Behavioral Clustering of HTTP-based Malware and Signature Generation using Malicious Network Traces

Roberto Perdisci, Wenke Lee, Nick Feamster

USENIX NSDI 2010

What is Malware?

Malware = Malicious Software

Viruses
Worms
Trojans
Bots
Spyware
Adware
Scareware
...

What harm can Malware do?

Most modern cyber-crimes are carried out using Malware

Send SPAM
Phishing Infrastructure
Identity Theft
Steal Banking Credential
Denial of Service Attacks
...

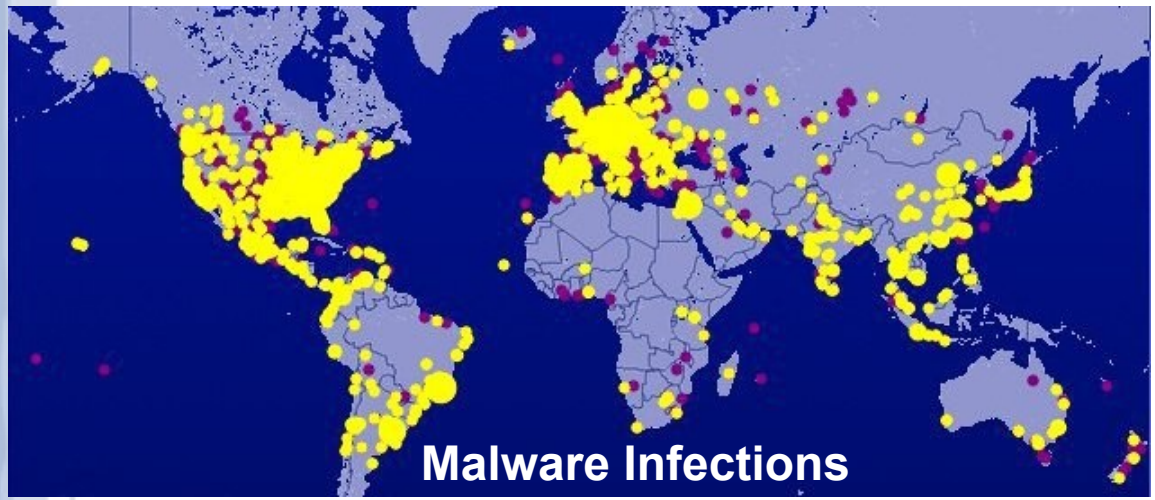
Malware is a global problem

One quarter of the Internet is infected by malware

Source: Vint Cerf, “father of the Internet”

The annual financial loss for US organizations amounts to hundreds of millions of dollars.

source: CSI/FBI Computer Crime and Security Survey (Dec. 2009)

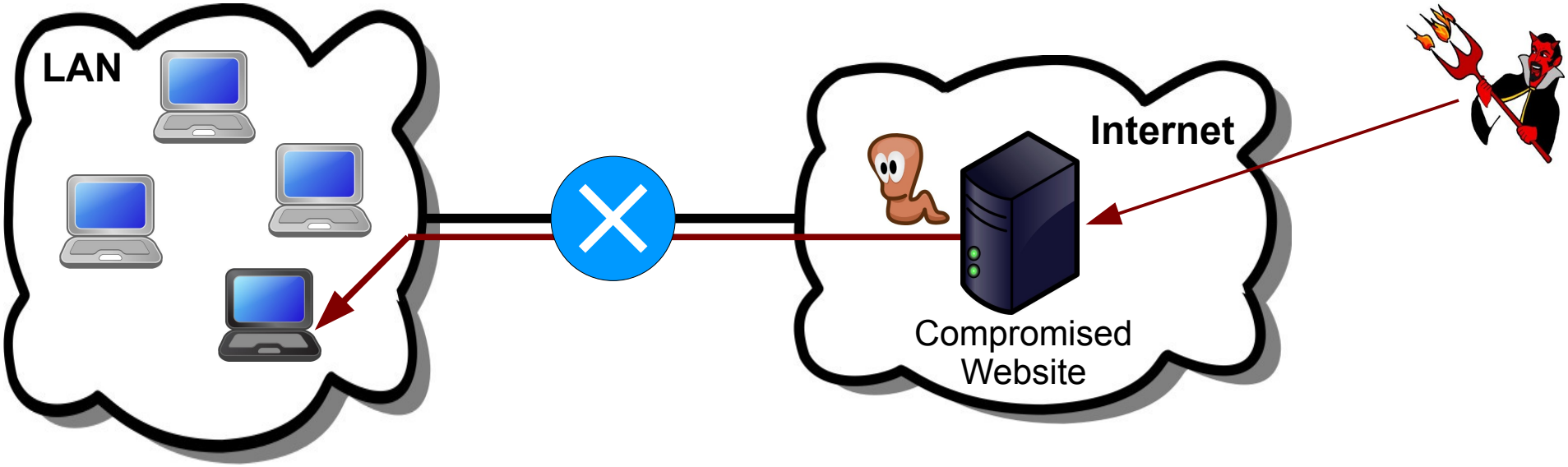


source: shadowserver.org

Operation Aurora

Malware Infection Vectors

"Drive-by" Malware Downloads



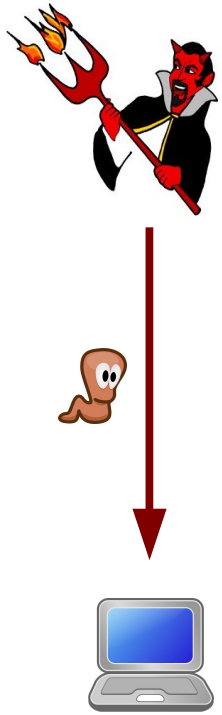
- Simply visiting a legitimate (compromised) Website can cause a malware infection!

Mawlare Infection Vectors

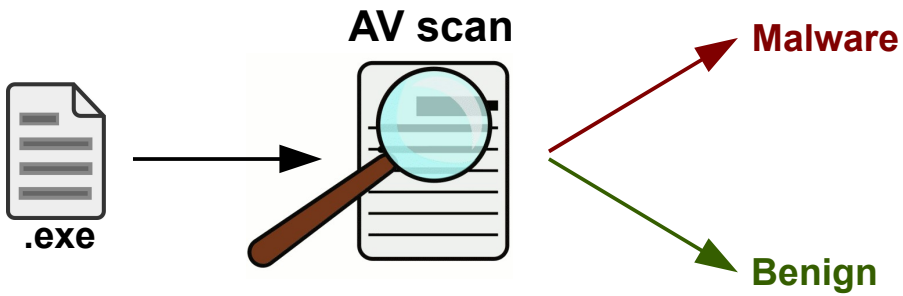
Social engineering attacks!

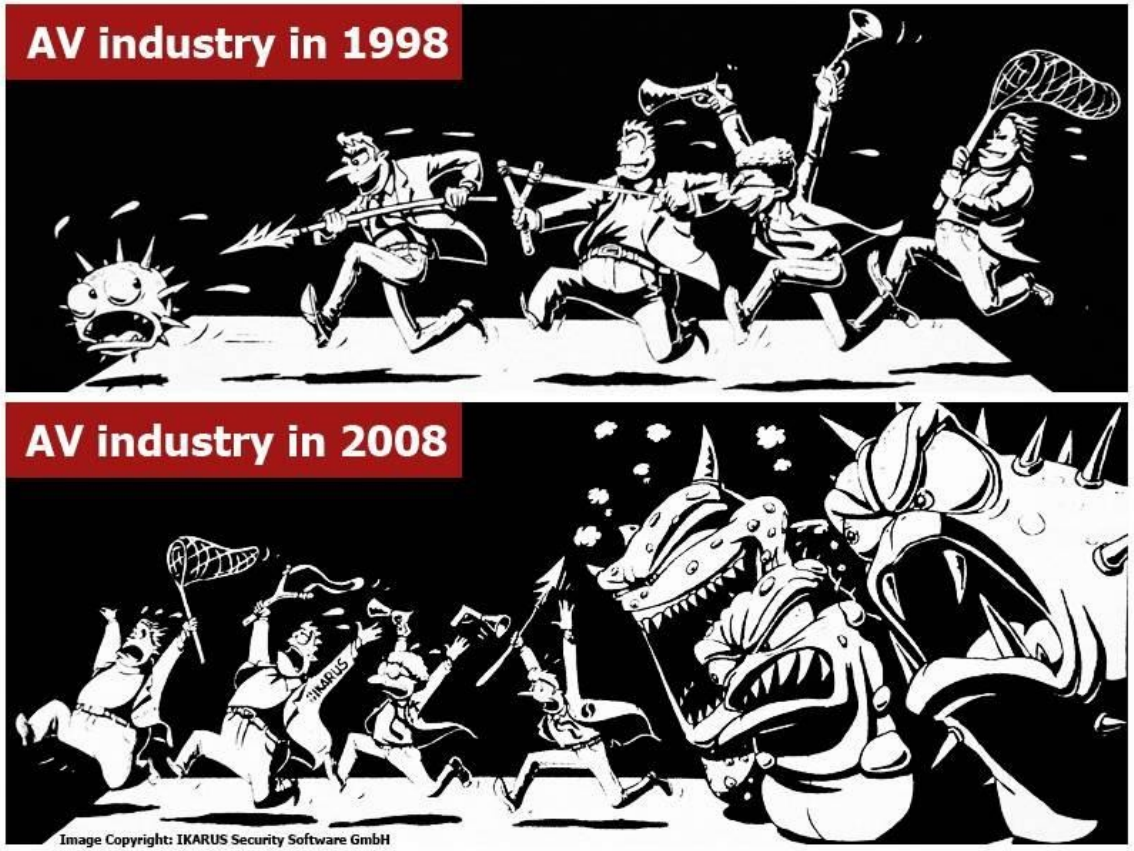
Infected external disk!

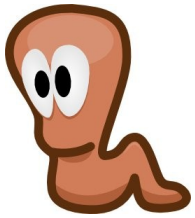
Direct *remote exploits*!



Traditional AVs are not enough!







Original Malware

Executable
Packing
(obfuscation)

Hidden
Malware

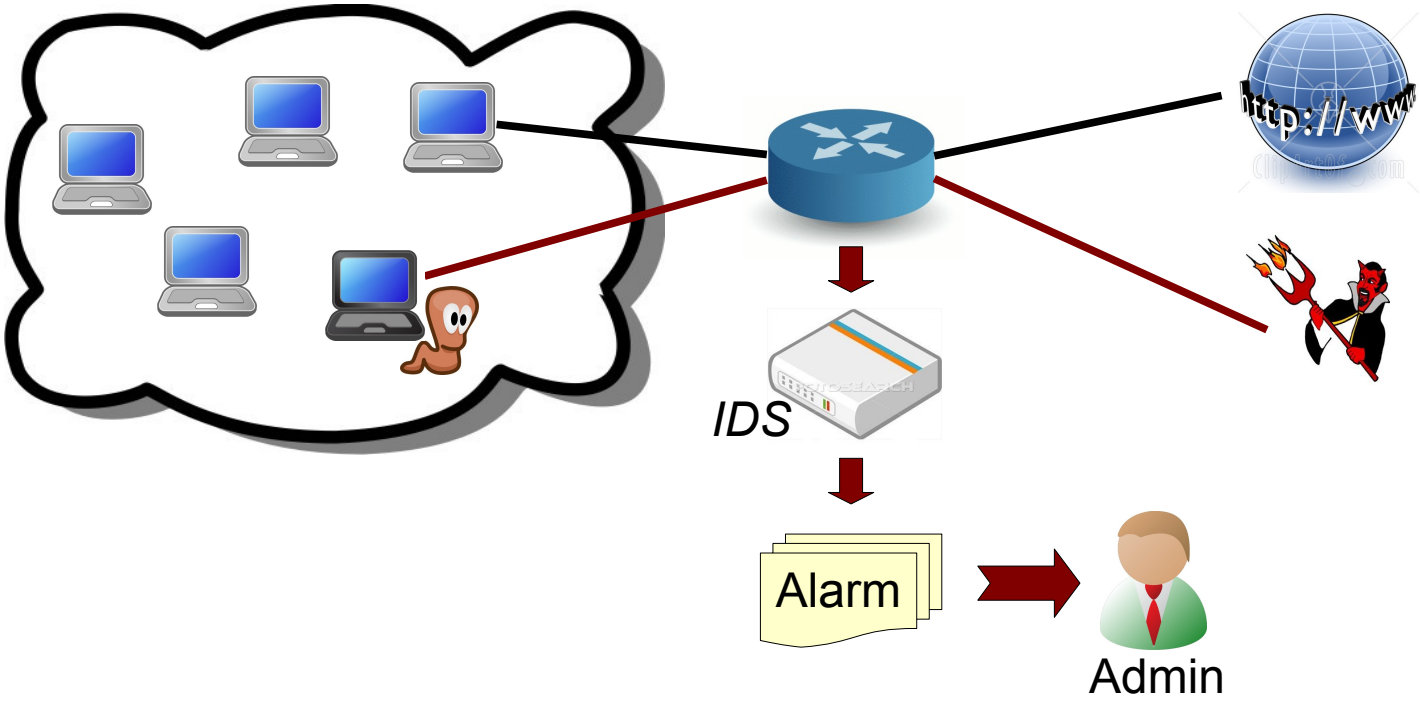
What can we do to secure the Internet?

- No silver bullet solution...
- Different approaches to tackle the problem from different points of views
  - Host-level solutions
  - Network-level solutions
  - Usable security
  - Educate users

Defense-in-Depth strategy

Our Approach

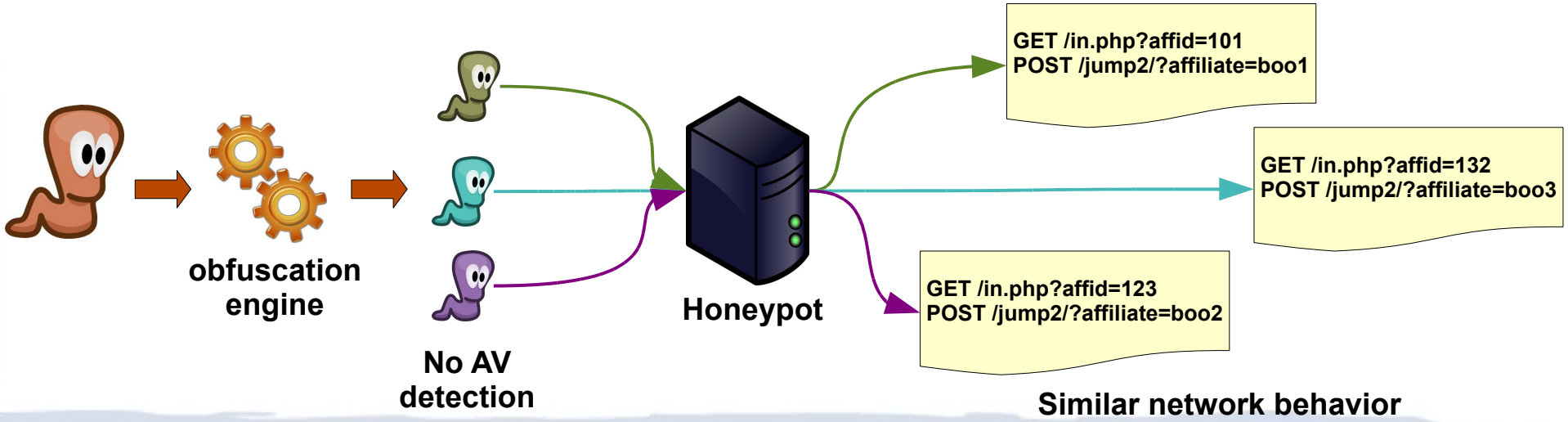
- Detect the Network Behavior of Malware



  - Complement existing host-based detection systems
  - Improve “coverage”

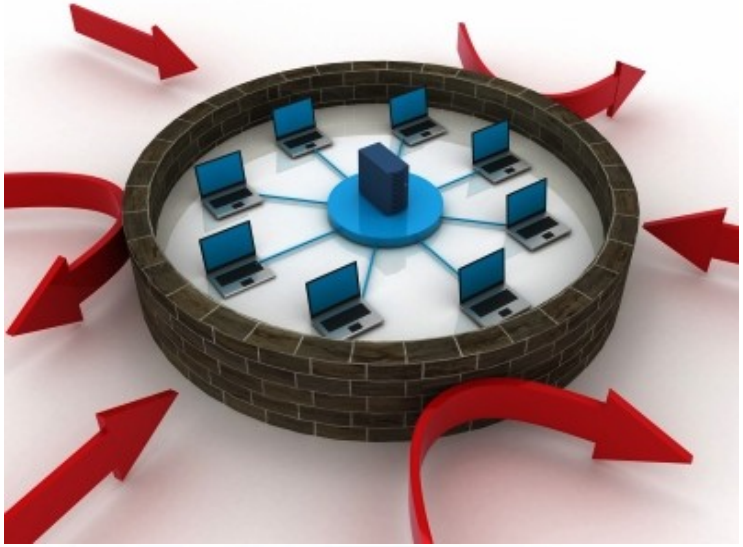
Key observations

- Most malware need a network connection to perpetrate malicious activities
  - ***Bots*** need to contact C&C server, send spam, etc...
  - ***Spyware*** need to exfiltrate private info
  - ***Trojan droppers*** need to download further malicious software ...
- Obfuscated variants of the same malware can evade AVs
  - When executed they generate ***similar malicious behavior***



Attractive Properties of Network-based Approach

- Monitor large number of machines with no overhead at the end host
  - Host-based malware behavior detection often requires costly VM monitoring
- Leverage existing *network perimeter* monitoring infrastructure
  - Enable detection of malware behavior



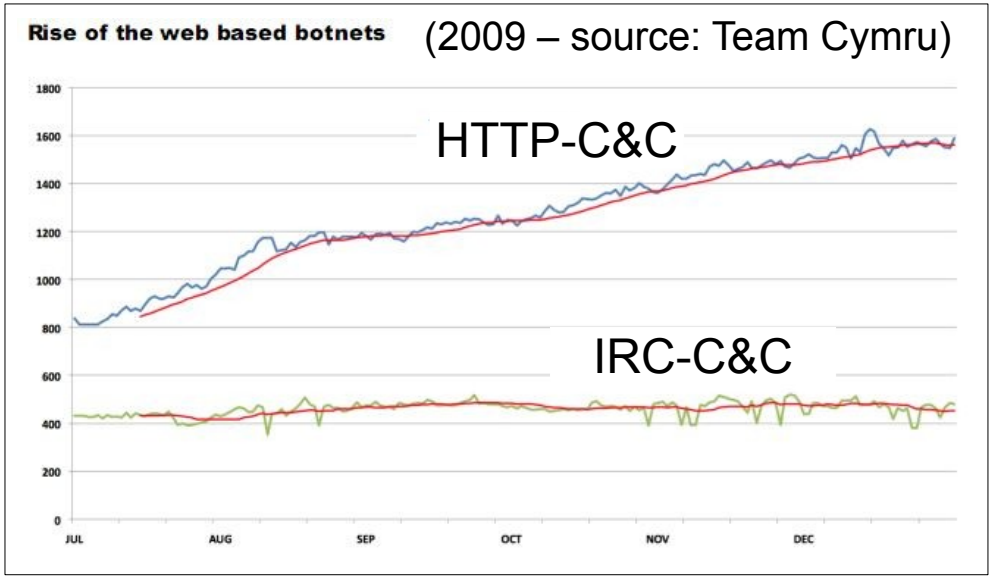
Challenges

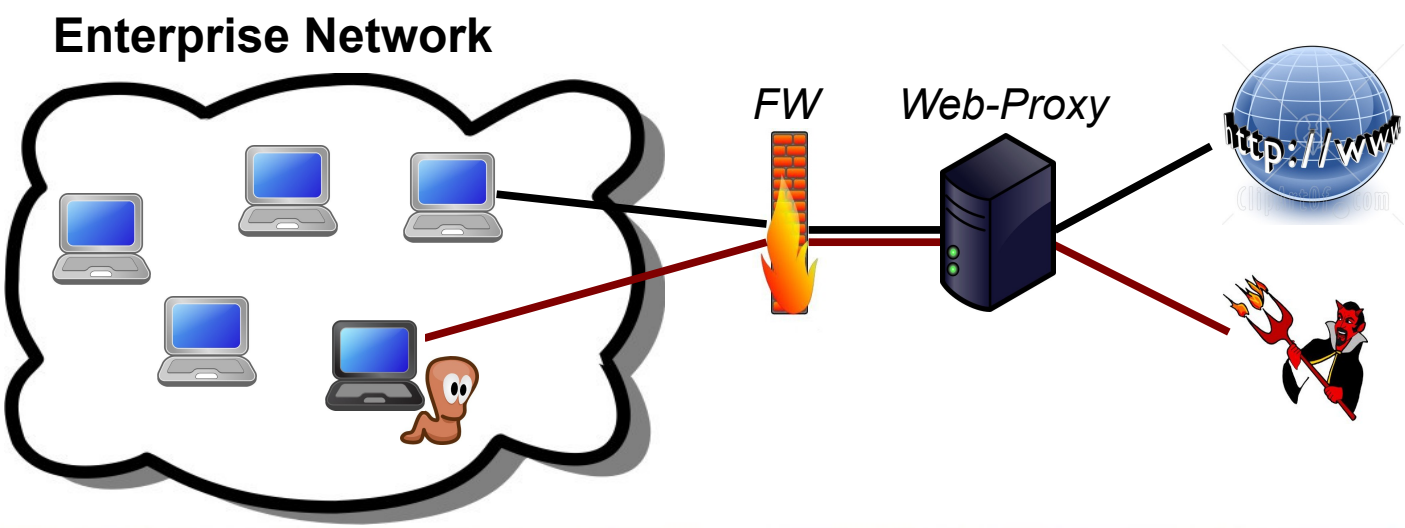
- Detecting malware traffic is hard
  - Many different types of malware
  - Different communication protocols
  - Malware can use legitimate protocols to communicate (e.g., HTTP)
  - Identify malware traffic among **very large** volumes of legitimate traffic

Find a needle in haystack!

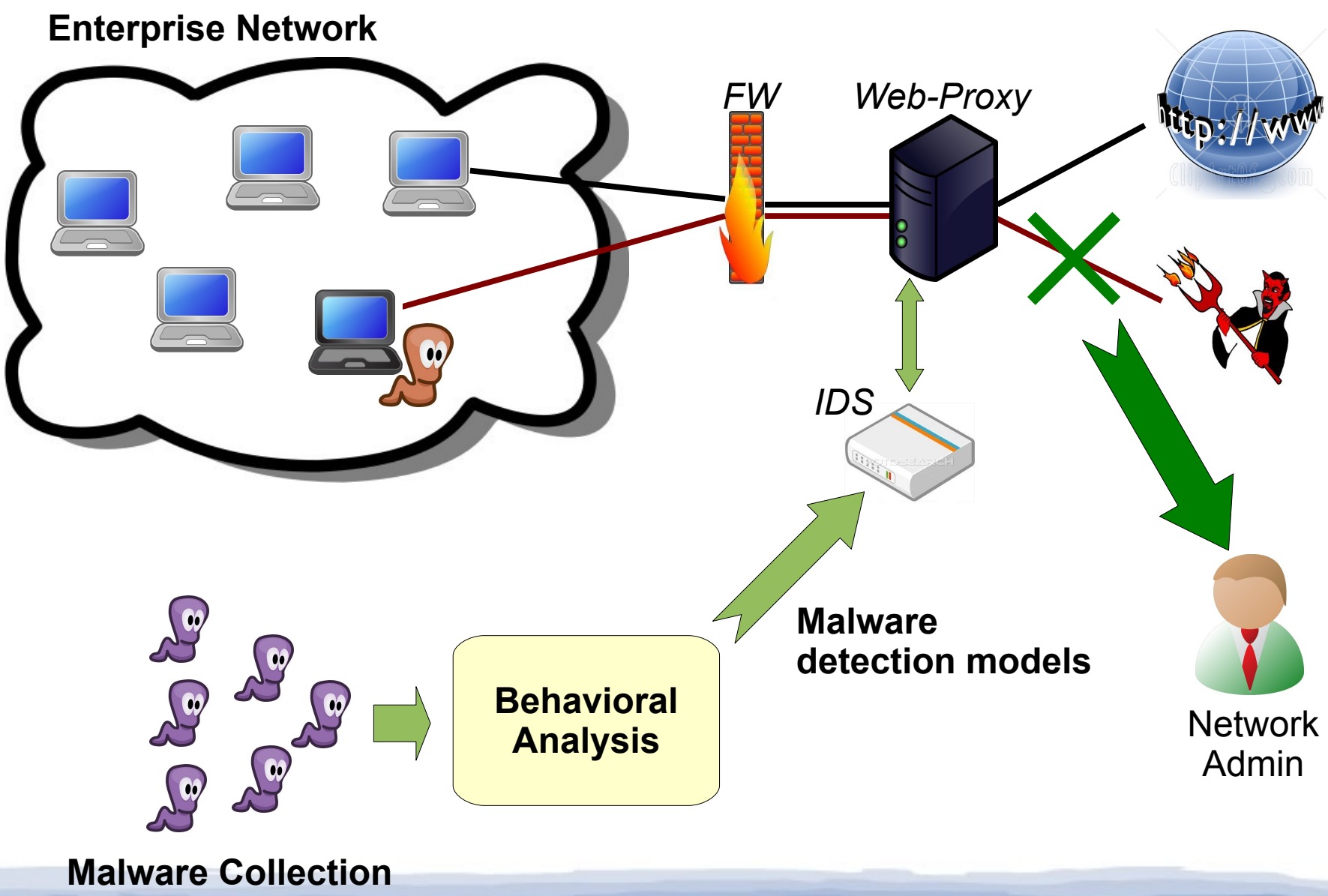
Web-based Malware

- Use HTTP protocol
- Bypass existing network defenses
  - Firewalls
- Web kits for malware control available

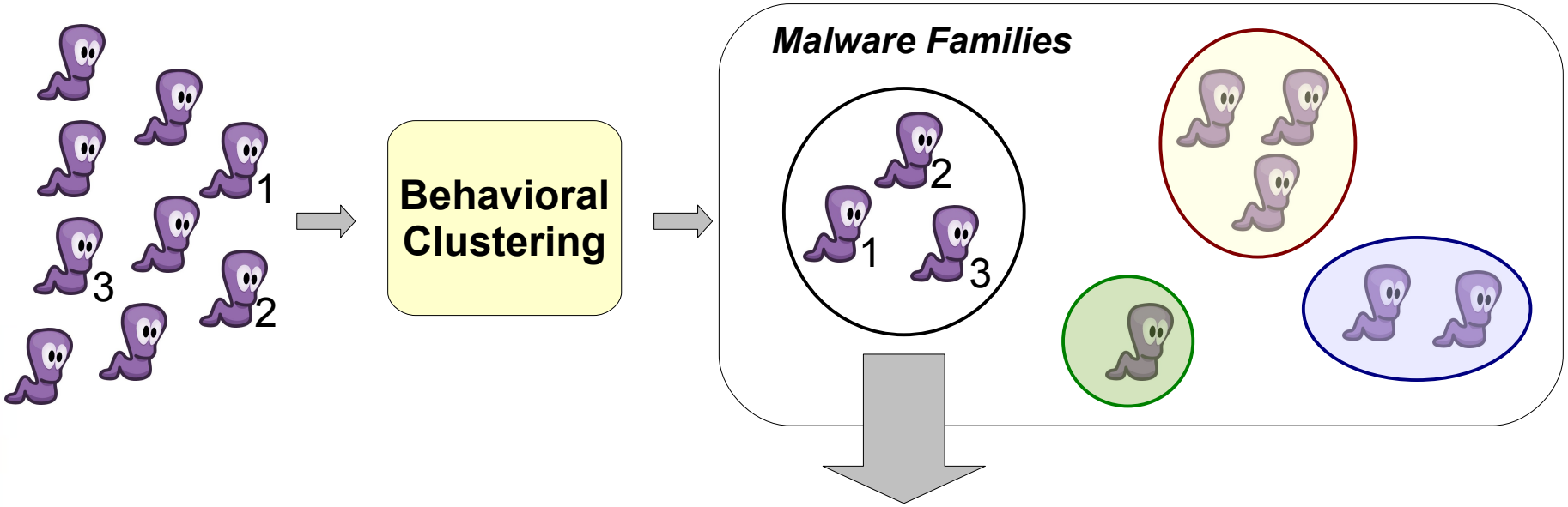




Detecting Web-based Malware



System Overview



Malware Traffic:

1 GET /in.php?affid=94901&url=5&win=Windows%20XP+2.0&sts=|US|1|6|4|1|284|0

2 GET /in.php?affid=43403&url=5&win=Windows%20XP+2.0&sts=

3 GET /in.php?affid=94924&url=5&win=Windows%20XP+2.0&sts=|US|1|6|8|1|184|0

Malware Detection Signature:

GET /in\.php\?affid=.*&url=5&win=Windows%20XP\+2\.0&sts=.*

Behavioral Malware Clustering

- System-level approaches
  - Based on *dynamic analysis*
    - Automated analysis of Internet malware [Bailey et al., RAID 2007]
    - Scalable malware clustering [Bayer et al., NDSS 2009]
  - Based on *static analysis*
    - Malware indexing using function-call graphs [Hu et al., CCS 2009]

source: honeyblog.org

Malware with similar network behavior may behave differently at the system level (and vice versa!)

v_1 = 076b81e8c6622e9c6a94426e8c2dfe33
AV Labels = Generic FakeAlert.h [McAfee];
TR/Dropper.Gen [Avira]

HTTP Traffic

```
[1249356561 192.168.14.2:1037 => 94.247.2.193:80]
POST /cgi-bin/generator HTTP/1.0
Content-Length: 45
[... DATA ...]

[1249356562 192.168.14.2:1038 => 94.247.2.193:80]
POST /extra.php HTTP/1.0
Content-Type:  application/x-www-form-urlencoded
Content-Length: 44
[... DATA ...]
```

File System Operations

```
Delete  c:\docume~1\admini~1\locals~1\temp\tmp1.tmp
Read    \\?\globalroot\systemroot\system32\msvcrt.dll
Write   c:\docume~1\admini~1\locals~1\temp\tmp1.tmp
```

v_2 = 7ee251d8d13ed32914a4e39740b91ae2
AV Labels = DR/PCK.Tdss.A.21 [Avira]

HTTP Traffic

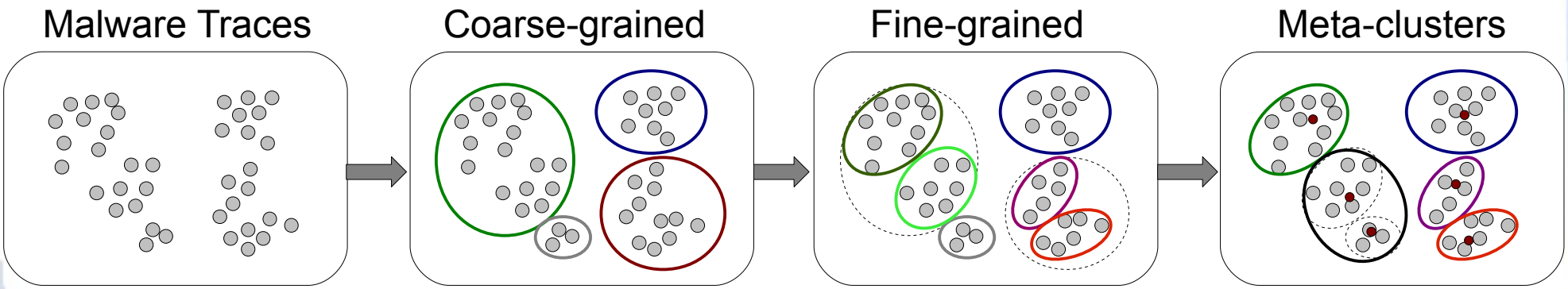
```
[1249345674 192.168.12.2:1034 => 94.247.2.193:80]
POST /cgi-bin/generator HTTP/1.0
Content-Length: 45
[... DATA ...]

[1249345674 192.168.12.2:1038 => 94.247.2.193:80]
POST /extra.php HTTP/1.0
Content-Type:  application/x-www-form-urlencoded
Content-Length: 44
[... DATA ...]
```

File System Operations

```
Delete  c:\docume~1\admini~1\locals~1\temp\tmp4.tmp
Delete  c:\docume~1\admini~1\locals~1\temp\tmp5.tmp
Write   c:\docume~1\admini~1\locals~1\temp\tmp5.tmp
Read    \\?\globalroot\systemroot\system32\advapi32.dll
Write   c:\docume~1\admini~1\locals~1\temp\tmp4.tmp
Write   c:\docume~1\admini~1\locals~1\temp\nso3.tmp\modern-header.bmp
Delete  c:\docume~1\admini~1\locals~1\temp\nso3.tmp
Write   c:\docume~1\admini~1\locals~1\temp\matrix329411.exe
Read    (MALWARE_PATH)
Delete  c:\docume~1\admini~1\locals~1\temp\nsc1.tmp
```

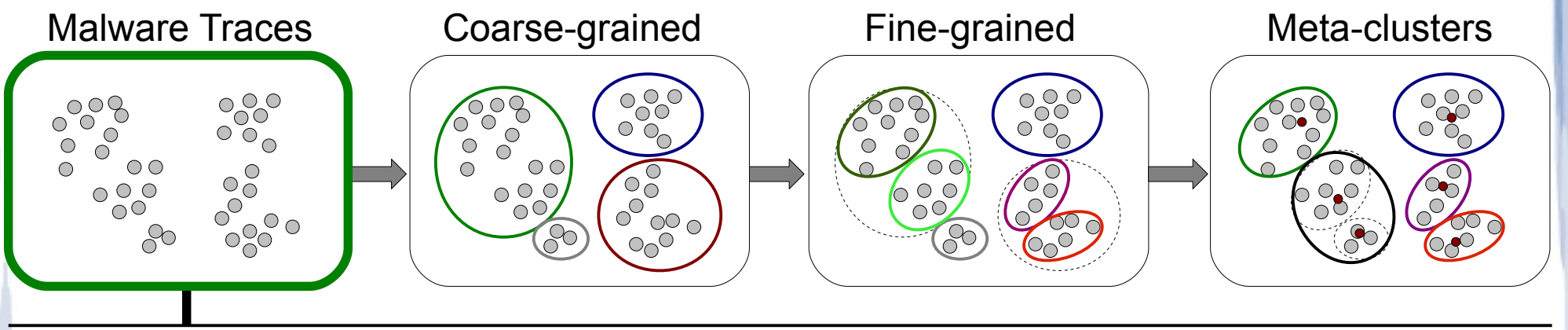
- Our approach
  - Focus on **network-level behavior**
  - Clusters and related signatures should be independent from specific server IPs or domain names
  - Better *network-based* malware detection signatures compared to using host-level approaches

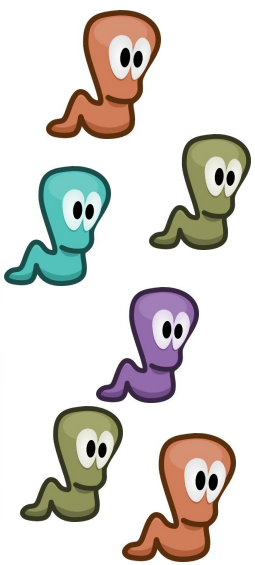
Network Behavioral Clustering

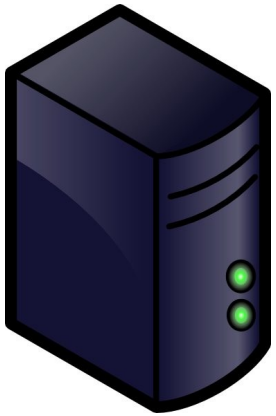


- ***Three-step*** clustering refinement process
- Good trade-off between ***efficiency*** and ***accuracy***
- ***High-quality clusters*** are essential to extract good signatures

Network Behavioral Clustering





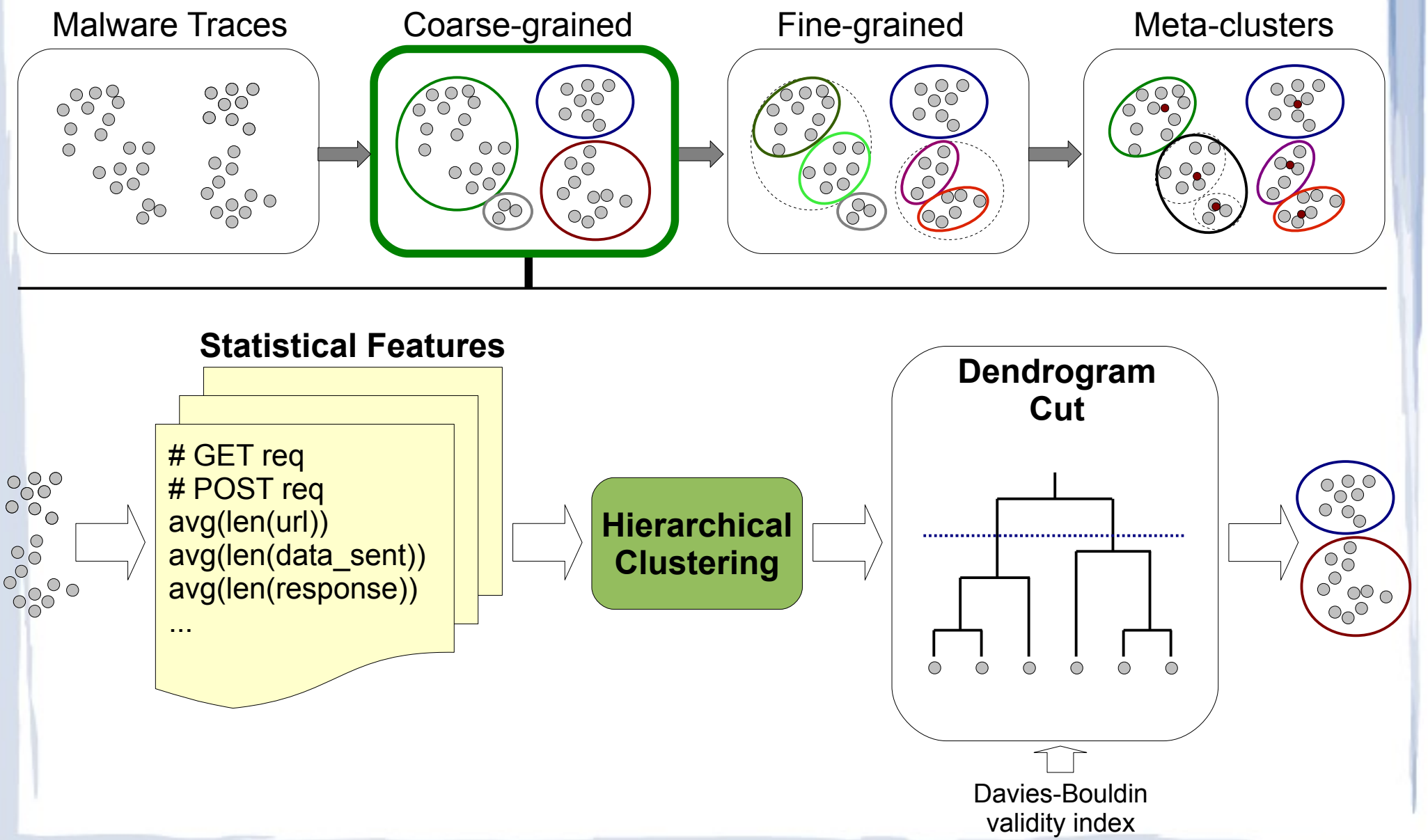


Honeypot

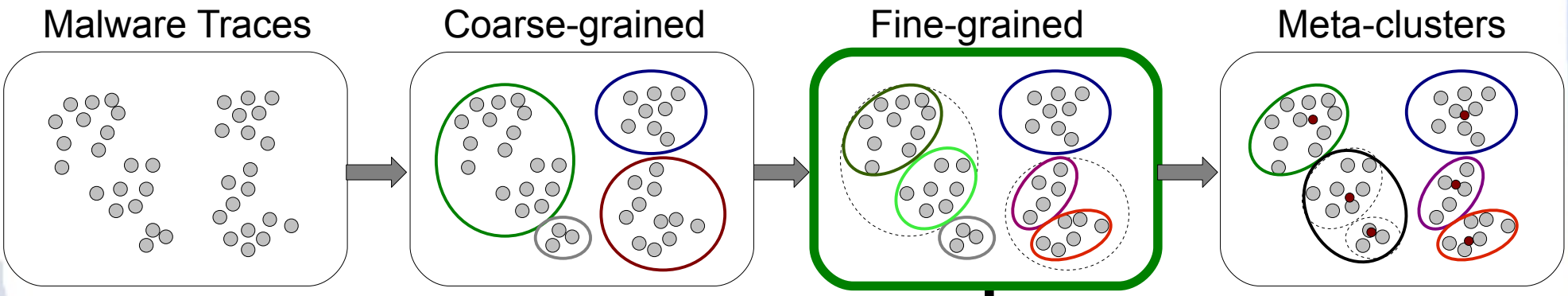
GET /bins/int/9kgen_up.int?fxp=6d HTTP/1.1
User-Agent: Download
Host: X1569.nb.host192-168-1-2.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Connection: close
Server: Yaws/1.68 Yet Another Web Server
Date: Mon, 15 Mar 2010 11:47:11 GMT
Content-Length: 573444
Content-Type: application/octet-stream

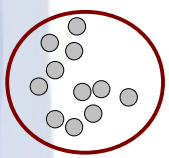
Network-level Clustering



Network-level Clustering

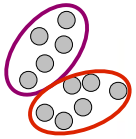


Structural Features



GET /in.php?affid=94900
GET /bins/int/9kgen_up.int?fxp=6dc23
POST /jump2/?affiliate=boo1
POST /trf?q=Keyword1&bd=-5%236

Hierarchical Clustering ... ***DC***



Malware Trace $\mathcal{M}_1$

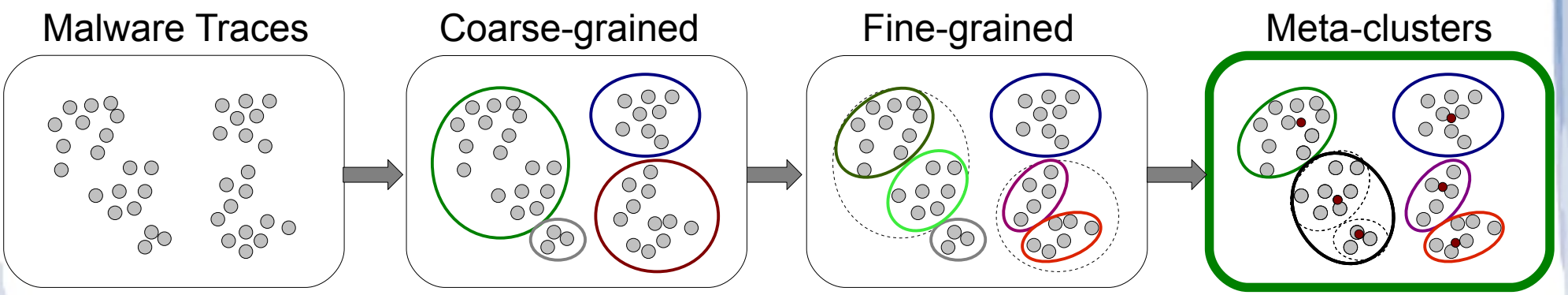
GET /in.php?affid=94900
GET /bins/int/9kgen_up.int?fxp=6dc23
POST /jump2/?affiliate=boo1
POST /trf?q=Keyword1&bd=-5%236

$d(\mathcal{M}_1,\mathcal{M}_2)$

Malware Trace $\mathcal{M}_2$

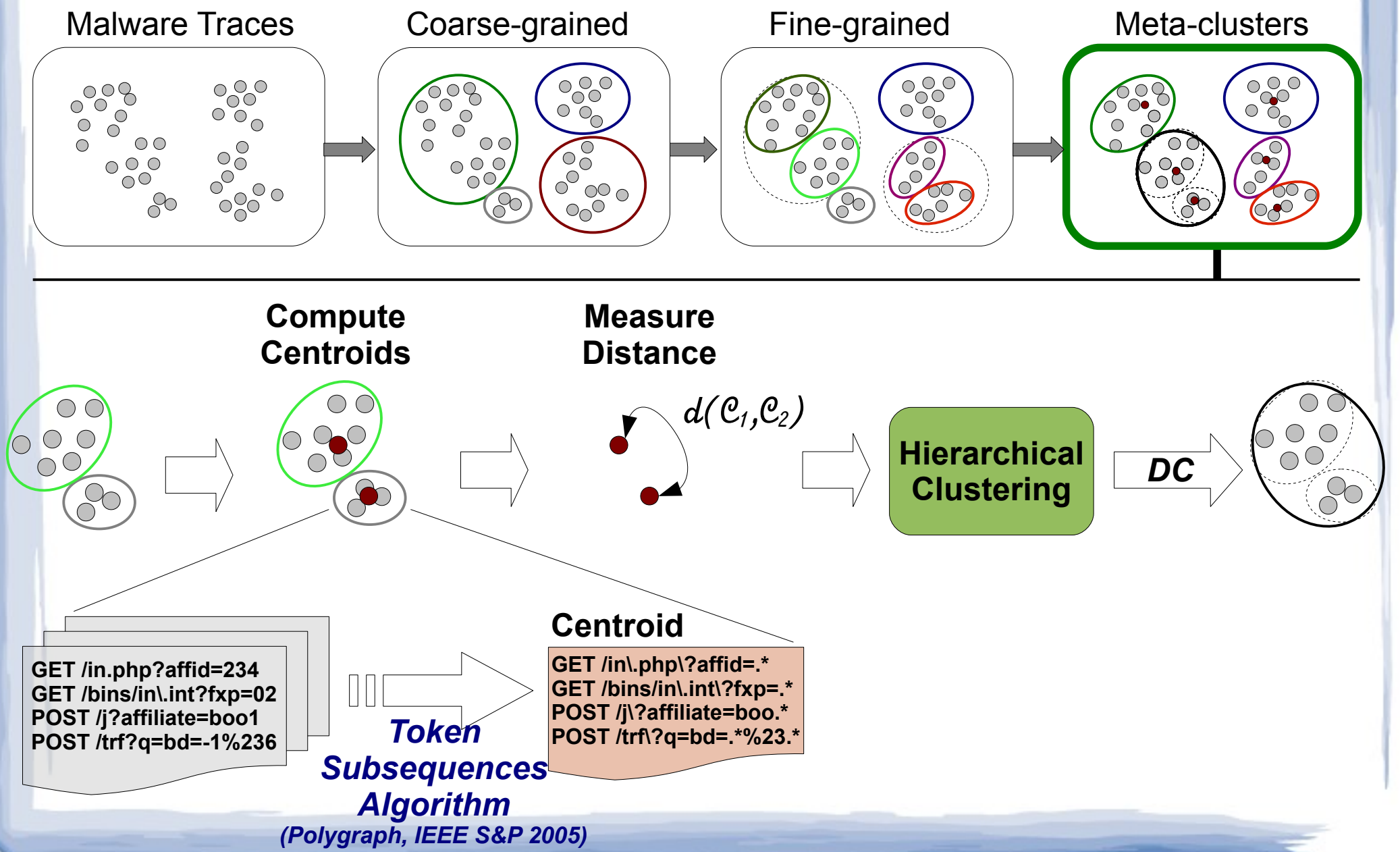
GET /index.php?v=1.3&os=WinXP
GET /kgen/config.txt
POST /bots/command.php?a=6.6.6.6
POST /attack.php?ip=10.0.1.2&c=dos

Network-level Clustering



- ***Meta-clustering*** recovers from possible mistakes made in previous steps
- Improves overall **quality** of malware clusters and malware detection models

Network-level Clustering



Evaluating the Quality of Clusters

- Hard task, no standard way to do it...

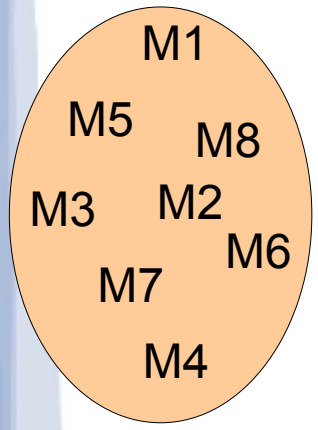
  Clustering can be viewed as an unsupervised learning task, and analyzing the validity of the clustering results is intrinsically hard.

  Cluster validity analysis often involves the use of a subjective criterion of optimality.

- Previous work [Bayer et al., NDSS 2009]
  - compare to AV family labels
  - (semi-)manual *reference clustering*
  - Precision and Recall

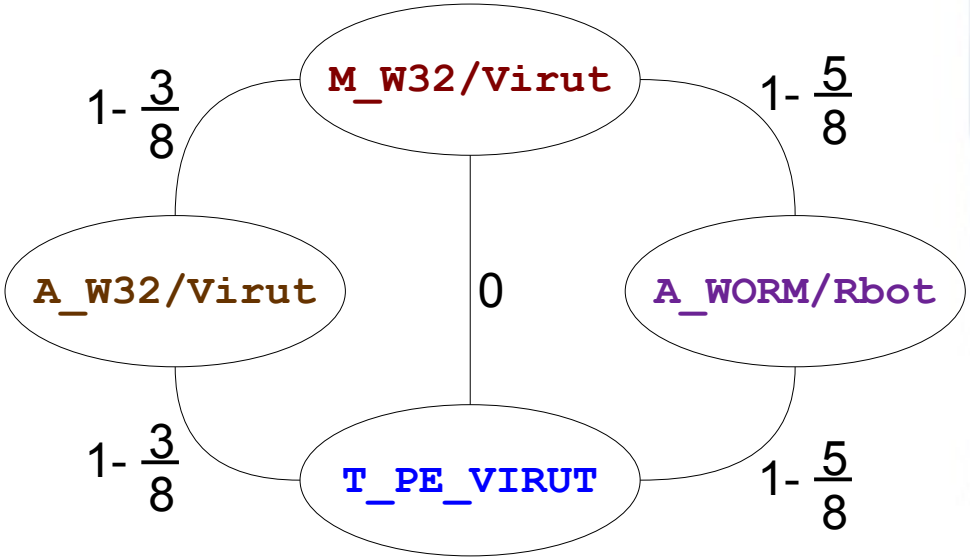
Our Cluster Validity Analysis

Malware Cluster



	McAfee	Avira	Trend Micro
M1 :	W32/Virut.gen	WORM/Rbot.50176.5	PE_VIRUT.D-1
M2 :	W32/Virut.gen	WORM/Rbot.50176.5	PE_VIRUT.D-2
M3 :	W32/Virut.gen	W32/Virut.Gen	PE_VIRUT.D-4
M4 :	W32/Virut.gen	W32/Virut.X	PE_VIRUT.XO-2
M5 :	W32/Virut.gen	WORM/Rbot.50176.5	PE_VIRUT.D-2
M6 :	W32/Virut.gen	W32/Virut.H	PE_VIRUT.NS-2
M7 :	W32/Virut.gen	WORM/Rbot.50176.5	PE_VIRUT.D-2
M8 :	W32/Virut.gen	WORM/Rbot.50176.5	PE_VIRUT.D-1

AV-Label Graph



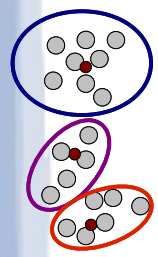
Cohesion Index

$$\mathrm{c}(\mathbf{C}_i) = 1 - \frac{1}{\gamma}\frac{2}{n \cdot v(n \cdot v - 1)} \sum_{l_1 < l_2} \delta_{l_1, l_2}$$

Separation Index

$$\mathrm{s}(\mathbf{C}_i, \mathbf{C}_j) = \frac{1}{\gamma} \mathrm{avg}_{k,h}\{\Delta(V_k^{(i)}, V_h^{(j)})\}$$

Signature Generation and Pruning



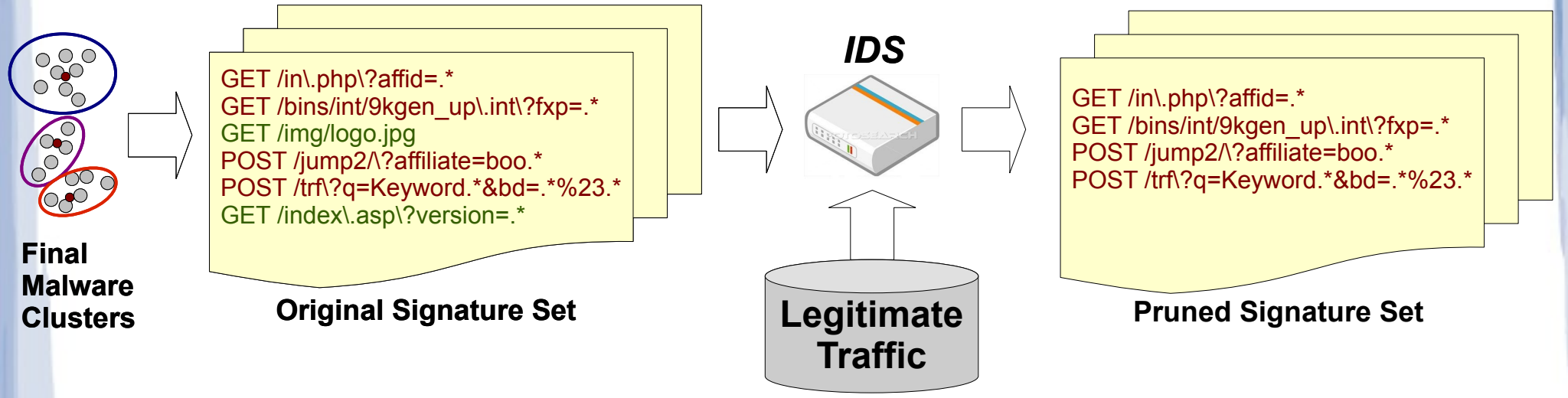
Final Malware Clusters

```
GET /in\.php\?affid=.*
GET /bins/int/9kgen_up\.int\?fxp=.*
GET /img/logo.jpg
POST /jump2/\?affiliate=boo.*
POST /trf\?q=Keyword.*&bd=.*%23.*
GET /index\.asp\?version=.*
```

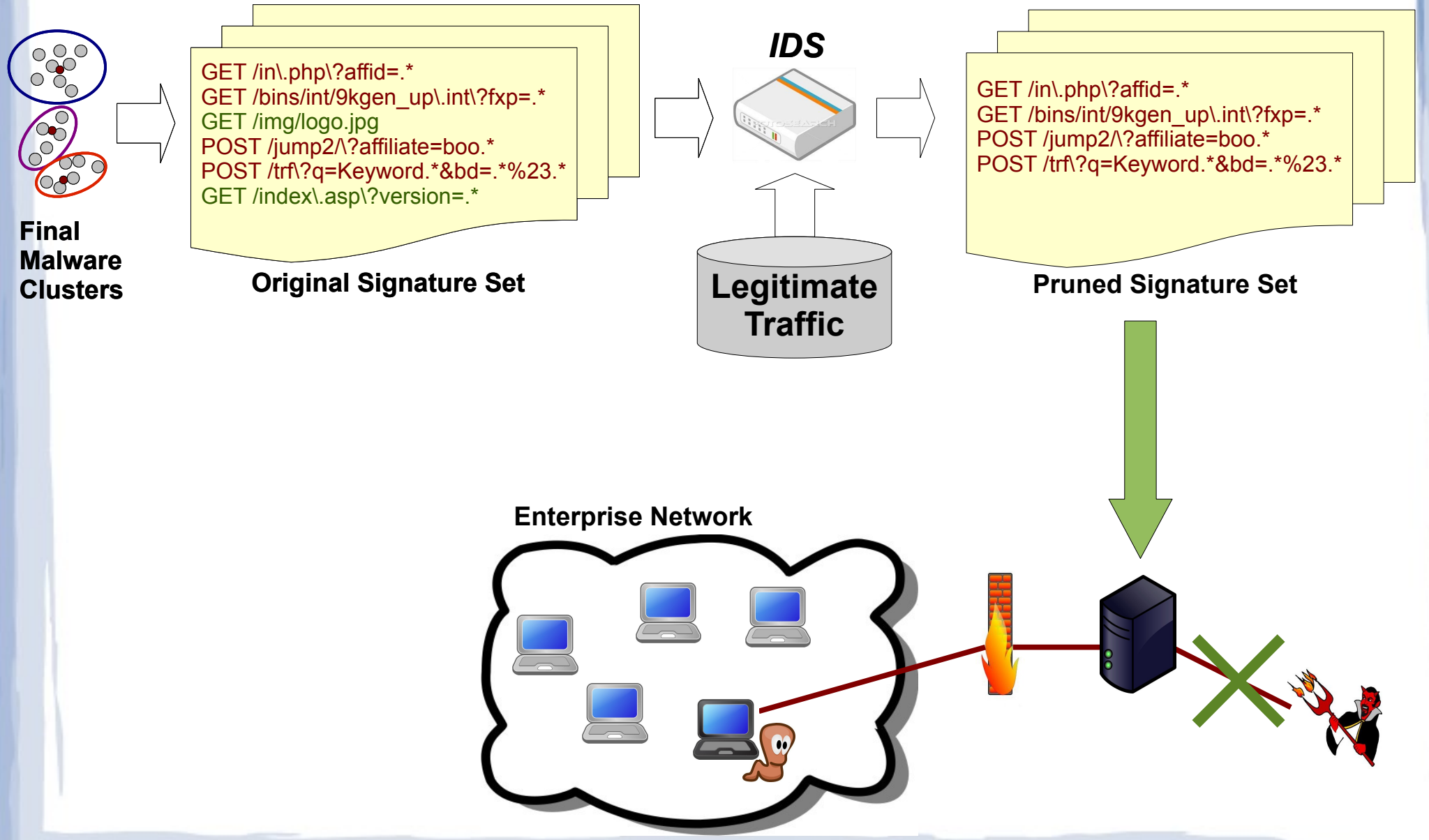
Original Signature Set

Token Subsequences Algorithm
(Polygraph, IEEE S&P 2005)

Signature Generation and Pruning



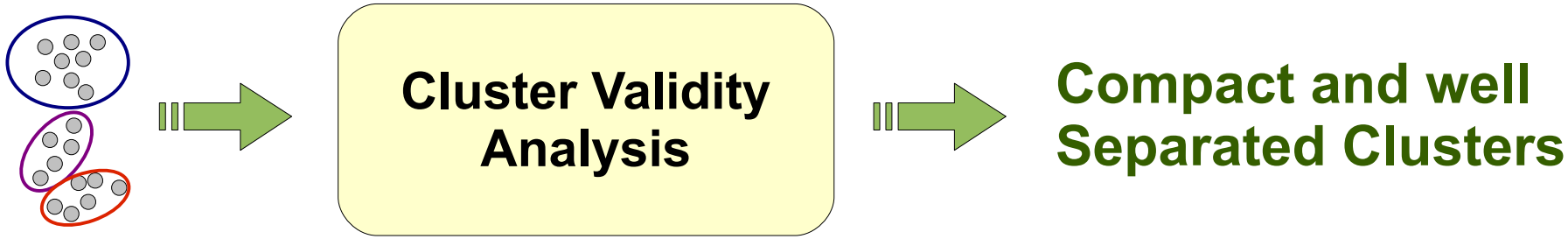
Signature Generation and Pruning



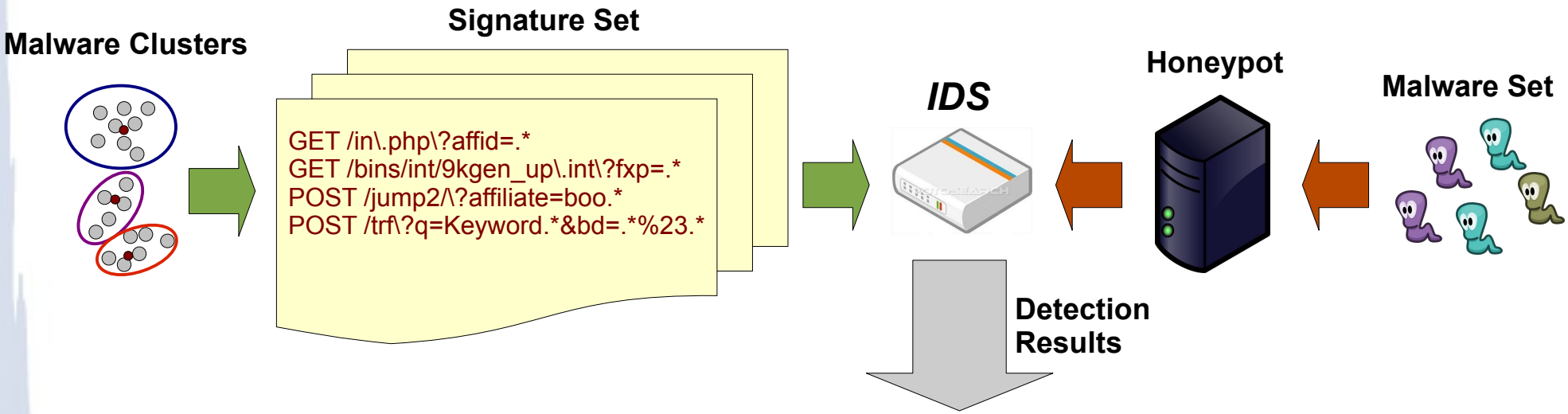
Experimental Results

- Malware Dataset
  - 6 months of malware collection (Feb-Jul 2009)
  - **~25k** distinct *real-world* malware samples
- Clustering Results

Dataset	Samples	Malware Families	Modeled Samples	Signatures	Time
Feb-2009	4,758	234	3,494	446	~8h



Malware Detection Results



Detection Test on All Samples

	Feb09	Mar09	Apr09	May09	Jun09	Jul09
Sig. Feb09	**85.9%**	**50.4%**	47.8%	27.0%	21.7%	23.8%

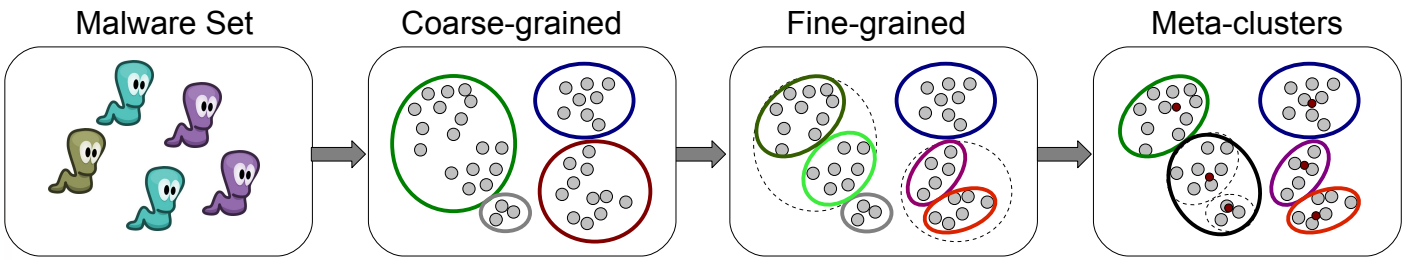
Detection Test on Malware undetected by commercial AVs

	Feb09	Mar09	Apr09	May09	Jun09	Jul09
Sig. Feb09	**54.8%**	**52.8%**	29.4%	6.1%	3.6%	4.0%

Sig. Feb09 **No False Alerts** → Tested on 12M legitimate HTTP queries

Comparison with other approaches

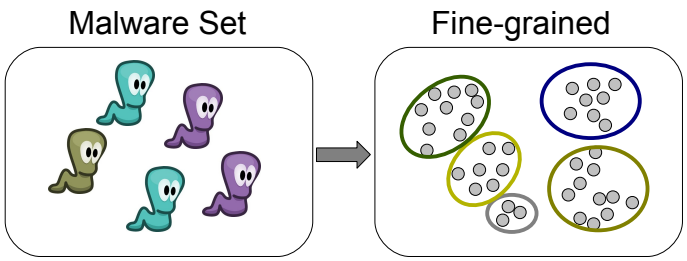
Signature extracted from reduced malware set of ~2k malware samples



Malware Detection

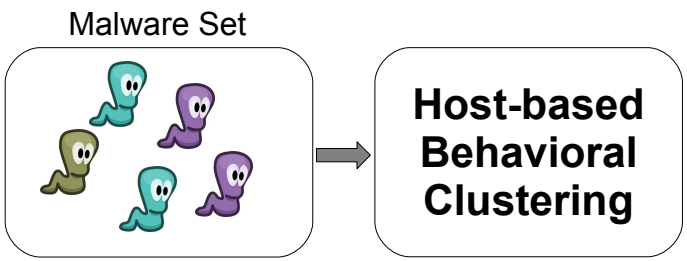
Feb09	Mar09
78.6%	48.9%

Using only fine-grained clustering



Feb09	Mar09
60.1%	35.1%

Using approach proposed in [Bayer et al. NDSS 2009]



Feb09	Mar09
56.9%	33.9%

Real-World Signature Deployment

- Deployed in large enterprise network
  - ~ 2k-3k active nodes
  - 4 days of testing
  - ~2k distinct signatures
- Findings
  - **25** machines infected by ***spyware***
  - **19** machines infected by ***scareware*** (fake AVs)
  - **1** ***bot***-compromised machine
  - **1** machine compromised by ***banker trojan***

Detecting Zero Day Malware

- *Guilty by association* policy
  - EXEs that cluster with known malware are bad!

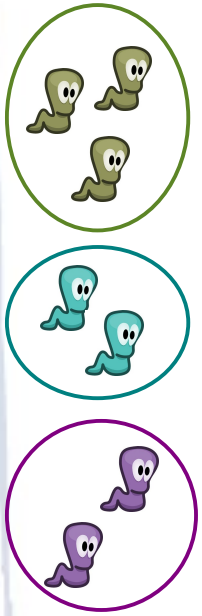
cluster_id	md5	scanner_name	virus_name
80594	102244534227faa399703abead1ba9e9		
80594	6fbe18753c9ce480e9b8b7d4cb1909d8		
80594	4b6ce9cef117ac18eba1d9c07969a374		
80594	9ce21ca99ad7bc2f6be266786bfd44cd	avira	TR/Crypt.XPACK.Gen
80594	9ce21ca99ad7bc2f6be266786bfd44cd	symantec	Trojan.FakeAV
80594	9ce21ca99ad7bc2f6be266786bfd44cd	trend	TROJ_OFICLA.SM
80594	cb5689c4982f05ca1027472ecffd3dff	avira	TR/Crypt.XPACK.Gen
80594	cb5689c4982f05ca1027472ecffd3dff	symantec	Trojan.FakeAV
80594	cb5689c4982f05ca1027472ecffd3dff	trend	TROJ_OFICLA.SM

md5	url	host	user_agent
9ce21ca99ad7bc2f6be266786bfd44cd	GET /loads.php?code=	domen-zaibisya.com	wget
9ce21ca99ad7bc2f6be266786bfd44cd	GET /firewall.dll	domen-zaibisya.com	wget
9ce21ca99ad7bc2f6be266786bfd44cd	GET /cgi-bin/ware.cgi?adv=	domen-zaibisya.com	wget
9ce21ca99ad7bc2f6be266786bfd44cd	GET /cgi-bin/get.pl?l=	kakleglo2335.com	wget

md5	url	host	user_agent
6fbe18753c9ce480e9b8b7d4cb1909d8	GET /loads.php?code=	get-money-now.net	wget
6fbe18753c9ce480e9b8b7d4cb1909d8	GET /firewall.dll	get-money-now.net	wget
6fbe18753c9ce480e9b8b7d4cb1909d8	GET /cgi-bin/ware.cgi?adv=	get-money-now.net	wget
6fbe18753c9ce480e9b8b7d4cb1909d8	GET /cgi-bin/get.pl?l=	mamapapalol.com	wget

Technology Transfer

- **HTTPrecon**



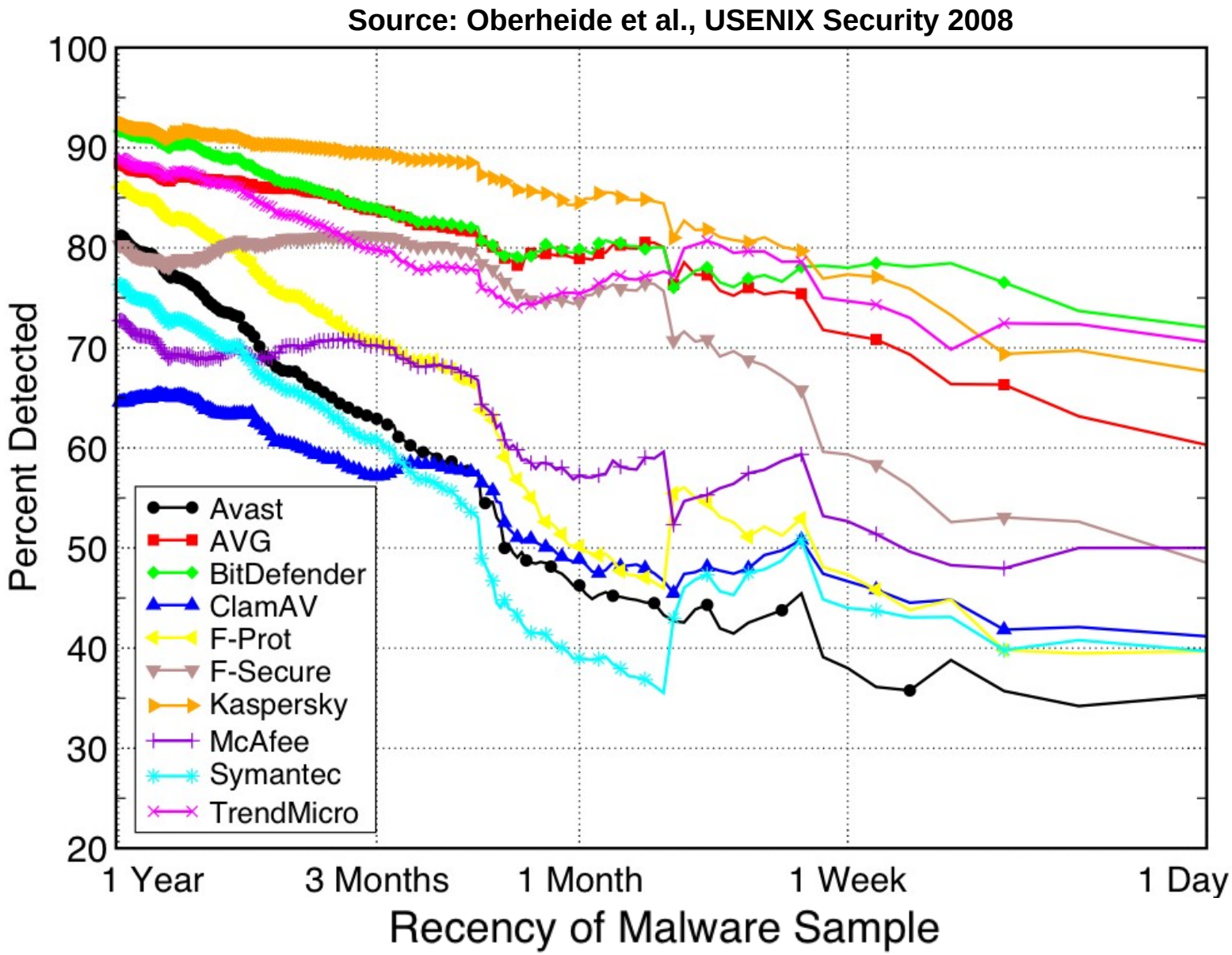
  - Filed for US patent
  - Deployed **since January 2010**
  - Analysis of **~10k** distinct malware samples/month
  - Used to categorize groups of bot-malware into distinct ***botnets***
  - Finds previously unknown compromised assets
  - Results analyzed on a daily bases by Damballa's Threat Analysts

Future Research

- More efficient clustering process
- Generalize to all kinds of malware, not only HTTP-based
- Automatic generation of "signature-less" statistical detection models

Appendix

AV malware detection stats



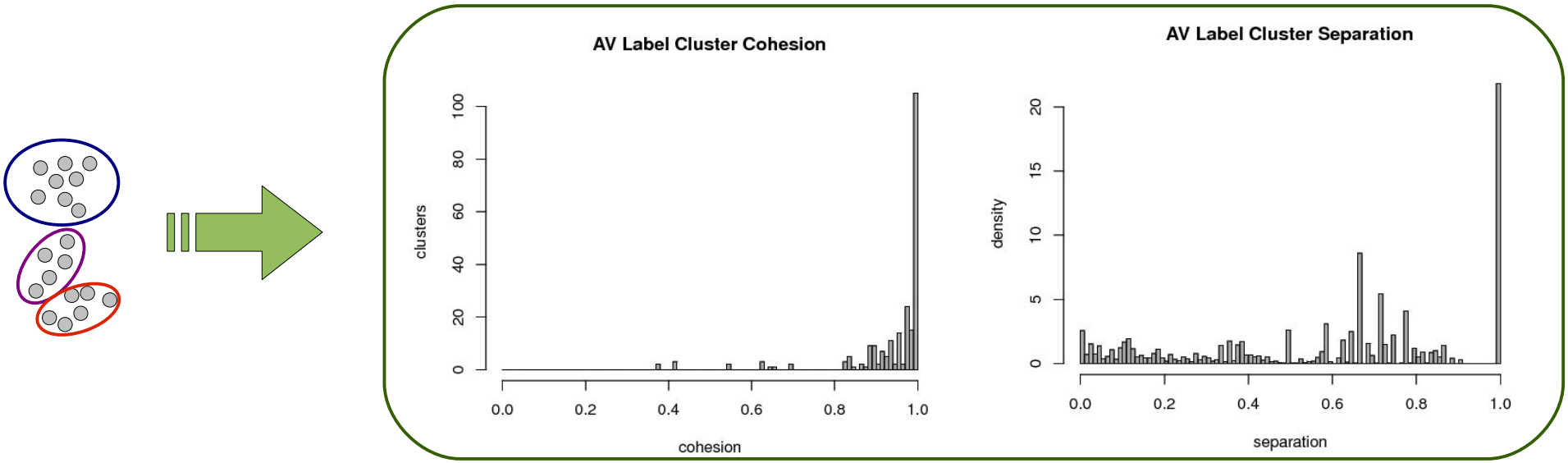
Experimental Results

6 months malware collection → over 25k distinct samples

	Malware Samples			Number of Clusters			Processing Time		
dataset	**samples**	**undetected by all AVs**	**undetected by best AV**	**coarse**	**fine**	**meta**	**coarse**	**fine**	**meta+sig**
Feb09	4,758	208 (4.4%)	327 (6.9%)	2,538	2,660	1,499	34min	22min	6h55min
Mar09	3,563	252 (7.1%)	302 (8.6%)	2,160	2,196	1,779	19min	3min	1h3min
Apr09	2,274	142 (6.2%)	175 (7.7%)	1,325	1,330	1,167	8min	5min	28min
May09	4,861	997 (20.5%)	1,127 (23.2%)	3,339	3,423	2,593	56min	8min	2h52min
Jun09	4,677	1,038 (22.2%)	1,164 (24.9%)	3,304	3,344	2,537	57min	3min	37min
Jul09	5,587	1,569 (28.1%)	1,665 (29.8%)	3,358	3,390	2,724	1h5min	5min	2h22min

Compact and well Separated Clusters

Cluster Validity Analysis



Experimental Results

Malware Detection rate (all samples)

	Feb09	*Mar09*	*Apr09*	*May09*	*Jun09*	*Jul09*
Sig_Feb09	85.9%	50.4%	47.8%	27.0%	21.7%	23.8%
Sig_Mar09	-	64.2%	38.1%	25.6%	23.3%	28.6%
Sig_Apr09	-	-	63.1%	26.4%	27.6%	21.6%
Sig_May09	-	-	-	59.5%	46.7%	42.5%
Sig_Jun09	-	-	-	-	58.9%	38.5%
Sig_Jul09	-	-	-	-	-	65.1%

Detects significant fraction of current and *future* malware variants

False Positives as measured on 12M legitimate HTTP requests from 2,010 clients

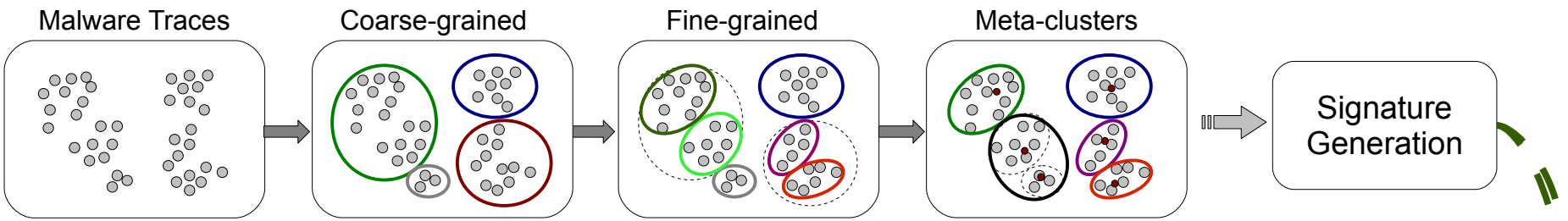
	Sig_Feb09	*Sig_Mar09*	*Sig_Apr09*	*Sig_May09*	*Sig_Jun09*	*Sig_Jul09*
FP rate	0% (0)	$3 \cdot 10^{-4}$% (38)	$8 \cdot 10^{-6}$% (1)	$5 \cdot 10^{-5}$% (6)	$2 \cdot 10^{-4}$% (26)	10^{-4}% (18)
Distinct IPs	0% (0)	0.3% (6)	0.05% (1)	0.2% (4)	0.4% (9)	0.3% (7)
Processing Time	13 min	10 min	6 min	9 min	12 min	38 min

"Zero-Day" Malware Detection rate

	Feb09	*Mar09*	*Apr09*	*May09*	*Jun09*	*Jul09*
Sig_Feb09	54.8%	52.8%	29.4%	6.1%	3.6%	4.0%
Sig_Mar09	-	54.1%	20.6%	5.0%	3.1%	5.4%
Sig_Apr09	-	-	41.9%	5.8%	3.8%	5.2%
Sig_May09	-	-	-	66.7%	38.8%	16.1%
Sig_Jun09	-	-	-	-	48.9%	21.8%
Sig_Jul09	-	-	-	-	-	62.9%

Complements traditional AV detection systems

Comparison with other approaches

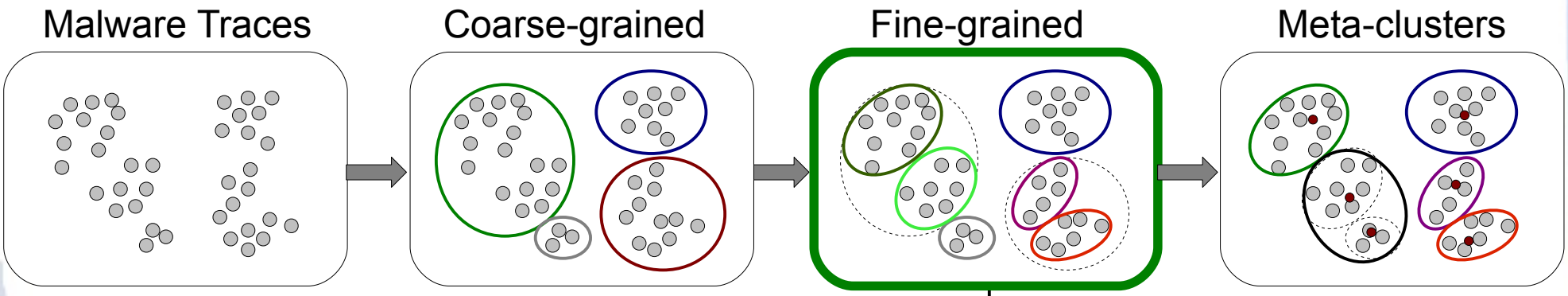


	Feb09	*Mar09*	*May09*	*Jun09*
Sig_Feb09 net-clusters	78.6%	48.9%	-	-
Sig_Feb09 net-fg-clusters	60.1%	35.1%	-	-
Sig_Feb09 sys-clusters	56.9%	33.9%	-	-
Sig_May09 net-clusters	-	-	56.0%	44.3%
Sig_May09 net-fg-clusters	-	-	50.8%	42.5%
Sig_May09 sys-clusters	-	-	32.7%	32.0%

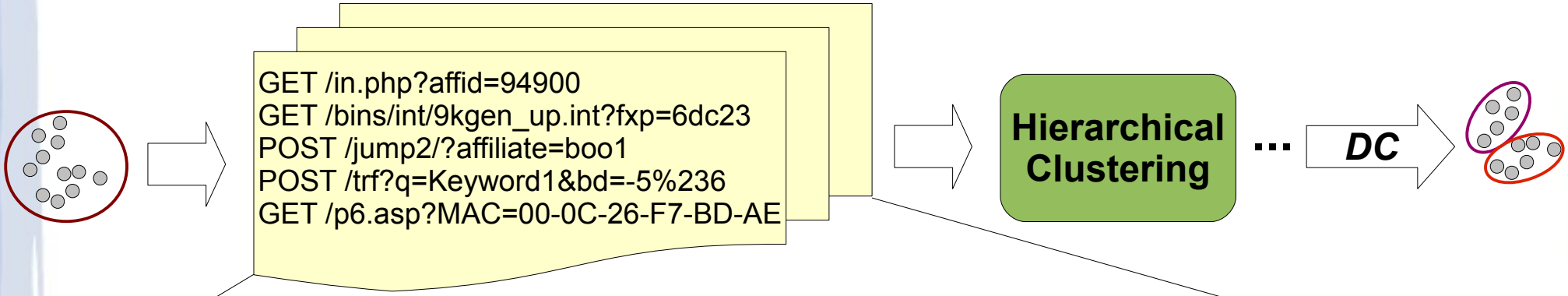
Reduced dataset of ~4k malware samples

***net-clusters* = our three-step clustering approach**
***net-fg-clusters* = only fine-grained clustering**
***sys-clusters* = using approach proposed in [Bayer et al. NDSS 2009]**

Network-level Clustering



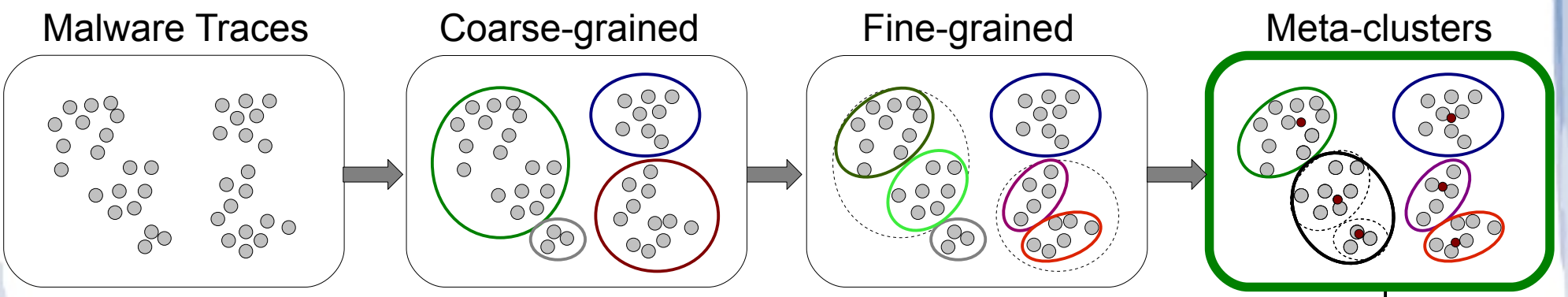
Structural Features



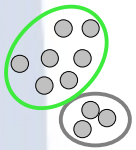
- **POST /trf?q=Keyword1&bd=-5%236**
- **POST /trf?q=Keyword3&bd=-1%231**

$$d_r(r_k, r_h) = w_m \cdot d_m(r_k, r_h) + w_p \cdot d_p(r_k, r_h) + w_n \cdot d_n(r_k, r_h) + w_v \cdot d_v(r_k, r_h)$$

Network-level Clustering

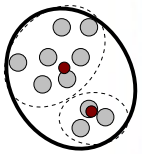


Centroid Signatures



```
GET /in\.php\?affid=.*
GET /bins/int/9kgen_up\.int\?fxp=.*
POST /jump2/\?affiliate=boo.*
POST /trf\?q=Keyword.*&bd=.*%23.*
GET /p6\.asp\?MAC=.*
```

Hierarchical Clustering ... ***DC***



POST /trf\?q=Keyword.*&bd=+.*%236

POST /trf\?q=Keyword.*&bd=-1%20.*

$$d(s_i, s_j) = \frac{agrep(s_i, s'_j)}{length(s'_i)}$$