The Case for Real Time DNS Exfiltration Detection and Prevention

By: Yarin Ozery
yozery@akamai.com

Agenda

- DNS exfiltration overview
  - How it is performed
  - Risks imposed by DNS exfiltration
- Existing detection methods
  - General overview
  - Limitation of existing methods
- Proposed solution
- Evaluation
  - Comparison with prominent methods on synthetic dataset
  - Real-world evaluation
- Discussion and Conclusion
  - Limitations
  - Future Work

About the Presenter

- Senior software engineer and security researcher at Akamai Technologies, inc.
- M.Sc Student at the software and system information department at Ben-Gurion University of the Negev, Beer Sheva, Israel.

What is DNS tunneling and exfiltration?

- The practice of establishing covert communication channel over the DNS protocol to enable unauthorized data exchange [1]
- Malicious use cases:
  - Botnets communication with C&C servers [2]
  - Bypassing paid WiFi captive portals [3]
  - Data exfiltration out of protected networks [3]
- Some benign (yet, unintended) use cases:
  - DNS-based anti malware and anti spam services [4]
  - Antivirus agents file signature search [4]

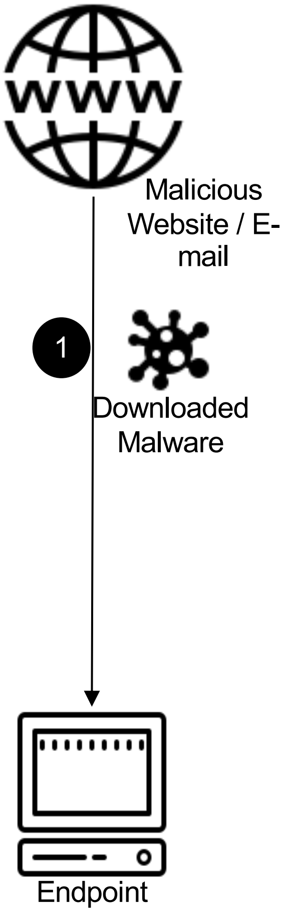
Different Information Vectors

- Query name based - encoding data as a prefix of the DNS query name to be resolved
  - Up to 255 Bytes per packet
- Query type based - encoding data within the requested DNS query resource record type, known as QTYPE
  - Up to 2 Bytes per packet
- Timing based - encoding data based on DNS queries timing
  - Not unique to DNS

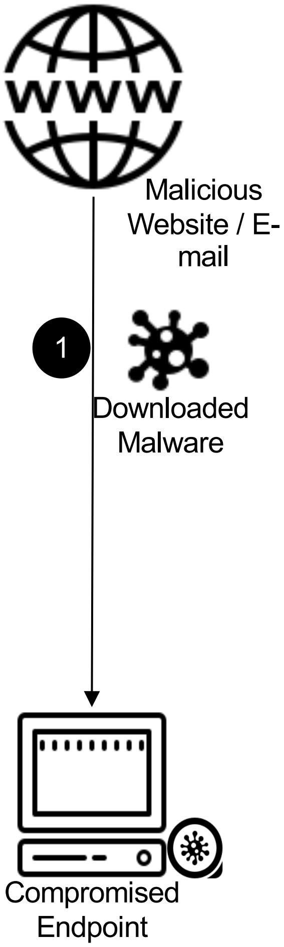
DNS Exfiltration In Practice

- Register a domain name (or multiple domain names)
- Setting an authoritative domain nameserver to the registered domain
- Encoding data within DNS packets
- Many publicly available tools:
  - Iodine [5]
  - DNS2tcp [6]
  - DNSCat2 [7]
  - Heyoka [8]
  - Many more

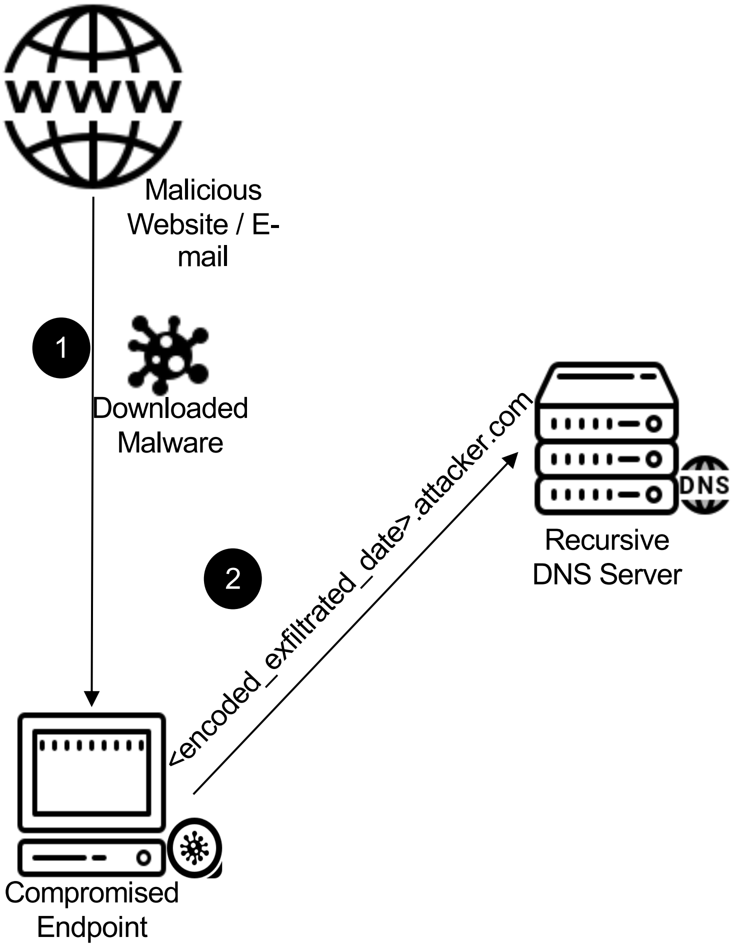
* Based on a figure from [4]



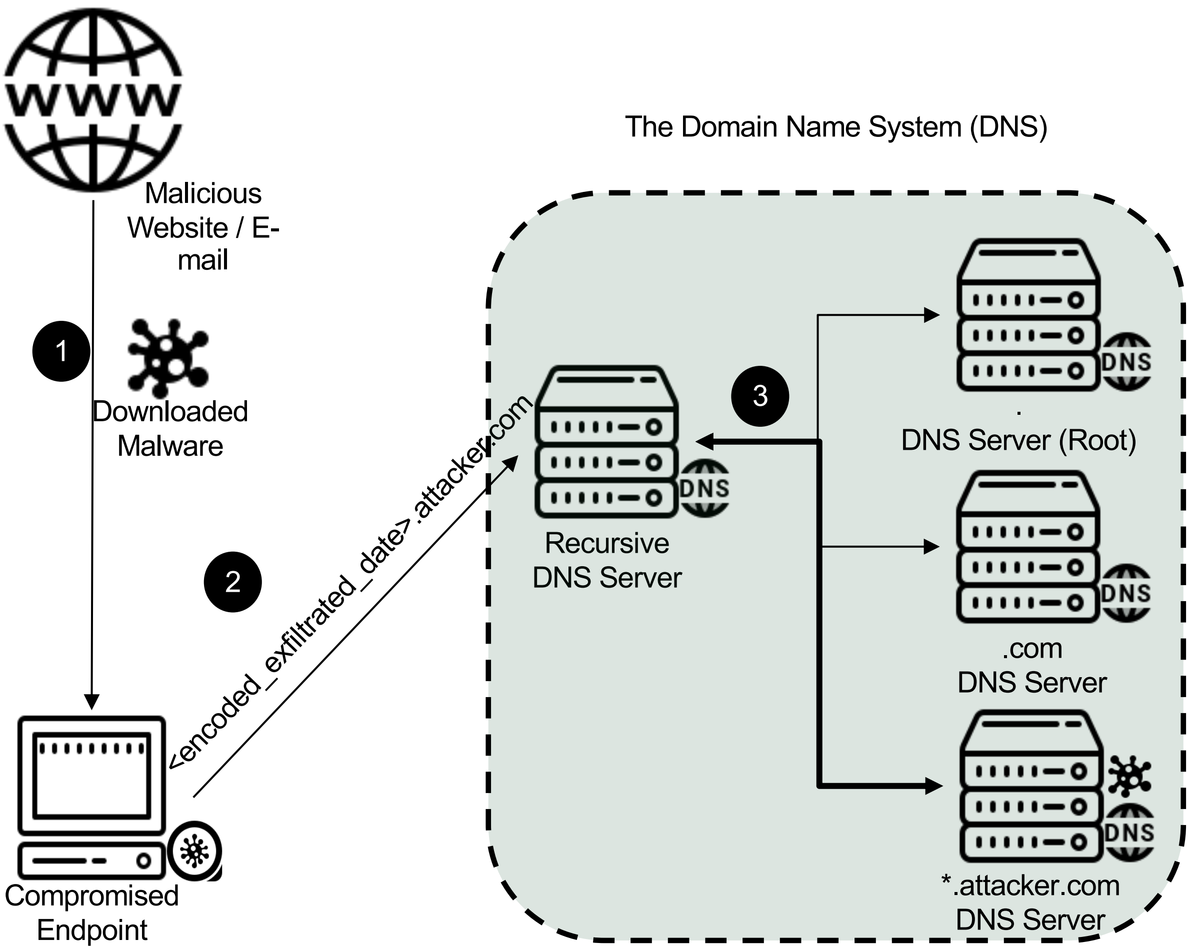
* Based on a figure from [4]



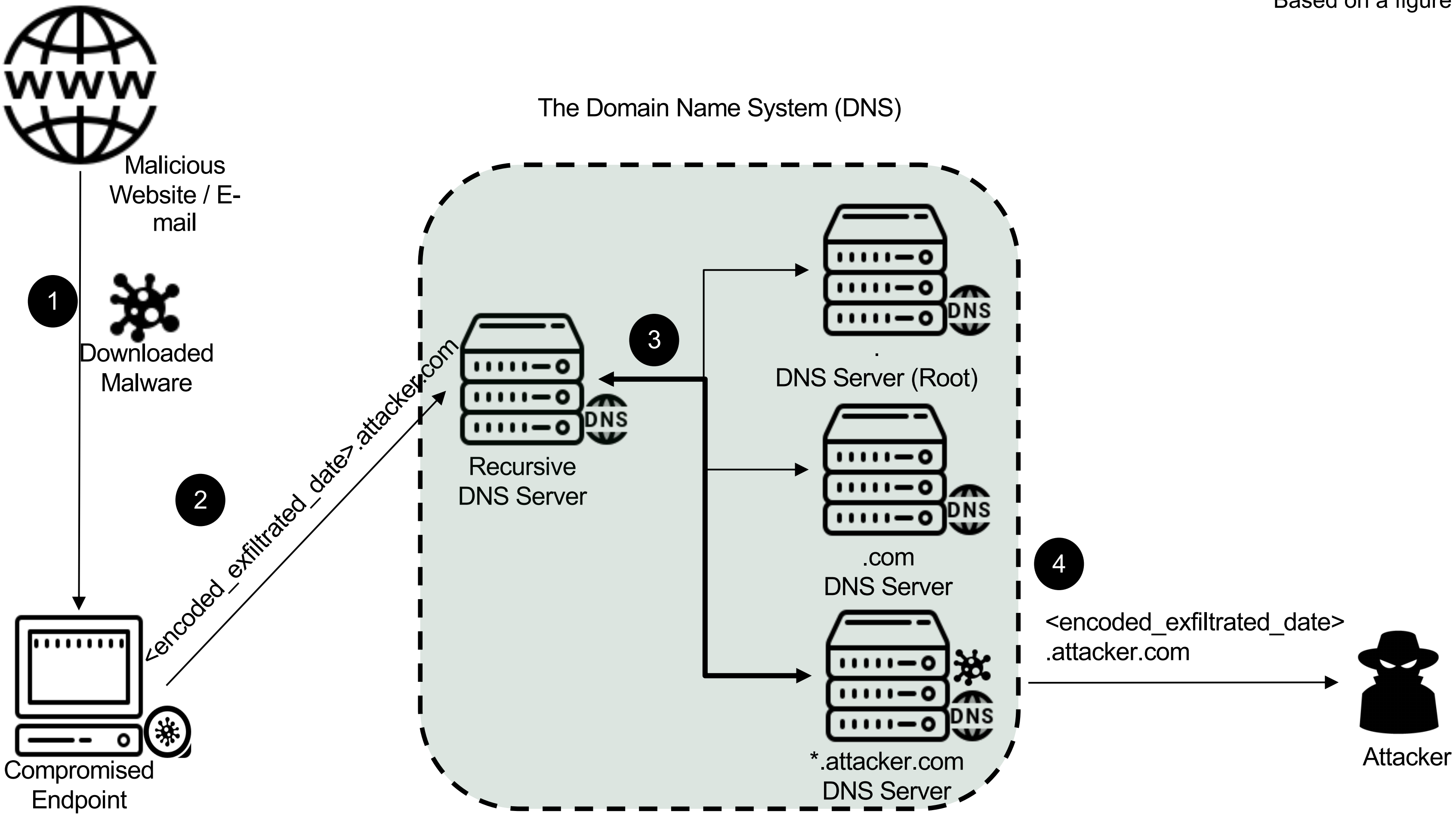
* Based on a figure from [4]



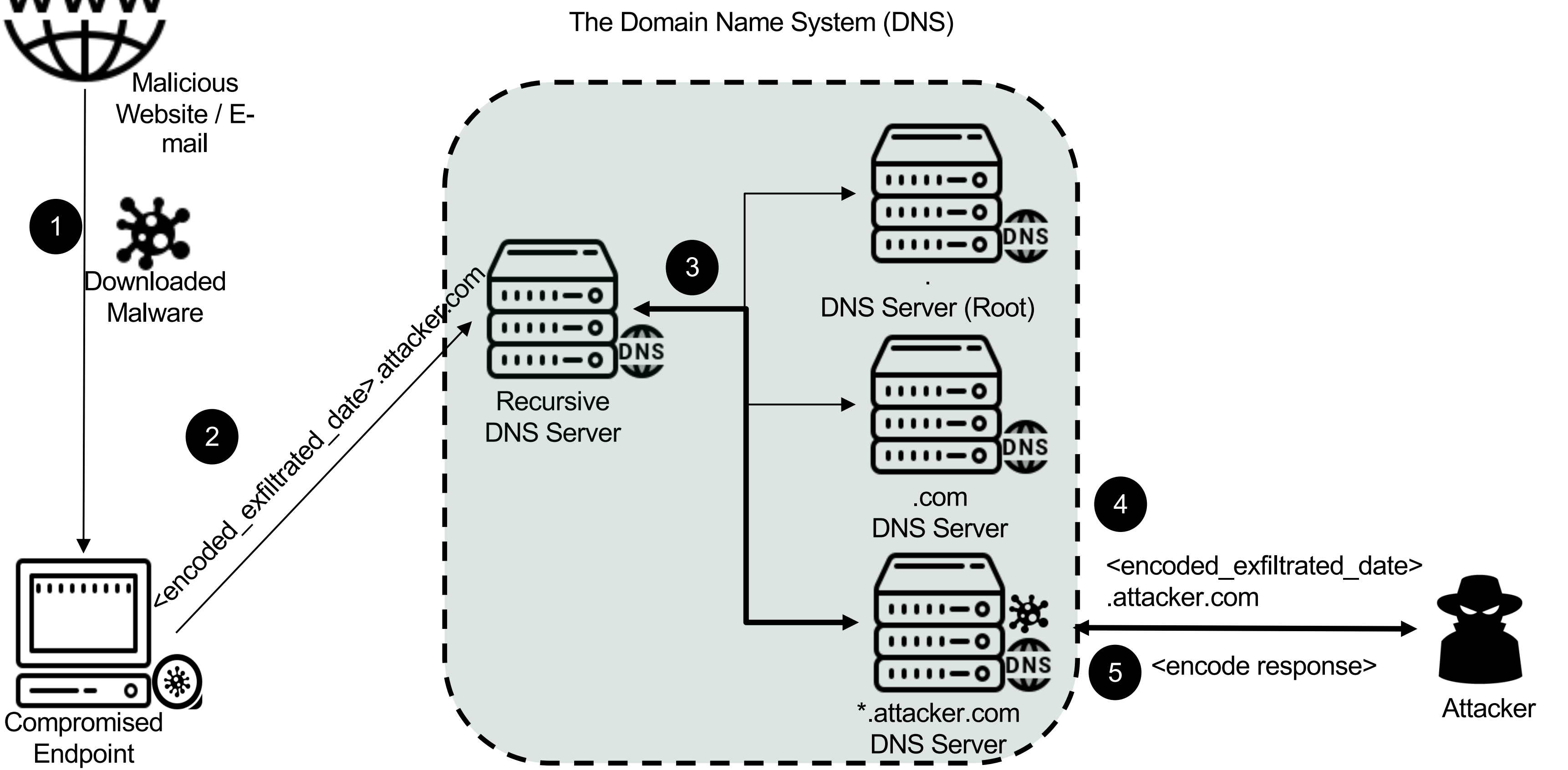
* Based on a figure from [4]



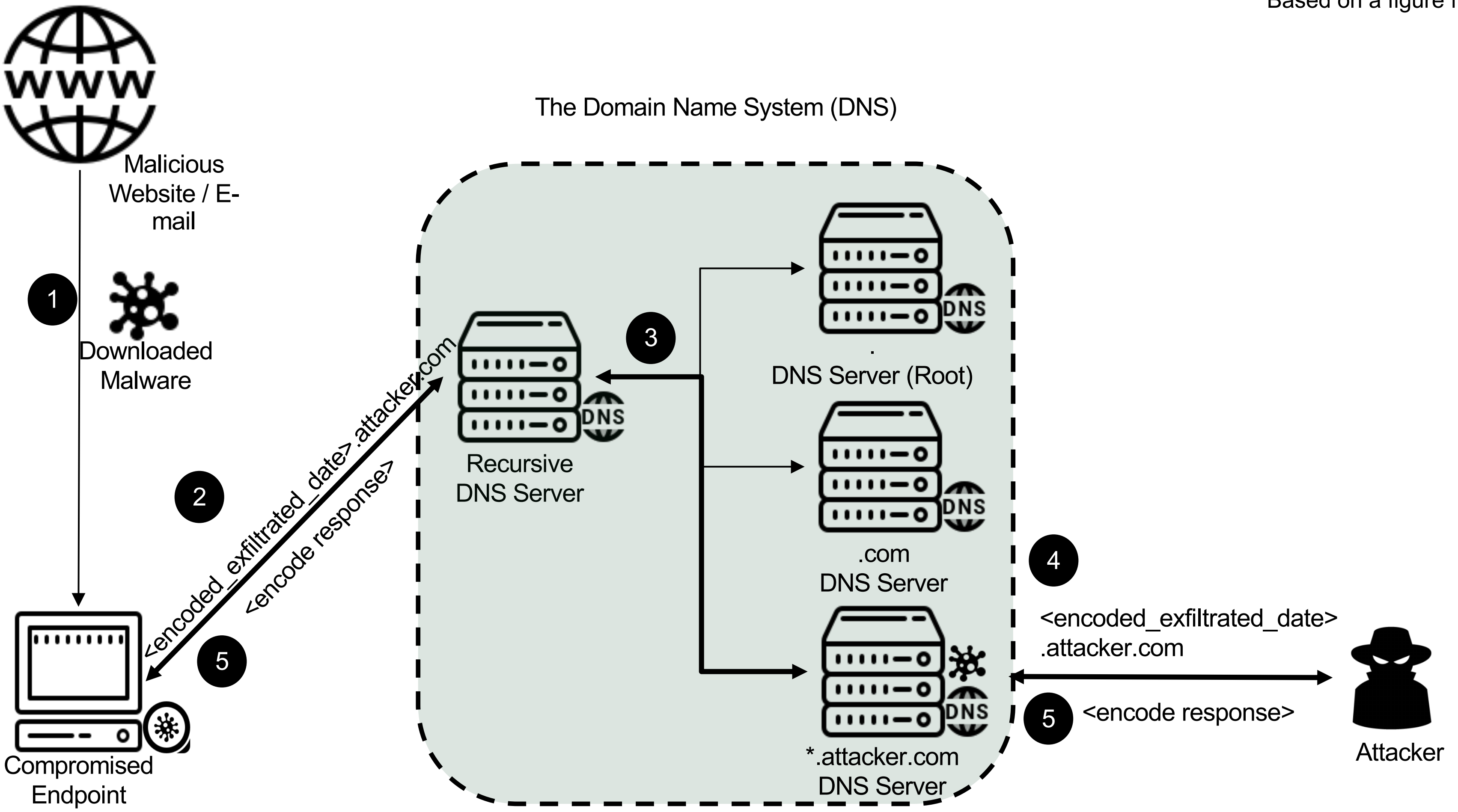
* Based on a figure from [4]



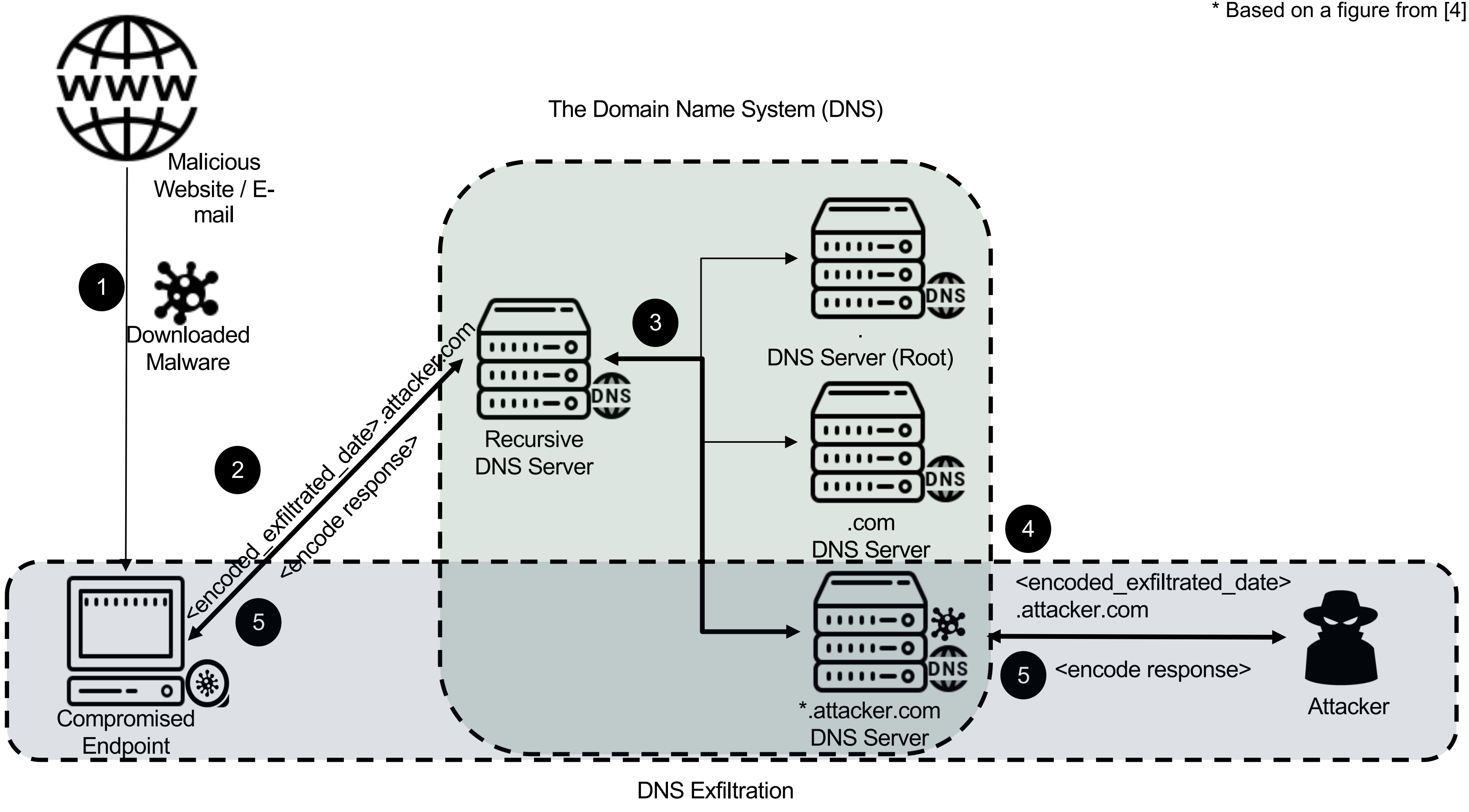
* Based on a figure from [4]



* Based on a figure from [4]



* Based on a figure from [4]



DNS Tunneling and Exfiltration in the Wild

[9]

Software used by Home Depot hackers different from Target attack

The malware used against Home Depot is a different type than that used against Target. Its name, FrameworkPOS, is derived from the McAfee Inc. antivirus agent it impersonates.

[10]

Feederbot Botnet Using DNS as Carrier for Command and Control (C2)

[27]

New Linux botnet exploits Log4J, uses DNS tunneling for comms

[11]

OilRig Deploys "ALMA Communicator" – DNS Tunneling Trojan

[12]

Wekby APT Gang Using DNS Tunneling for Command and Control

[13]

RANSOMWARE ACTORS LEANING ON DNS TUNNELING

Existing DNS Exfiltration Detection Solutions

- Over 30 research papers have been published on the topic in recent years[17]
- Two main strains:
  - Payload based - classification is done on a per-packet (or per small number of packets)
  - Traffic based - classification is done based on overall DNS traffic.
- Most focus dedicated to machine learning based solutions
  - Supervised learning
  - Unsupervised learning
  - Deep learning
  - Also, some statistics-based and rule-based solutions

Features Used for DNS Exfiltration Detection [26]

- Size of DNS requests and responses
- Length of destination hostname
- Entropy of hostname
- Volume of DNS traffic per destination
- Volume of DNS traffic from source
- Volume of uncommon DNS query types
  - NULL
  - TXT
- Signatures of specific tools

Paxson et al. [14]

- DNS queries are aggregated daily per client and registered domain
- Information quantification is done by compressing the information vector and taking the length of the output as the information quantity
- All queries over the time window are needed to be stored
  - substantial computation and memory requirements
- Can be applied to any information vector

Nadler et al. [15]

- Traffic-based unsupervised machine learning model
  - Based on the isolation forest algorithm
- Feature extraction is based on a per domain basis in a sliding window manner
  - Requires holding all the queries in each classification window
  - Data is needed to be kept up to six hours for a successful detection in some cases
  - ”Expensive” features – average longest meaningful word, average entropy
- Can detect very slow campaigns
  - As slow as 0.11 B/s

Ahmed et al. [16]

- Payload-based unsupervised machine learning model
  - Based on the isolation forest algorithm
  - Only needs to store the trained model in memory
- Feature extraction is done based on the query name
  - Ten different features are extracted for every query
  - Example: query name length, subdomain length, query name entropy
- True real time solution
  - Questionable scalability

Limitation of Existing Methods

- Most focus dedicated to increasing detection efficacy
  - Not as much effort put into designing fast real-time scalable solutions
- Result: Significant amounts of data exfiltrated by the time of detection
- Solution: DNS exfiltration detection method that can classify DNS queries as they are resolved
  - Preferably, directly on the recursive DNS resolver
  - Must have a small memory footprint and fast classification

Information-based Heavy Hitter (ibHH) for Real-time DNS Exfiltration Detection

- Idea: Quantify the amount of information transmitted to a registered **domain** through DNS queries based on the length of **unique** subdomains, raise alert if the amount exceeds a predefined threshold
  - We call these **information heavy hitter** domains
  - Inspired by the works of Paxson et al. [14] and Afek et al. [23]
- Problem: Exact solution requires memory linearly proportional to the DNS queries stream size and long computation time
- Solution: approximate information quantities with sketching algorithms

Sketching Algorithms [19]

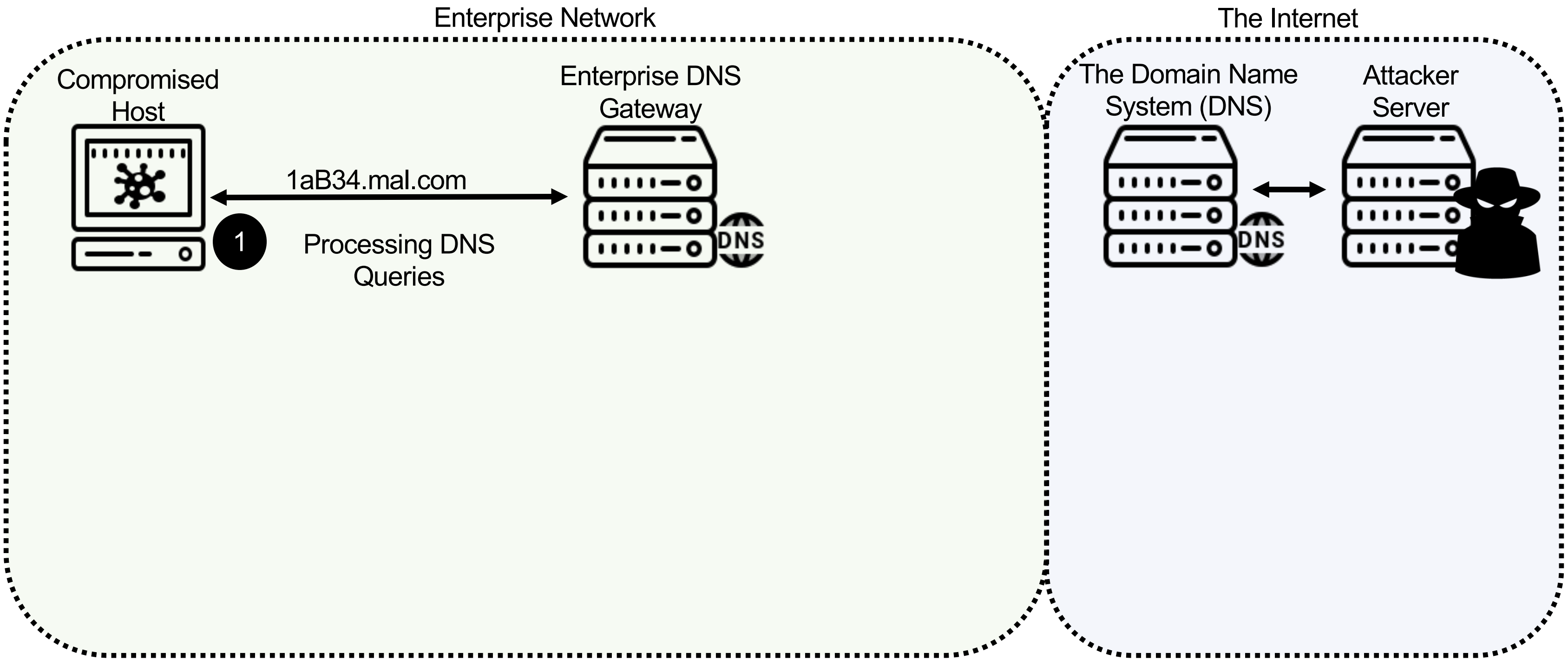
- A compressed representation of a stream of data
  - In the streaming model, each item is observed once [18]
  - Cannot store the entire stream in memory
- Enable accurate estimations of tasks that require inspection of the entire stream
- Examples
  - Count Min Sketch [20] – approximates the **frequencies** of elements in a stream
  - HyperLogLog (HLL) [21] – approximates the **number of distinct** elements in a stream
- Use cases
  - Traffic engineering: load balancing in high throughput environments [22]
  - DNS-based DDoS protection [23]

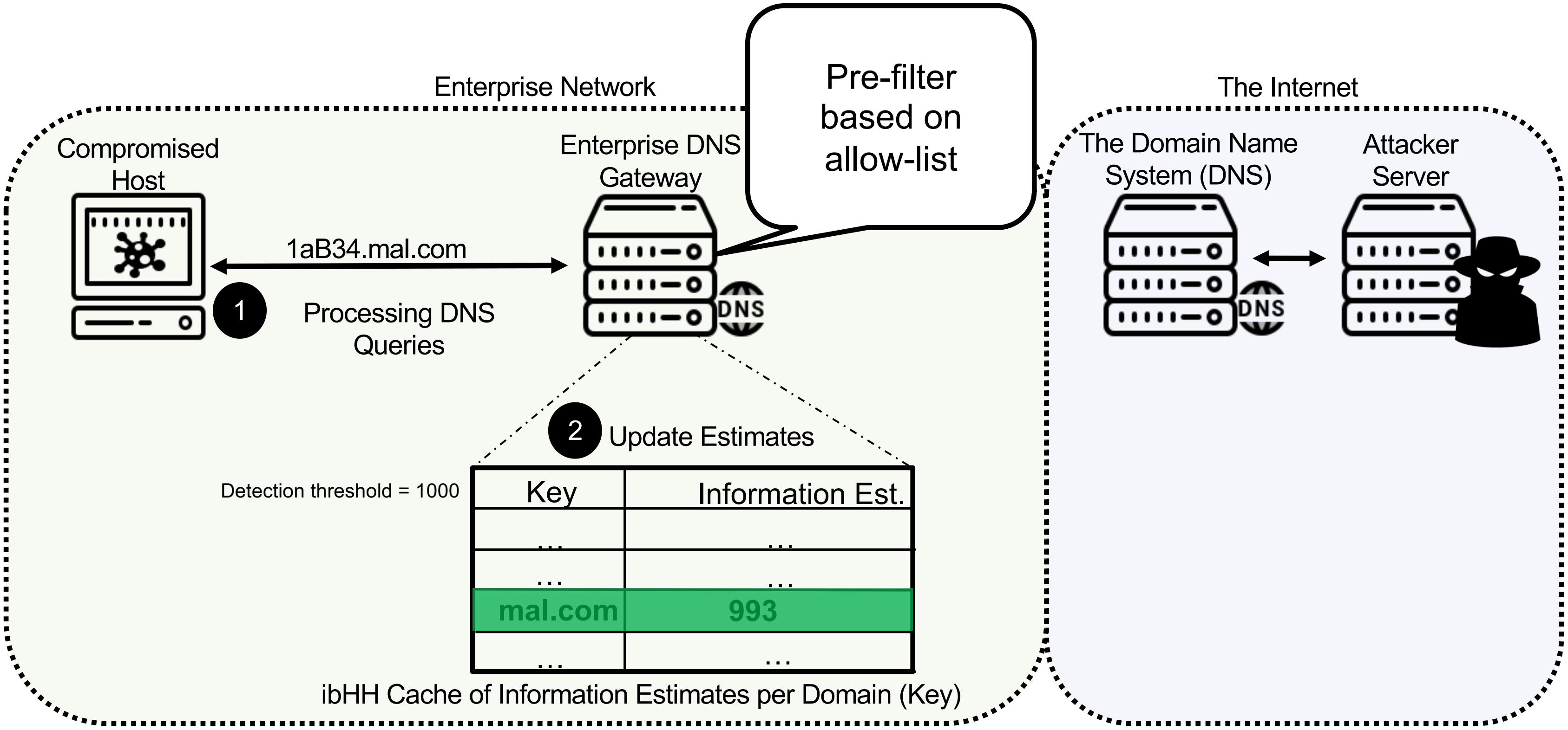
Information-based Heavy Hitters for Real-time DNS Exfiltration Detection

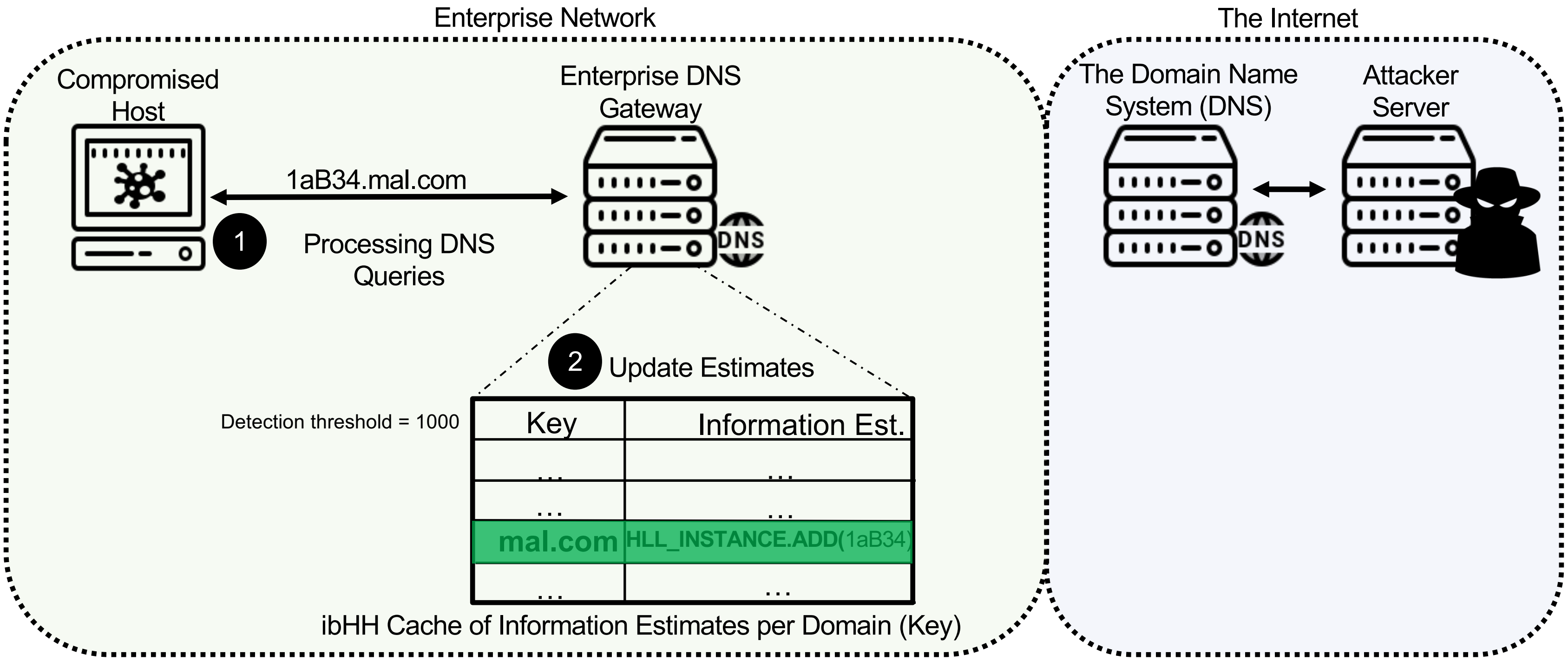
- We model the DNS queries as an online stream of (domain, subdomain) pairs
  - Every DNS query name is split into the registered domain name and the subdomain prefix
  - Example: www.google.com -> **domain** = google.com, **subdomain** = www
- With the use of hashing, weighted sampling technique and a variation of **HLL**, we detect information heavy hitters in the stream
- Hold a cache of **fixed** size *k*, storing information heavy hitter **domains** in the DNS stream and a **HLL** data sketch for every cached domain.
- Alert is raised when the amount of information estimation exceeds the detection threshold

Handling False Positive Alerts

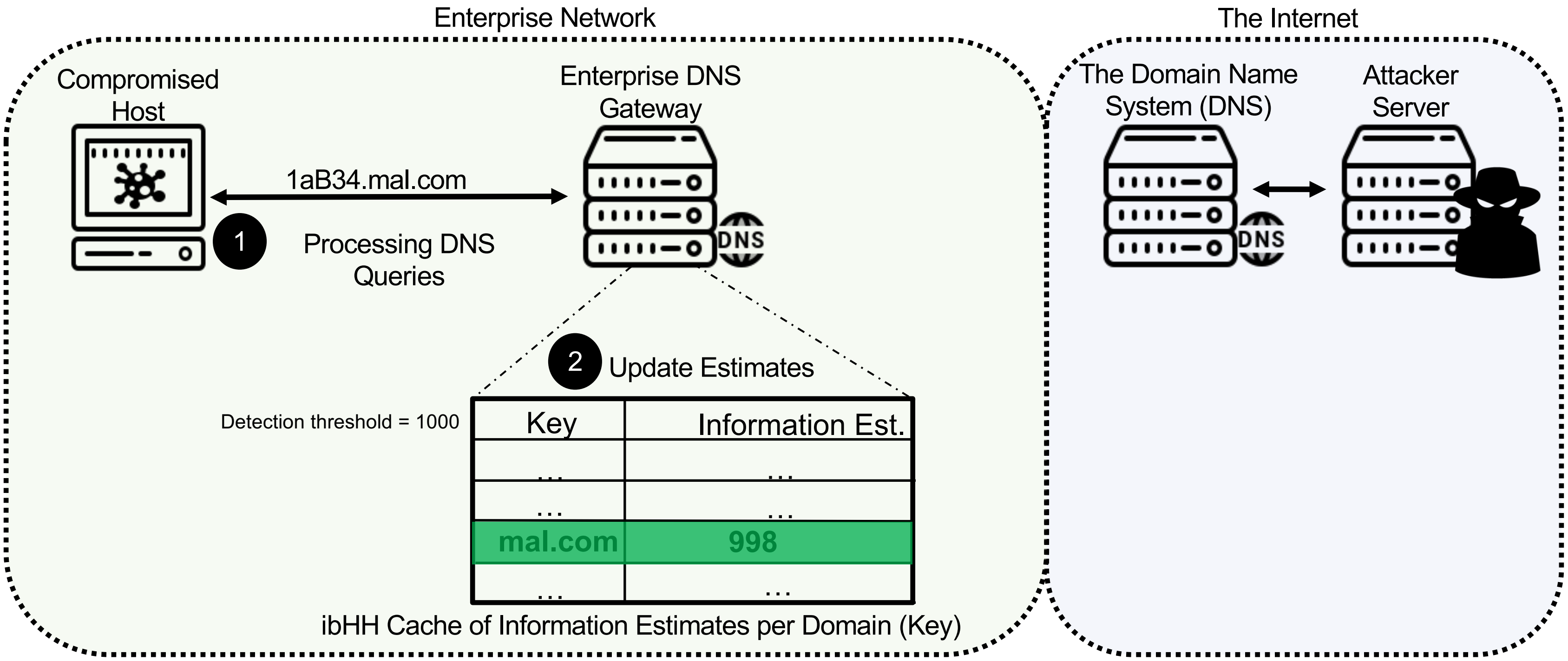
- Popularity-based allow-list to reduce the number of false positive
  - Specifically, the TRANCO [29] top sites ranking was used in our experiments
- Domains in the allow-list are pre-filtered
  - Avoid filling the cache with benign information heavy hitter
- Offers significant reduction in false positive alerts

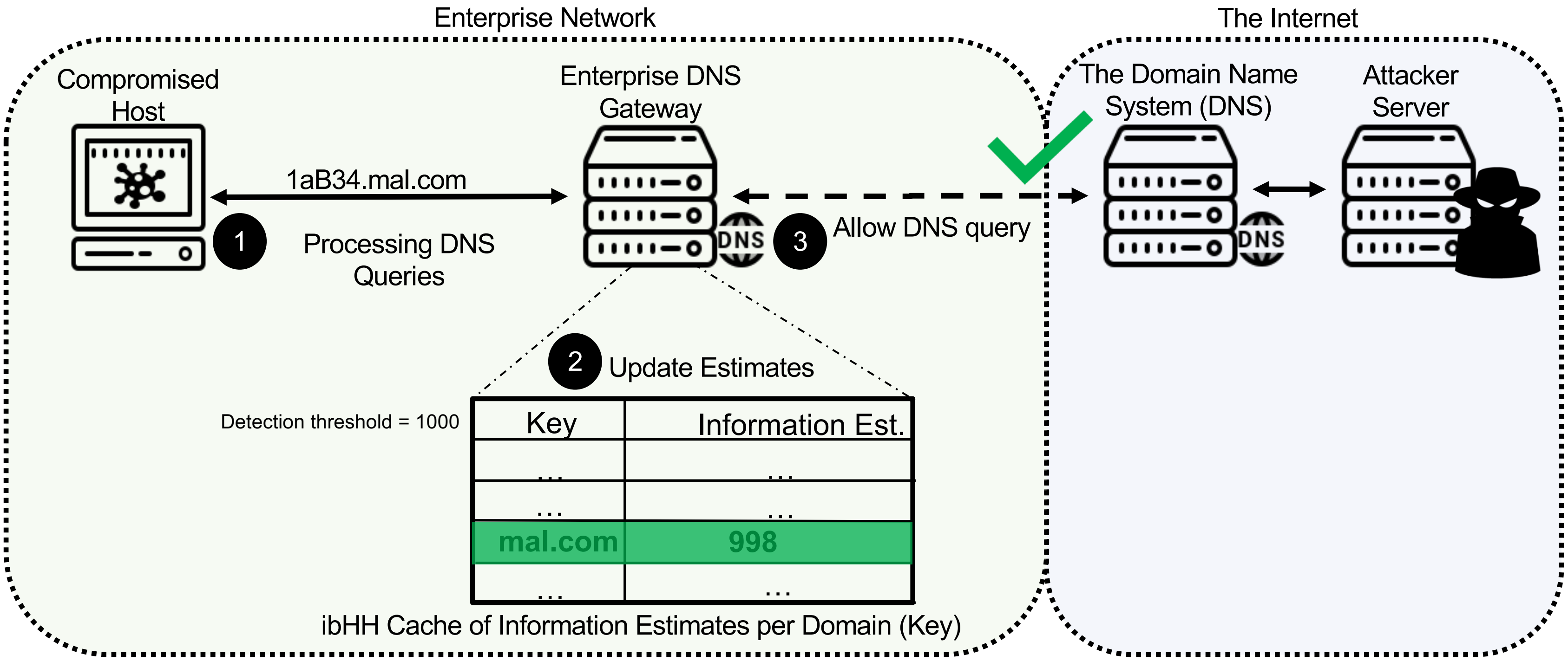

Enterprise Network
The Internet
Compromised Host
Enterprise DNS Gateway
The Domain Name System (DNS)
Attacker Server
1aB34.mal.com
1
Processing DNS Queries
DNS
DNS


Enterprise Network
Pre-filter based on allow-list
The Internet
Compromised Host
Enterprise DNS Gateway
The Domain Name System (DNS)
Attacker Server
1aB34.mal.com
1
Processing DNS Queries
DNS
DNS
2
Update Estimates
Detection threshold = 1000
Key
Information Est.
...
...
...
...
mal.com
993
...
...
ibHH Cache of Information Estimates per Domain (Key)

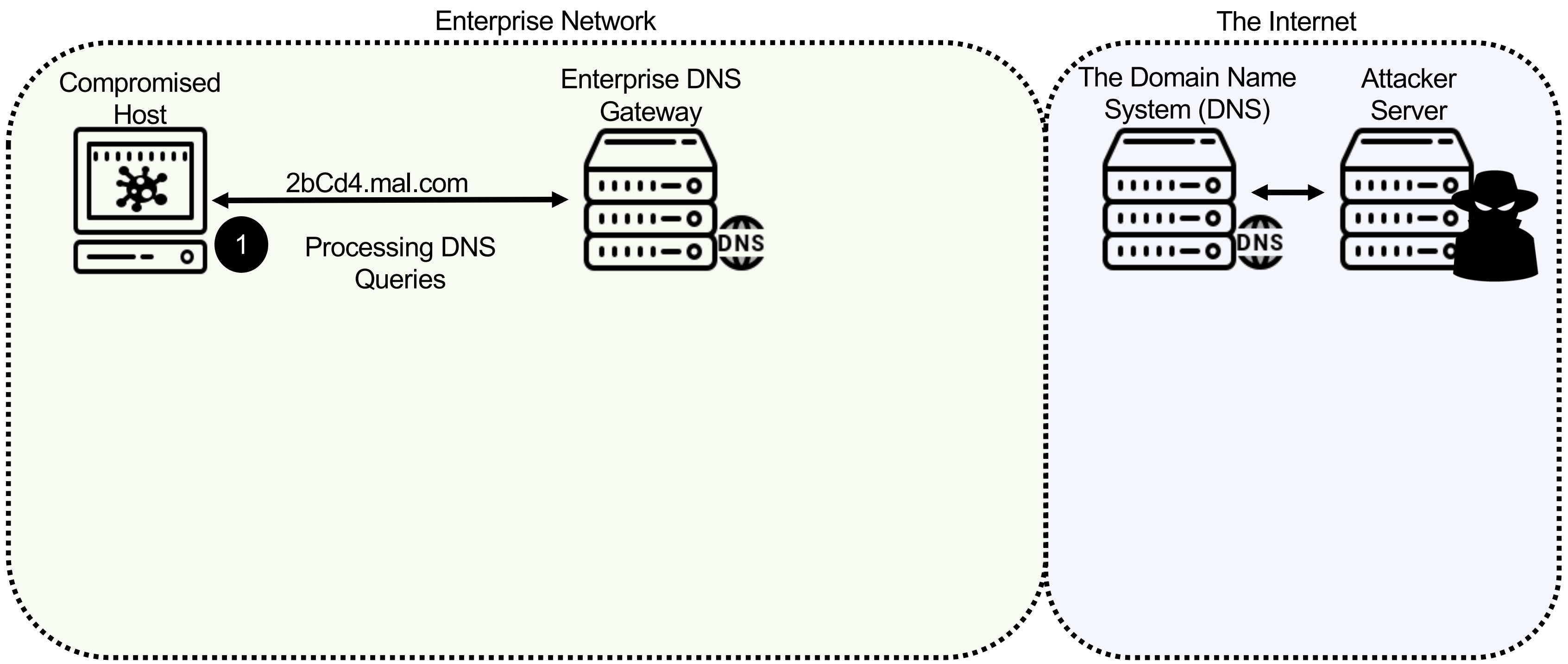


Enterprise Network
The Internet
Compromised Host
Enterprise DNS Gateway
The Domain Name System (DNS)
Attacker Server
1aB34.mal.com
1
Processing DNS Queries
DNS
DNS
2
Update Estimates
Detection threshold = 1000
Key
Information Est.
...
...
...
...
mal.com
HLL_INSTANCE.ADD(1aB34)
...
...
ibHH Cache of Information Estimates per Domain (Key)

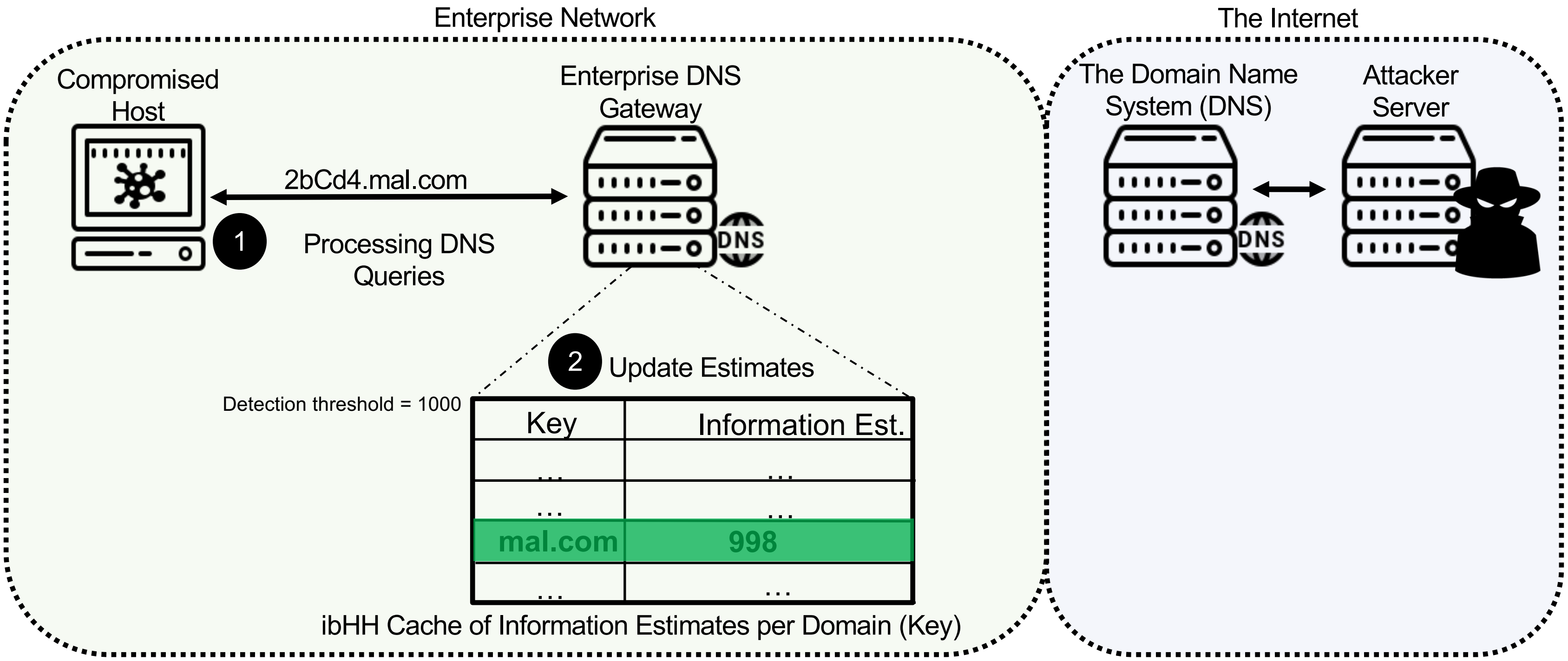

Enterprise Network
The Internet
Compromised Host
Enterprise DNS Gateway
The Domain Name System (DNS)
Attacker Server
1aB34.mal.com
1
Processing DNS Queries
DNS
DNS
2
Update Estimates
Detection threshold = 1000
Key
Information Est.
...
...
...
...
mal.com
998
...
...
ibHH Cache of Information Estimates per Domain (Key)

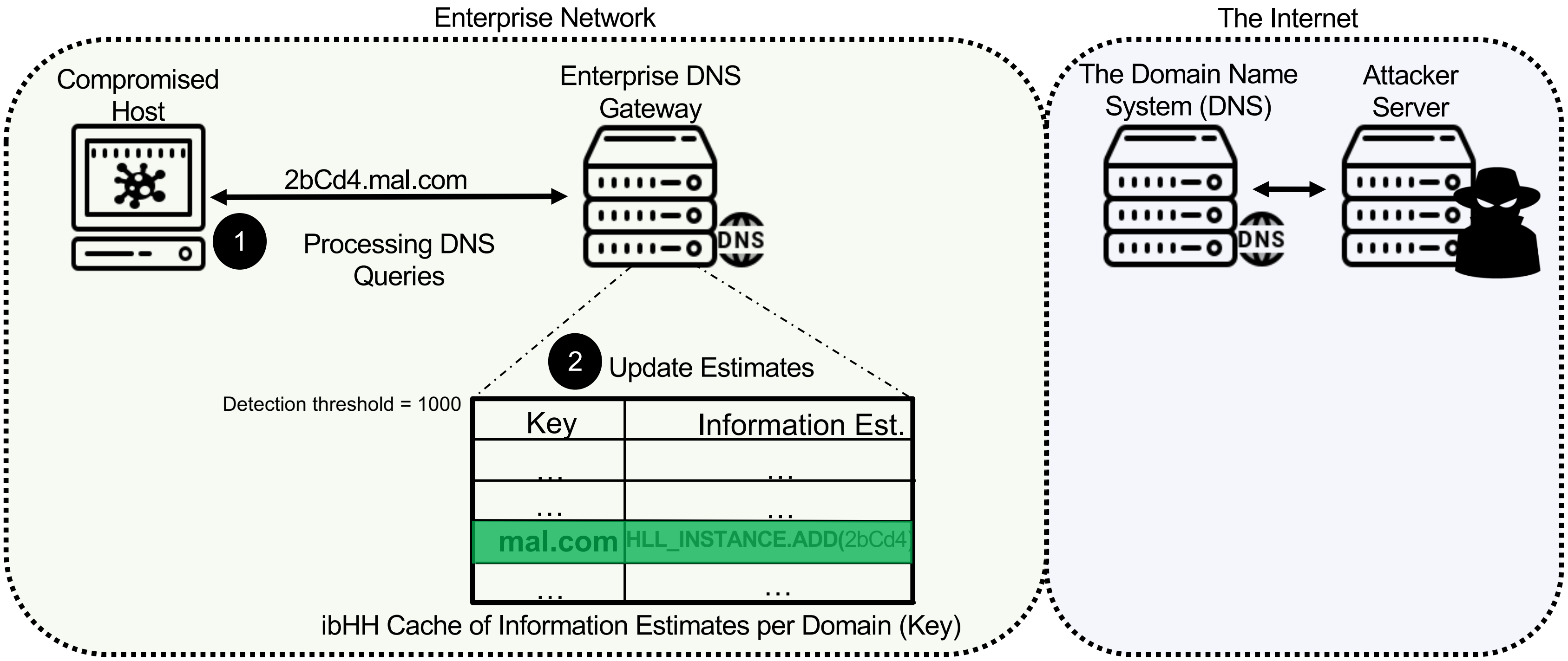


Enterprise Network
The Internet
Compromised Host
Enterprise DNS Gateway
The Domain Name System (DNS)
Attacker Server
1aB34.mal.com
1
Processing DNS Queries
DNS
3
Allow DNS query
DNS
2
Update Estimates
Detection threshold = 1000
Key
Information Est.
...
...
...
...
mal.com
998
...
...
ibHH Cache of Information Estimates per Domain (Key)

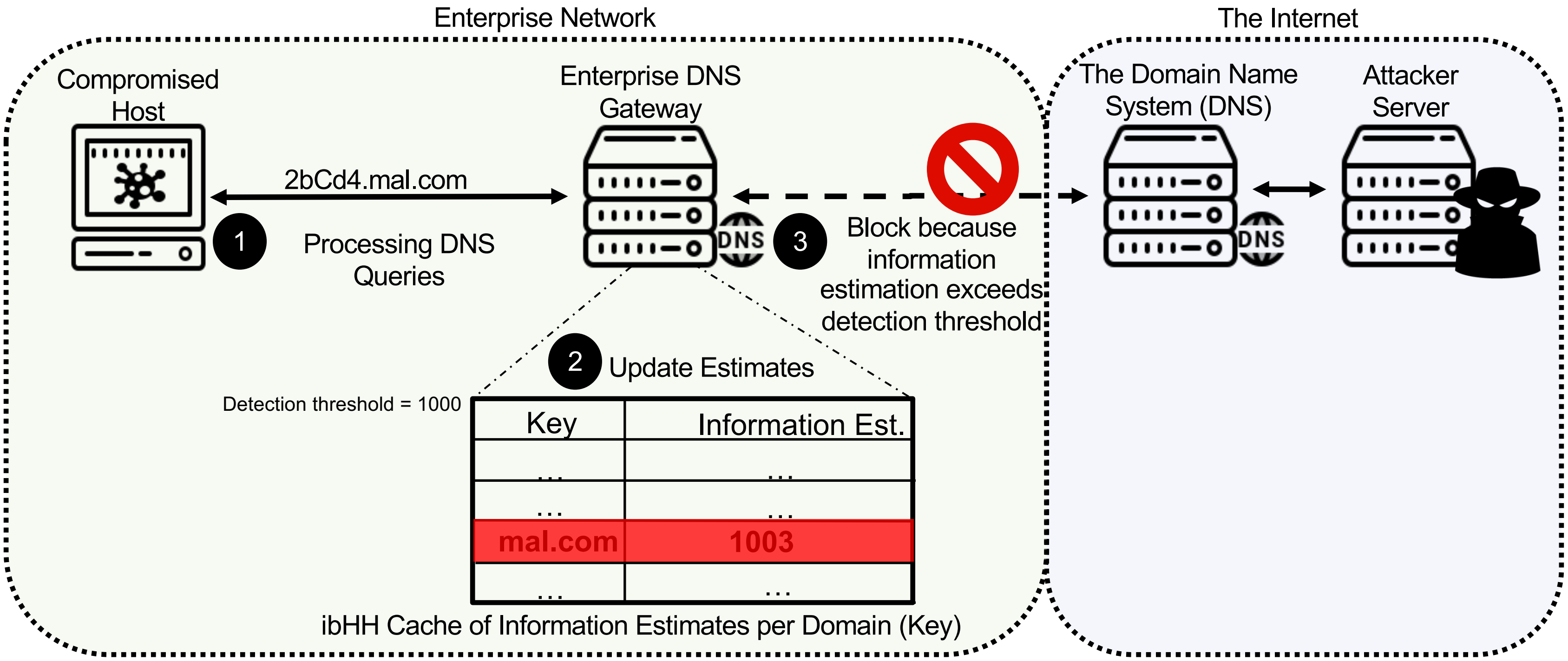


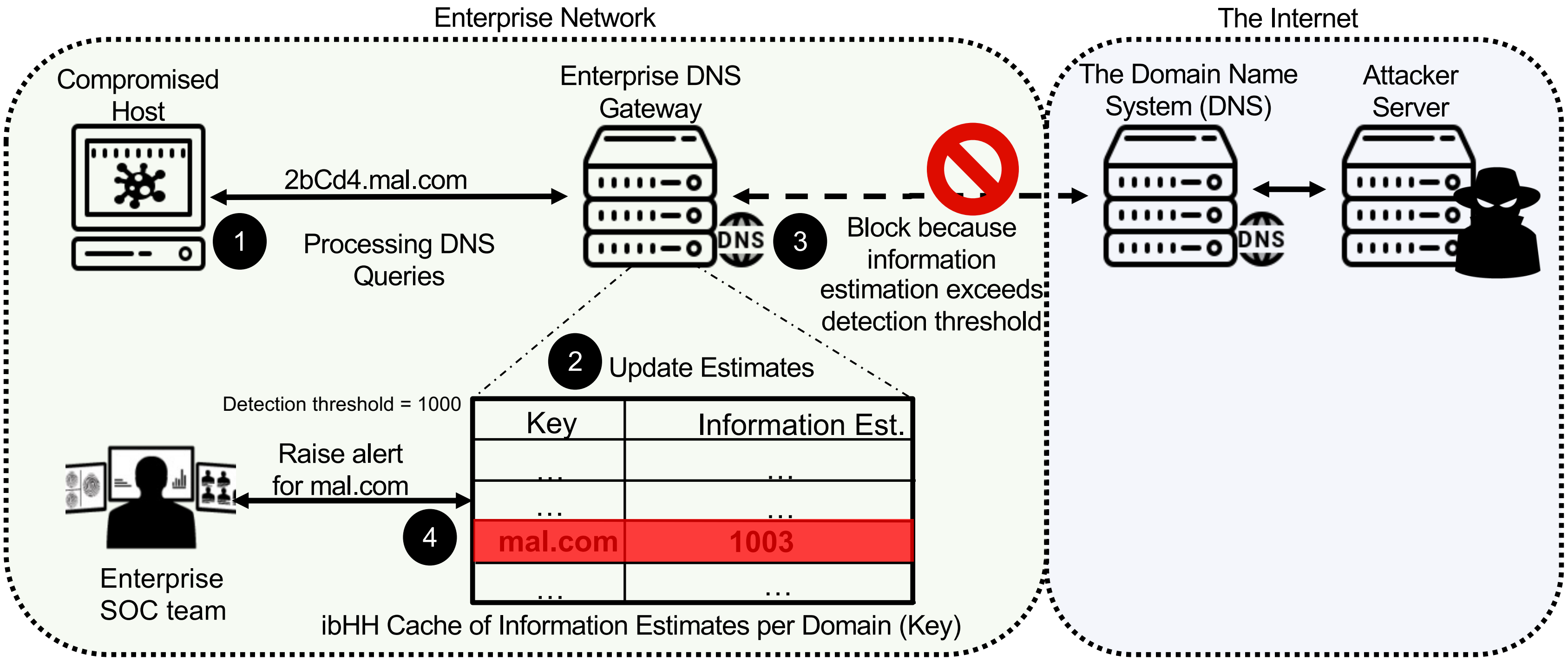
Enterprise Network
The Internet
Compromised Host
Enterprise DNS Gateway
The Domain Name System (DNS)
Attacker Server
2bCd4.mal.com
1
Processing DNS Queries
DNS
DNS


Enterprise Network
The Internet
Compromised Host
Enterprise DNS Gateway
The Domain Name System (DNS)
Attacker Server
2bCd4.mal.com
1
Processing DNS Queries
DNS
DNS
2
Update Estimates
Detection threshold = 1000
Key
Information Est.
...
...
...
...
mal.com
998
...
...
ibHH Cache of Information Estimates per Domain (Key)

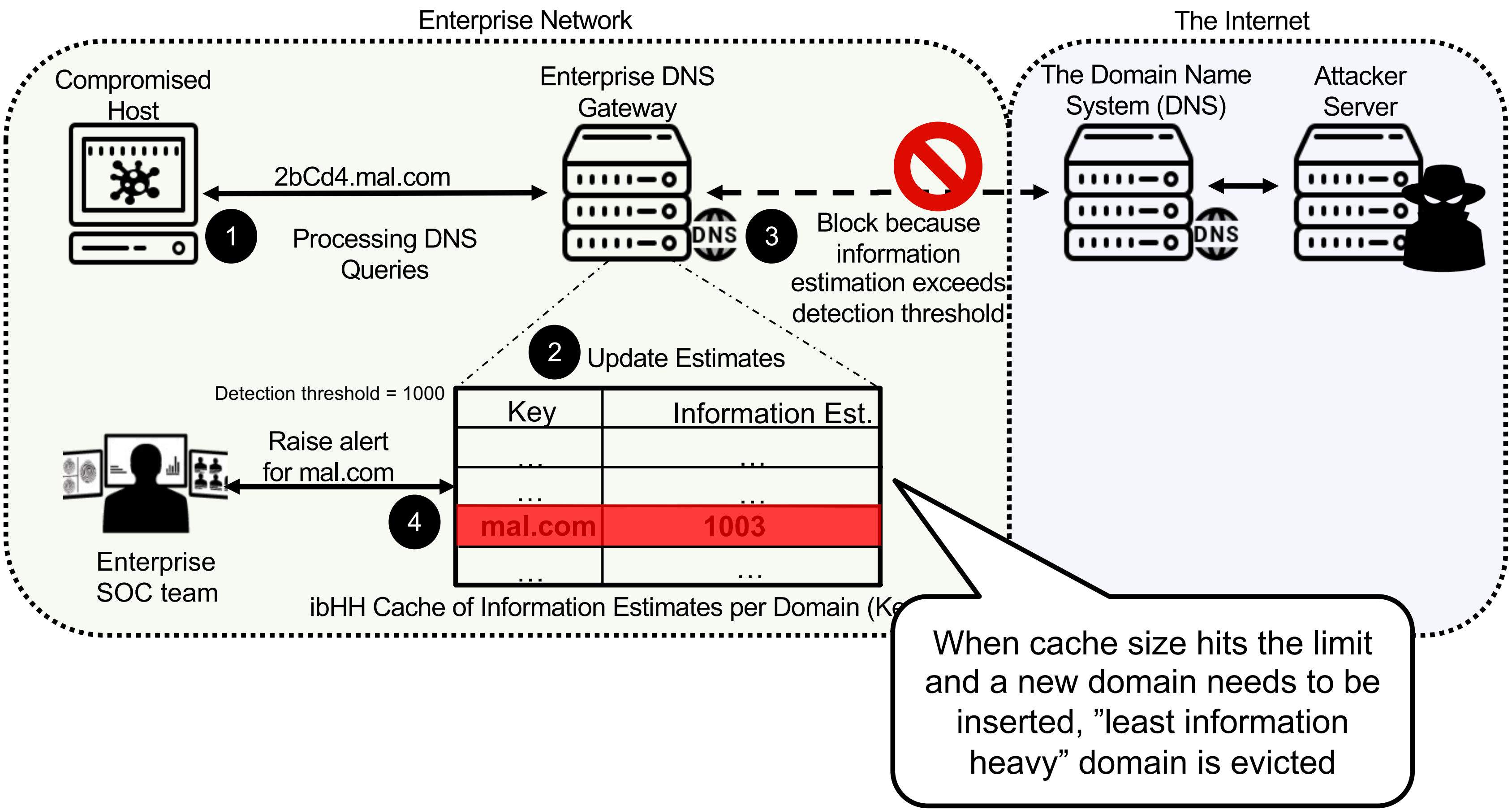


Enterprise Network
The Internet
Compromised Host
Enterprise DNS Gateway
The Domain Name System (DNS)
Attacker Server
2bCd4.mal.com
1
Processing DNS Queries
DNS
DNS
2
Update Estimates
Detection threshold = 1000
Key
Information Est.
...
...
...
...
mal.com
HLL_INSTANCE.ADD(2bCd4)
...
...
ibHH Cache of Information Estimates per Domain (Key)


Enterprise Network
The Internet
Compromised Host
Enterprise DNS Gateway
The Domain Name System (DNS)
Attacker Server
2bCd4.mal.com
1
Processing DNS Queries
DNS
3
Block because information estimation exceeds detection threshold
DNS
2
Update Estimates
Detection threshold = 1000
Key
Information Est.
...
...
...
...
mal.com
1003
...
...
ibHH Cache of Information Estimates per Domain (Key)



Enterprise Network
The Internet
Compromised Host
Enterprise DNS Gateway
The Domain Name System (DNS)
Attacker Server
2bCd4.mal.com
1
Processing DNS Queries
DNS
3
Block because information estimation exceeds detection threshold
2
Update Estimates
Detection threshold = 1000
Key
Information Est.
...
mal.com
1003
Raise alert for mal.com
4
Enterprise SOC team
ibHH Cache of Information Estimates per Domain (Key)


Enterprise Network
The Internet
Compromised Host
Enterprise DNS Gateway
The Domain Name System (DNS)
Attacker Server
2bCd4.mal.com
1
Processing DNS Queries
DNS
3
Block because information estimation exceeds detection threshold
DNS
2
Update Estimates
Detection threshold = 1000
Key
Information Est.
...
...
...
...
mal.com
1003
...
...
Raise alert for mal.com
4
Enterprise SOC team
ibHH Cache of Information Estimates per Domain (Ke
When cache size hits the limit and a new domain needs to be inserted, ”least information heavy” domain is evicted

Experiments

- Dataset:

# DNS queries	# Unique registered domains	Timespan
50,853,030,033	43,310,209	8 Days

- **Anonymized** monitored data of enterprise organizations customers

Methodology

- Injecting synthetic malicious DNS exfiltration traffic with 1,300 distinct domains to the dataset
- Malicious traffic generated with:
  - Iodine [5] – publicly available DNS tunneling tool, simulates browsing over
  - FrameworkPOS [24] – simulates exfiltration of credit card details, sending three queries per second
  - Backdoor.Denis [25] – simulates C2 communication over DNS, sending a query every 1.5 seconds.
- Each method trained with different acceptable false positive rates
  - Ranging between 0.01 to 0.0001
- Measure true positive rate (TPR) and false positive rate (FPR) of each method
  - Based on the count of **registered** domain alerts
- TRANCO top 1M allow-list applied to all methods

Results

Method	Dataset	FPR=0.01				FPR=0.001				FPR=0.0001			
		TD[1]	FPR	TPR	DER[1]	TD[1]	FPR	TPR	DER[1]	TD[1]	FPR	TPR	DER[1]
	$DS_p + I$	1734	**0.0037**	**1.0**	0.7	1420	**0.001**	**1.0**	5	1343	<0.001	**1.0**	65
ibHH	$DS_p + F$	1743	**0.0038**	**1.0**	0.7	1430	**0.001**	**1.0**	5	1298	<0.001	**0.98**	65
	$DS_p + D$	1728	**0.0037**	**1.0**	0.7	1417	**0.001**	**1.0**	5	1252	<0.001	**0.98**	65
	$DS_p + I$	3015	0.007	**1.0**		2132	0.0012	**1.0**		1342	<0.001	**1.0**	
Nadler et al.	$DS_p + F$	3015	0.007	0.99	N/A	2085	0.0012	0.96	N/A	1267	<0.001	**0.98**	N/A
	$DS_p + D$	3015	0.007	0.98		2058	0.0012	0.94		1240	<0.001	0.97	
	$DS_p + I$	3200	0.008	**1.0**		2659	0.014	**1.0**		1314	<0.001	**1.0**	
Ahmed et al.	$DS_p + F$	3214	0.008	**1.0**	N/A	2631	0.014	0.98	N/A	1107	<0.001	0.85	**N/A**
	$DS_p + D$	3170	0.008	0.98		2599	0.014	0.95		1039	<0.001	0.8	
	$DS_p + I$	1927	0.0041	**1.0**	0.9	1771	0.0023	**1.0**	12	1314	<0.001	**1.0**	70
Paxson et al.	$DS_p + F$	1927	0.0041	**1.0**	0.9	1771	0.0023	**1.0**	12	1249	<0.001	0.96	70
	$DS_p + D$	1927	0.0041	0.98	0.9	1771	0.0023	**1.0**	12	1230	<0.001	0.95	70

[1] Total Detections (#Distinct Hosts)
[2] Detectable Exfiltration Rate (B/s)

Real-world Evaluation

- Executed ibHH algorithm over the course of a month in a test environment, with different detection thresholds.
- Results:

Detection Threshold (B/s)	Number of Alerted Domains	True Positive	False Positive
50	1	1	0
25	2	1	1
15	**7**	**2-3**	**5-6**
10	15	2-3	12-13
5	38	-	-

Alerts by day

Date	Primary Domain	Classification
Day 1	TP_domain_1.com	TP
Day 3	FP_domain_1.com	FP
Day 14	TP_domain_2.com	TP
Day 21	TP_domain_3.com	TP
Day 27	FP_domain_2.com	FP
Day 27	FP_domain_3.com	FP
Day 28	FP_domain_2.com	FP
Day 30	FP_domain_4.com	FP

Examples of real DNS exfiltration queries detected

domain	subdomain	Response
TP_domain_1.com	vaaaakawgba.t1	VACK$ÔøΩ!4
TP_domain_1.com	schrqs.t1	Base128
TP_domain_1.com	pajymnaa.t1	<Base 128 encoded response>
TP_domain_1.com	pabajczq.t1	<Base 128 encoded response>
TP_domain_1.com	<Long base128 encoded data>	<Base 128 encoded response>

Examples of real DNS exfiltration queries detected

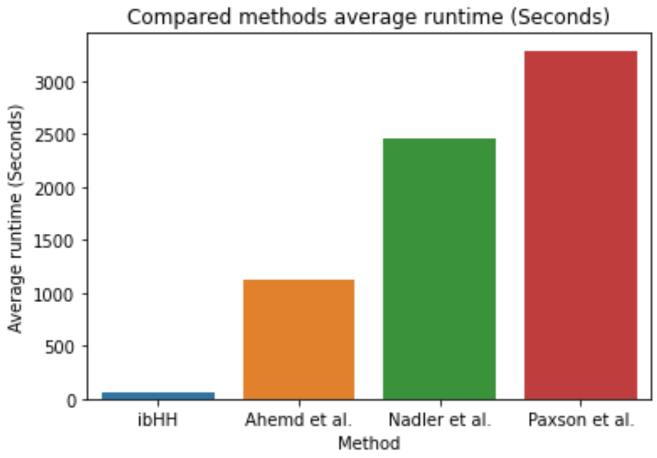
domain	subdomain	Response
TP_domain_2.com	6Ngv7RHeGapGXYnUupVf8te.tunnel.	divSaqh8aAr7F- SDRy10l2R75UijruRDGzuSNzZPx- JA04hvi+tmecvRX4SMirzRbi4sR40kPTSaB 4PmIfT8mWZC8iIGE5monrr2i5DkiekvUzyR E0zYCJMh0FJyCVO9- j+BGlTyaoBQGOzzJzauoZisladhA1kFWZW 2Bnr0Dhfo+vKNWoNZ3a2DxMT8B5-7YA- CR6p3M9GhW2HpradpFzIicUm3BVxEFfZ PEXfObQEMV4I_z++nAS16rQ__
TP_domain_2.com	6Ngv7RHeGapGTUYtcom7NmR.tunnel.	dk3Vaqh8asWwoVRUwe+srlG…xUeZJjuO xnF9v1XTJAYwMawiM8Or04UTXAkxLHv_ yUuOHE+wIuSm33Ha1v4zjiyqOIquYrB3N8 Ejin9Ec4Qtr5Pwwj-e73dT
TP_domain_2.com	backbxyqw0qkkbn6a28gtg4b.tunnel.	dmIZaqh8aok0- 3Qfz5B2Q1udf2wrJ2X3nvpd_2+9Mh0qT3+ y4iFSXKu…yZ_K0aqz0ACYF9sM7TRd+- 7eKJlih7QOEl5cfBi2quk
TP_domain_2.com	backbxyqw0qkkjdha28jdg8b.tunnel.	dlwbaqh8asifqKr5h6erYU8oB6q++_FwO6i RNpVV- AihJV9KsMJwsn9m…2D2MLO8eYFsAVZzi DoZYI9mGarqdfliljwQVAMLHHk0LHyHITS MfbT_BAVEt0iZwF+kIlC16fp1

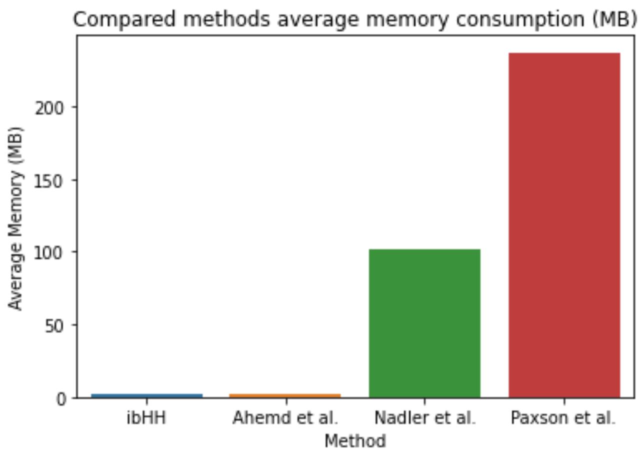
Examples of real DNS exfiltration queries detected

domain	subdomain	response
TP_domain_3.com	5F0AB605DD071A89DA7F0CF64D56FCFC.387E8330C4325908D01EDD0695F49C0A.A2523141909EE6042793C5A5ABE74428.9ED38D606EF60B8439AFA1D5EEB5182B.AA49A26C65950D76EDAAFF51E24A22AB.6EB1EF1714AFA54A884ADC9CB83EC9DC.5557B2CBE47259ACA25015EA423B6C1B.10000041.3	600ab663c9bf60d63d3b585b8ea4dee18f442c8f82676de3735cb10839dfa5ccef27771e209032d6f4703feae07eeb7a011551ab0c02a5ba0e75006e9089e2c05ef92200cc99dc4cb3ccfccae4926202142d88d50901ec409fb98e302c1d2875
TP_domain_3.com	620BB605C7DDA3F4B972F6E76AF5D4C1.8181A8FC39C43FEF218FFA3891F0A593.CEDA4CE8E0A530F320384542E06D5B49.7E8CF3B38686A5C2B9E1FB81139E892D.6616A2F9AC79B1482C891A2E895FF885.0CC811DBCEAFBD0513406E25CC931F3F.49D8436B4733B0A0771286066023EE05.10000041.3	800ab6630f605c3c55cce69532f7cbdc97f0ccce2…81f
TP_domain_3.com	7F0AB605FF19183C4759F65D438781BD.86B397A2E2DD29A06CB51B6FDDA7F753.F03DEE2EB4DFD038AC80B362C7D4FB70.78A220BF4F85502B04BB1210D92FF707.207D53598B0B4A470F8BCCA2D3E76BA7.EAE5BAB2DE292CAC4F02FD12335217D0.442DC982217C860B66E932371FAEBE83.10000041.3	a70ab663dfcf8ce89ed78c0f64953d1e55e7a57acfc2d226f624d43aba520b…7f05
TP_domain_3.com	A60AB6055189DBFC9FA928EAD924FFBC.DCA58B5CDAD104057B84E8627C5F9A96.19088BD33511DEFEA9A1FCE1D49B3CD9.46BCA5F7681E61EAE10812851A92FD30.D02A5AF882FE97D902923AC43F8D0B01.670E7776AF69092D3D7E961EBF086E35.CB0BA48E9251220EB496325B9E6A38FA.10000041.3	8d0ab66365271bdc516fd8bb…26d7c
TP_domain_3.com	680BB605F8D918950C9D5A39E4AD504E.4C095977F235232838E699BFC8991097.256DF8EE2AB2C66C46CC624E57D99C5F.8314C325AF9E811DB580A08B5EC6D12B.AD2D65449D5464B8496D3C507AF40176.65180EFD2205AAED93B081977F469EF4.1A5F094FF9F053FA8F1C4949BDFD1827.10000041.3	600ab663c9bf60d63d3b585b8ea4dee18f442c8f82676de3735cb10839dfa5ccef27771e209032d6f4703feae07eeb7a011551ab0c02a5ba0e75006e9089e2c05ef92200cc99dc4cb3ccfccae4926202142d88d50901ec409fb98e302c1d2875

Performance Evaluation

- Simulated DNS queries stream
  - 35M DNS queries total
  - Machine with a 6 core Intel CPU and 16 GB RAM





Limitations

- Only query name based exfiltration is detectable
    - Applies to most other detection methods
- Unlikely to detect exfiltration campaigns spread across many domains
    - Idea (**unverified**): instead of detecting domain heavy hitter, detect source host heavy hitters
- Cannot detect DNS exfiltration of encrypted DNS traffic
    - Such as DoH and DoT
    - Enterprises should avoid external DNS resolvers for encrypted traffic [28]
- Information counting is based only on unique subdomains

Conclusions and Future Work

- ibHH: Simple yet effective and scalable real time DNS exfiltration detection method with explainable results
- Competitive results on synthetic dataset with state-of-the-art methods
- Real-world detections with minimal false positive alerts
- Future: Deploy on real DNS resolvers
- Future: Test the ability to detect compromised hosts instead of malicious domains
    - Adjust the ibHH algorithm to detect source IP information heavy hitter instead of destination domain information heavy hitters

Questions

References

1. van Leijenhorst, T., Chin, K. W., & Lowe, D. (2008). On the viability and performance of DNS tunneling
2. Dietrich, C. J., Rossow, C., Freiling, F. C., Bos, H., Van Steen, M., & Pohlmann, N. (2011, September). On Botnets that use DNS for Command and Control. In *2011 seventh european conference on computer network defense* (pp. 9-16). IEEE
3. Black, J. (2022, November 2). *DNS: The Easiest Way to Exfiltrate Data?* https://www.akamai.com/blog/security/dns-the-easiest-way-to-exfiltrate-data
4. Nadler, A., Bitton, R., Brodt, O., & Shabtai, A. (2022). On the vulnerability of anti-malware solutions to DNS attacks. *Computers & Security*, *116*, 102687
5. E. Ekman, B. Andersson, “Iodine (ip-over-dns, ipv4 over dns tunnel)”, https://code.kryo.se/iodine/,
6. O.Dembour, “dns2tcp”, https://github.com/alex-sector/dns2tcp
7. R.Bowser, “dnscat2”, https://github.com/iagox86/dnscat2

References (cont.)

8. Revelli, A., Leidecker, N., “Heyoka”, https://heyoka.sourceforge.net/, 2009
9. Lawrence, D., Riley, M., Software used by Home Depot hackers different from Target attack. (2014, September 11). Toronto Star, https://www.thestar.com/business/tech_news/2014/09/11/home_depots_breach_could_be_among_the_largest_to_date.html
10. Dietrich, C. J. (2011, September 2). Feederbot Botnet Using DNS as Carrier for Command and Control (C2). Prof. Dr. Christian J. Dietrich. https://chrisdietri.ch/post/feederbot-botnet-using-dns-command-and-control/
11. Falcone, R. (2017, November 8). OilRig Deploys “ALMA Communicator” – DNS Tunneling Trojan. Unit 42. https://unit42.paloaltonetworks.com/unit42-oilrig-deploys-alma-communicator-dns-tunneling-trojan
12. Spring, T. (2021, September 16). Wekby APT Gang Using DNS Tunneling for Command and Control. https://threatpost.com/wekby-apt-gang-using-dns-tunneling-for-command-and-control/118303/
13. Ransomware Actors Leaning on DNS Tunneling. (2022, June 9). Decipher. https://duo.com/decipher/ransomware-actors-leaning-on-dns-tunneling
14. Paxson, V., Christodorescu, M., Javed, M., Rao, J. R., Sailer, R., Schales, D. L., ... & Weaver, N. (2013, August). Practical Comprehensive Bounds on Surreptitious Communication over DNS. In USENIX Security Symposium (pp. 17-32)
15. Nadler, A., Aminov, A., & Shabtai, A. (2019). Detection of malicious and low throughput data exfiltration over the DNS protocol. Computers & Security, 80, 36-53

References (cont.)

16. Ahmed, J., Gharakheili, H. H., Raza, Q., Russell, C., & Sivaraman, V. (2019, April). Real-time detection of DNS exfiltration and tunneling from enterprise networks. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)* (pp. 649-653). IEEE
17. Wang, Y., Zhou, A., Liao, S., Zheng, R., Hu, R., & Zhang, L. (2021). A comprehensive survey on DNS tunnel detection. *Computer Networks*, *197*, 108322
18. Babcock, B., Babu, S., Datar, M., Motwani, R., & Widom, J. (2002, June). Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 1-16)
19. Nelson, J. (2012). Sketching and streaming algorithms for processing massive data. *XRDS: Crossroads, The ACM Magazine for Students*, *19*(1), 14-19
20. Cormode, G., & Muthukrishnan, S. (2005). An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, *55*(1), 58-75
21. Flajolet, P., Fusy, É., Gandouet, O., & Meunier, F. (2007, June). Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. In *Discrete Mathematics and Theoretical Computer Science* (pp. 137-156). Discrete Mathematics and Theoretical Computer Science

References (cont.)

22. Basat, R. B., Chen, X., Einziger, G., & Rottenstreich, O. (2020). Designing heavy-hitter detection algorithms for programmable switches. *IEEE/ACM Transactions on Networking*, *28*(3), 1172-1185
23. Afek, Y., Bremler-Barr, A., Cohen, E., Feibish, S. L., & Shagam, M. (2016). Efficient distinct heavy hitters for DNS DDoS attack detection. *arXiv preprint arXiv:1612.02636*
24. Kremez, V. (2021, September 2). *FIN6 “FrameworkPOS”: Point-of-Sale Malware Analysis & Internals - SentinelLabs*. SentinelOne. https://www.sentinelone.com/labs/fin6-frameworkpos-point-of-sale-malware-analysis-internals
25. Yunakovsky, S., Pomerantsev, I., (2019, July 11). *Denis and Co.* Securelist. https://securelist.com/denis-and-company/83671/
26. Sammour, M., Hussin, B., Othman, M. F. I., Doheir, M., AlShaikhdeeb, B., & Talib, M. S. (2018). DNS tunneling: a review on features. *International Journal of Engineering and Technology*, *7*(3.20), 1-5
27. Gatlan, S. (2022, March 15). *New Linux botnet exploits Log4J, uses DNS tunneling for comms*. BleepingComputer. https://www.bleepingcomputer.com/news/security/new-linux-botnet-exploits-log4j-uses-dns-tunneling-for-comms
28. Agency, N.S.: Adopting Encrypted DNS in Enterprise Environments. https://media.defense.gov/2021/Jan/14/2002564889/-1/-1/0/CSI_ADOPTING_ENCRYPTED_DNS_U_OO_102904_21.PDF (2021)
29. Pochat, V. L., Van Goethem, T., Tajalizadehkhoob, S., Korczyński, M., & Joosen, W. (2018). Tranco: A research-oriented top sites ranking hardened against manipulation. *arXiv preprint arXiv:1806.01156*