  CA 55 00 00.C0 50 00 00.00 00 00 00.00 00 00 00  ╩U  └P
.00405190:  00 00 00 00.00 00 00 00.00 00 00 00.70 52 00 00              pR
.004051A0:  84 52 00 00.9C 52 00 00.AE 52 00 00.BE 52 00 00  äR  £R  «R  ╛R
.004051B0:  CA 52 00 00.E0 52 00 00.F8 52 00 00.04 53 00 00  ╩R  αR  °R  ♦S
.004051C0:  10 53 00 00.20 53 00 00.30 53 00 00.3E 53 00 00  ►S   S  0S  >S
.004051D0:  00 00 00 00.64 53 00 00.76 53 00 00.82 53 00 00      dS  vS  éS
.004051E0:  9A 53 00 00.00 00 00 00.B8 53 00 00.C4 53 00 00  ÜS      ╕S  ─S
.004051F0:  D6 53 00 00.E2 53 00 00.EE 53 00 00.FC 53 00 00  ╓S  ΓS  εS  ⁿS
.00405200:  0C 54 00 00.22 54 00 00.00 00 00 00.42 54 00 00  ♀T  "T      BT
.00405210:  00 00 00 00.60 54 00 00.70 54 00 00.7C 54 00 00      `T  pT  |T
.00405220:  90 54 00 00.A4 54 00 00.00 00 00 00.C2 54 00 00  ÉT  ñT      ┬T
.00405230:  CE 54 00 00.E8 54 00 00.FC 54 00 00.10 55 00 00  ╬T  ΦT  ⁿT  ►U
.00405240:  00 00 00 00.2E 55 00 00.40 55 00 00.00 00 00 00      .U  @U
.00405250:  5E 55 00 00.70 55 00 00.00 00 00 00.96 55 00 00  ^U  pU      ûU
.00405260:  A2 55 00 00.B2 55 00 00.BE 55 00 00.00 00 00 00  óU  ▓U  ╛U
.00405270:  2F 00 47 65.74 4D 6F 64.75 6C 65 48.61 6E 64 6C  / GetModuleHandl
.00405280:  65 41 00 00.06 01 47 65.74 43 75 72.72 65 6E 74  eA  ♠☺GetCurrent
.00405290:  44 69 72 65.63 74 6F 72.79 41 00 00.FB 00 47 65  DirectoryA  √ Ge
.004052A0:  74 43 6F 6D.6D 61 6E 64.4C 69 6E 65.57 00 4C 00  tCommandLineW L
.004052B0:  43 72 65 61.74 65 45 76.65 6E 74 41.00 00 F6 00  CreateEventA  ÷
.004052C0:  41 64 64 41.74 6F 6D 41.00 00 AF 00.47 65 74 46  AddAtomA  » GetF
.004052D0:  69 6C 65 41.74 74 72 69.62 75 74 65.73 57 00 00  ileAttributesW
.004052E0:  9D 00 53 79.73 74 65 6D.54 69 6D 65.54 6F 46 69  ¥ SystemTimeToFi
.004052F0:  6C 65 54 69.6D 65 00 00.28 01 41 64.64 41 74 6F  leTime  (☺AddAto
.00405300:  6D 57 00 00.13 00 6C 73.74 72 63 70.79 6E 57 00  mW  ‼ lstrcpynW
.00405310:  CC 00 43 72.65 61 74 65.4D 75 74 65.78 41 00 00  ╠ CreateMutexA
.00405320:  1B 00 47 65.74 41 74 6F.6D 4E 61 6D.65 41 00 00  ← GetAtomNameA
.00405330:  09 00 47 65.74 56 65 72.73 69 6F 6E.00 00 1B 00  ○ GetVersion  ←
```

“Static” signature collides with Trojan:Win32/Simda

F8A12B809909112BA9E4F175F4D262EE9DEC8DB1

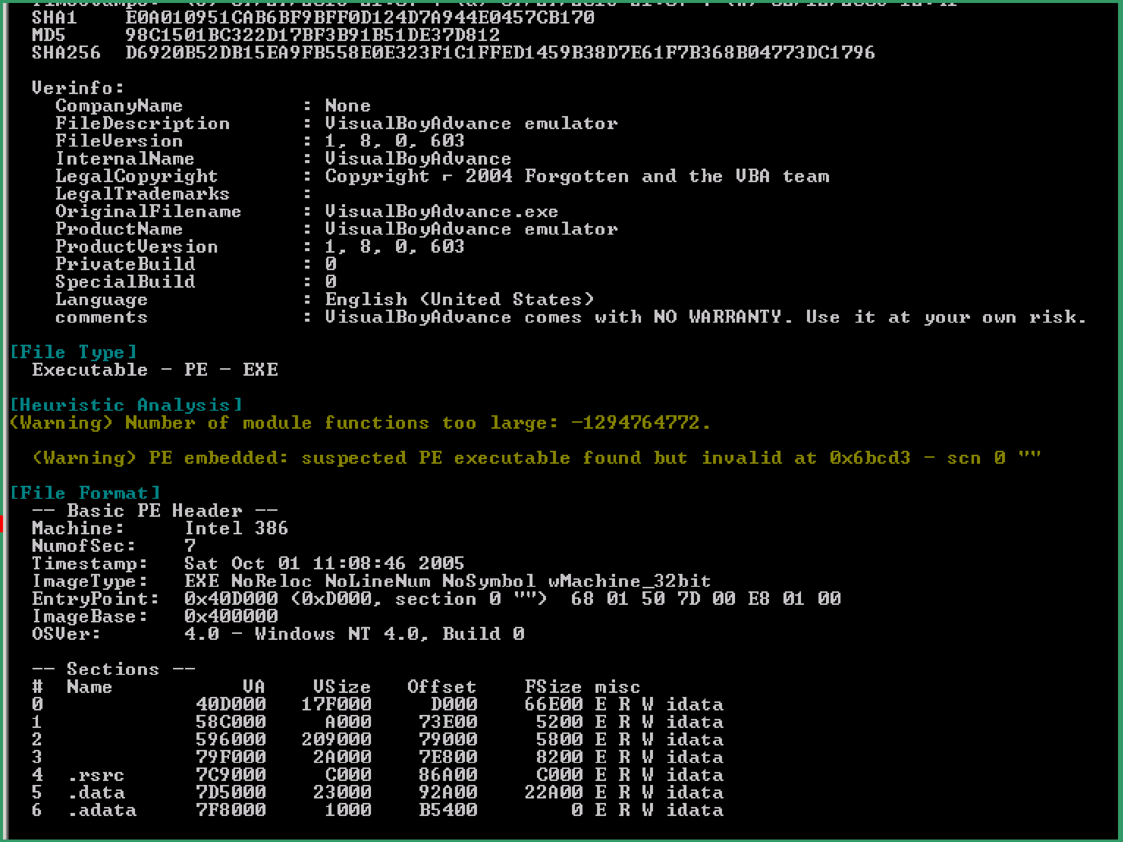
Junk file, signature collision with clean



```
[File Type]
  Executable - PE - EXE

[Heuristic Analysis]
(Warning) Number of module functions too large: -1294764772.

  (Warning) image end 0xb5400 beyond file size 0x90000.
  (Warning) PE embedded: suspected PE executable found but invalid at 0x6bcd3 - scn 0 ""
  PE irregular: section 4, raw data end 0x00092a00 larger than file size 0x00090000.
  PE irregular: section 5, raw data end 0x000b5400 larger than file size 0x00090000.
  PE irregular: section 6, raw data end 0x000b5400 larger than file size 0x00090000.
  PE irregular: image end 0xb5400 beyond file size 0x90000

[File Format]
  -- Basic PE Header --
  Machine:      Intel 386
  NumofSec:     7
  Timestamp:    Sat Oct 01 11:08:46 2005
  ImageType:    EXE NoReloc NoLineNum NoSymbol wMachine_32bit
  EntryPoint:   0x40D000 (0xD000, section 0 "")  68 01 50 7D 00 E8 01 00
  ImageBase:    0x400000
  OSVer:        4.0 - Windows NT 4.0, Build 0

  -- Sections --
  #  Name            VA     VSize    Offset     FSize misc
  0              40D000    17F000      D000     66E00 E R W idata
  1              58C000      A000     73E00      5200 E R W idata
  2              596000    209000     79000      5800 E R W idata
  3              79F000     2A000     7E800      8200 E R W idata
  4  .rsrc       7C9000      C000     86A00      C000 E R W idata
  5  .data       7D5000     23000     92A00     22A00 E R W idata
  6  .adata      7F8000      1000     B5400         0 E R W idata
```

Junk file

E701CB39382BB6349BCCD0861F7BFB1BB4F76EA1



```
  SHA1     E0A010951CAB6BF9BFF0D124D7A944E0457CB170
  MD5      98C1501BC322D17BF3B91B51DE37D812
  SHA256   D6920B52DB15EA9FB558E0E323F1C1FFED1459B38D7E61F7B368B04773DC1796

  Verinfo:
    CompanyName          : None
    FileDescription      : VisualBoyAdvance emulator
    FileVersion          : 1, 8, 0, 603
    InternalName         : VisualBoyAdvance
    LegalCopyright       : Copyright ┌ 2004 Forgotten and the VBA team
    LegalTrademarks      :
    OriginalFilename     : VisualBoyAdvance.exe
    ProductName          : VisualBoyAdvance emulator
    ProductVersion       : 1, 8, 0, 603
    PrivateBuild         : 0
    SpecialBuild         : 0
    Language             : English (United States)
    comments             : VisualBoyAdvance comes with NO WARRANTY. Use it at your own risk.

[File Type]
  Executable - PE - EXE

[Heuristic Analysis]
(Warning) Number of module functions too large: -1294764772.

  (Warning) PE embedded: suspected PE executable found but invalid at 0x6bcd3 - scn 0 ""

[File Format]
  -- Basic PE Header --
  Machine:      Intel 386
  NumofSec:     7
  Timestamp:    Sat Oct 01 11:08:46 2005
  ImageType:    EXE NoReloc NoLineNum NoSymbol wMachine_32bit
  EntryPoint:   0x40D000 (0xD000, section 0 "")  68 01 50 7D 00 E8 01 00
  ImageBase:    0x400000
  OSVer:        4.0 - Windows NT 4.0, Build 0

  -- Sections --
  #  Name            VA     VSize    Offset     FSize misc
  0              40D000    17F000      D000     66E00 E R W idata
  1              58C000      A000     73E00      5200 E R W idata
  2              596000    209000     79000      5800 E R W idata
  3              79F000     2A000     7E800      8200 E R W idata
  4  .rsrc       7C9000      C000     86A00      C000 E R W idata
  5  .data       7D5000     23000     92A00     22A00 E R W idata
  6  .adata      7F8000      1000     B5400         0 E R W idata
```

"Static" signature collides with VisualBoyAdvance

e0a010951cab6bf9bff0d124d7a944e0457cb170

Future (weaponized)

Assumed goal: targeted FP

Method to craft

- Modify real malicious file to cause signature hash collisions with victim clean file
- Compel target vendor to sign with signature fragments from trusted vendor
- Submit to VirusTotal via TOR

Results

- Target vendor signs automatically
- Victim suffers FP against clean file

Learn system weaknesses

- ❑ What causes us to accept samples
- ❑ How samples spread around the industry
- ❑ Which vendor determinations we trust
- ❑ What triggers us to use different kinds of signatures
- ❑ Holes in our automation
- ❑ Holes in our signatures

✓ Launch the attack

- ✓ ❑ Craft a sample that:
  - ✓ ❑ Encourages target vendor to sign it
  - ✓ ❑ Exploits target vendors signature weakness
- ✓ ❑ Inject sample and telemetry into the system
- ❑ Wait, the e mess

Our recent investigations

Did we get used as a weapon?

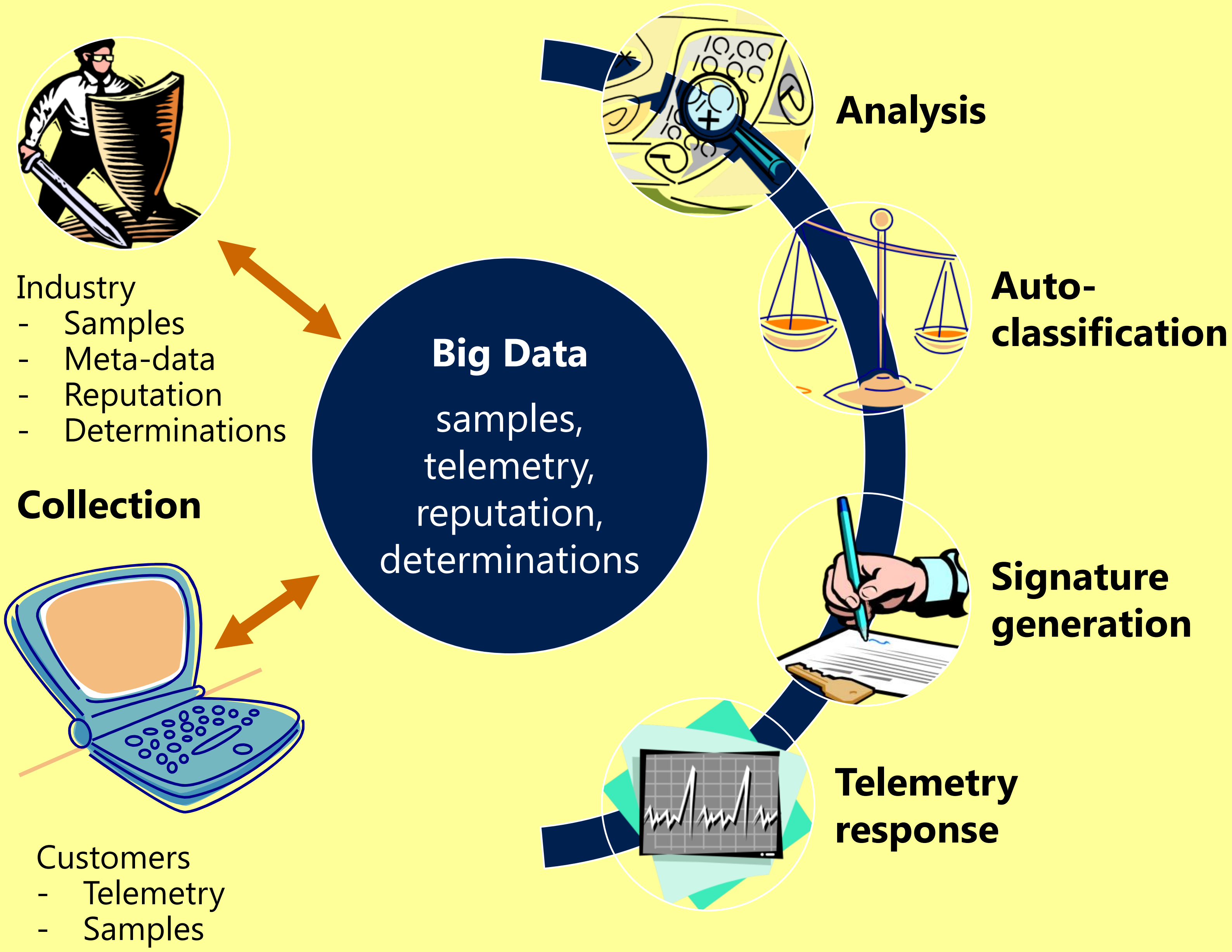
- We searched for an event in past 3 months
  - Static signature weaknesses: searched for inadvertent “test” FPs
  - Nothing conclusive (6 suspicious events)

Is some of our telemetry also crafted?

- We are monitoring TOR-based telemetry
  - 1 out of 100,000 of our endpoints use TOR
  - TOR endpoints seem 4 times as infected as normal users
  - TOR endpoints send one tenth the rate of junk telemetry
  - Nothing found

The aftermath

Changes we've made



Industry
- Samples
- Meta-data
- Reputation
- Determinations

Collection

Customers
- Telemetry
- Samples

Big Data

samples, telemetry, reputation, determinations

Analysis

Auto-classification

Signature generation

Telemetry response

Issue	**Changes**
Signature generation using clean sections when signing crafted clean files	1) Auto-detect crafted clean 2) Sign only with static signatures
Static signatures used in automation had CRC collision weakness	1) Harden signature type to require SHA1 match
Potential poisoned telemetry	1) Anomaly monitoring
Not handling artificial escalations very well	1) Sample sharing requirements to include attestation of sourcing 2) Automation rules stop "credit" for detections 3) Issue awareness 4) Cross-vendor working group

Contaminating AV-Test

2 crafted files showed up in AV-Test’s August testing set

- 0xf019bceae867415dc2027b12b282486973759fa5
- 0x186f720f76bcd6fcc83055a64989ed45cd7b5d66

Andreas Marx investigated

- Vendors give to aggregators
- Aggregators share with testers and vendors
- Testers curate samples, but in the end, they trust vendor sources

Highlights need for vendor control of what is shared

- Artificially inflates the value of these files
- Encourages useless vendor detections
- Could lead to becoming a victim of weaponization

Industry Recommendations

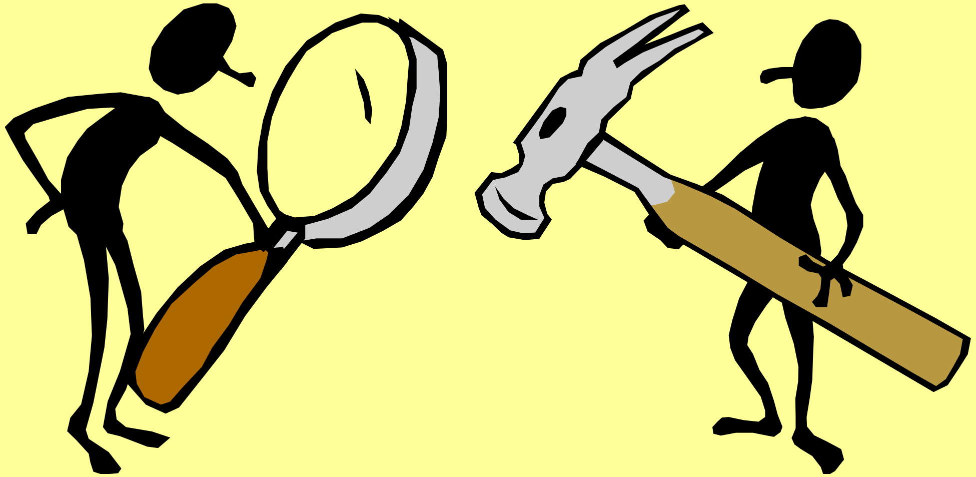
Exchanging unseen samples

- Causes artificial escalations and drives useless detections
- ***If your customers don't see it, don't exchange it***

Automated blind reliance on partner detections

- detections != determinations
- ***Rely only on vendor samples for vendor determinations***

More Industry Recommendations



Treat this as a serious threat

- Before somebody weaponizes you
- ***Find and fix your automation and signature weaknesses***

We need to work together

- ***Let's share crafted file/telemetry awareness and detection/mitigation techniques***

Microsoft