# Information-Based Heavy Hitters for Real-Time DNS Exfiltration Detection

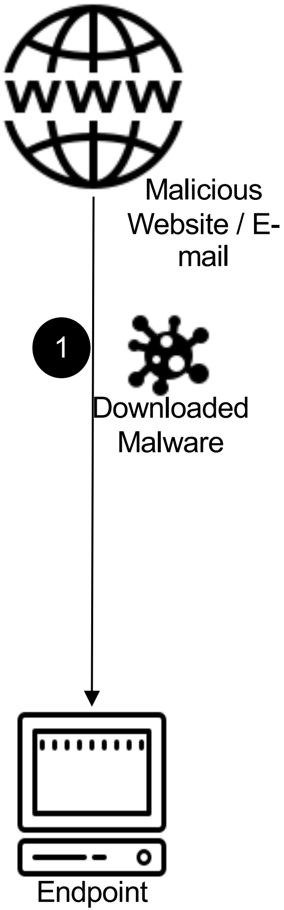
Yarin Ozery (Ben-Gurion University of the Negev, Akamai)
Asaf Nadler (Ben-Gurion University of the Negev)
Asaf Shabtai (Ben-Gurion University of the Negev)

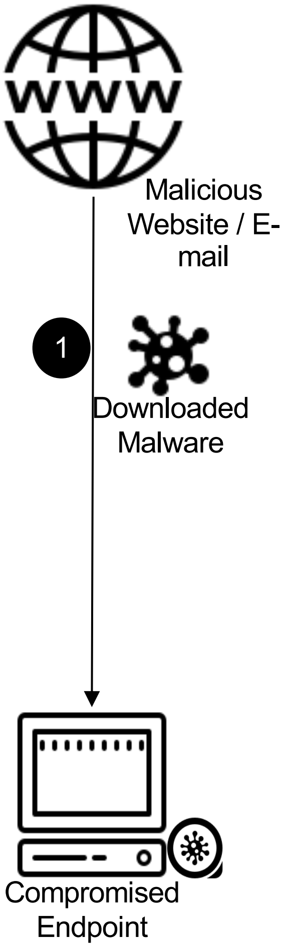
**Presented By: Yarin Ozery**

# Every Year, A New DNS Exfiltration Malware Unveiled

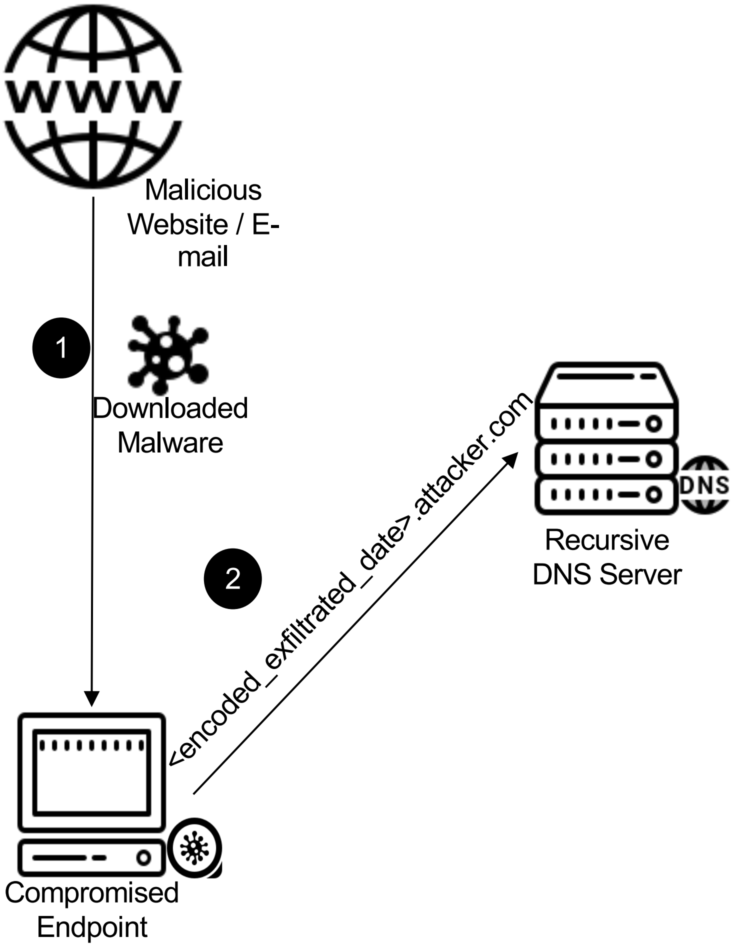
# >25 Years of DNS Data Exfiltration

- DNS Exfiltration/Tunnelling was first described in 1998
- >100 Academic Papers on DNS tunnelling and detection
  - Rarely tested on real-world, large-scale datasets
  - Aren’t applicable in real-time
  - Complicated for implementation

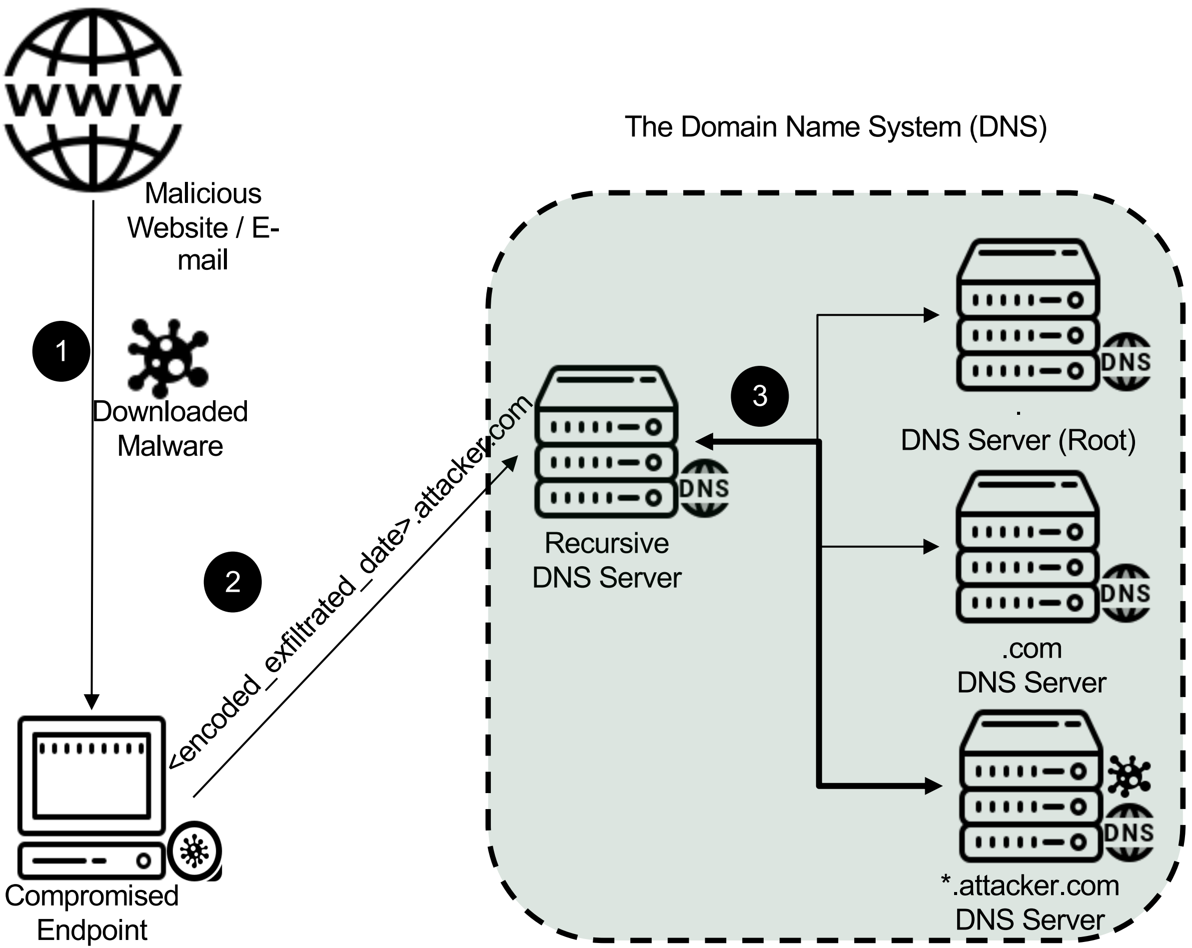

Malicious Website / E-mail
1
Downloaded Malware
Endpoint



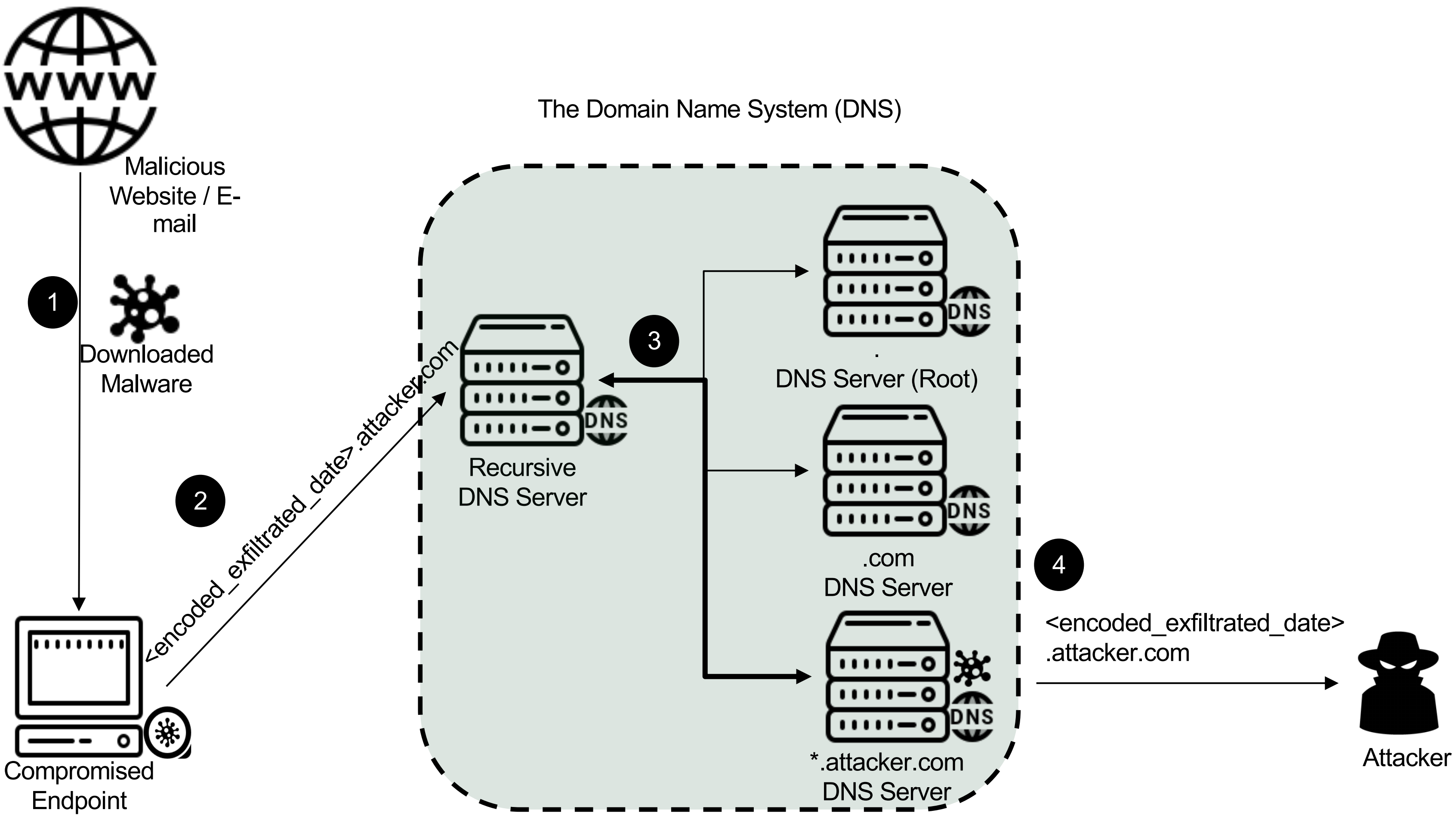
Malicious Website / E-mail
1
Downloaded Malware
Compromised Endpoint

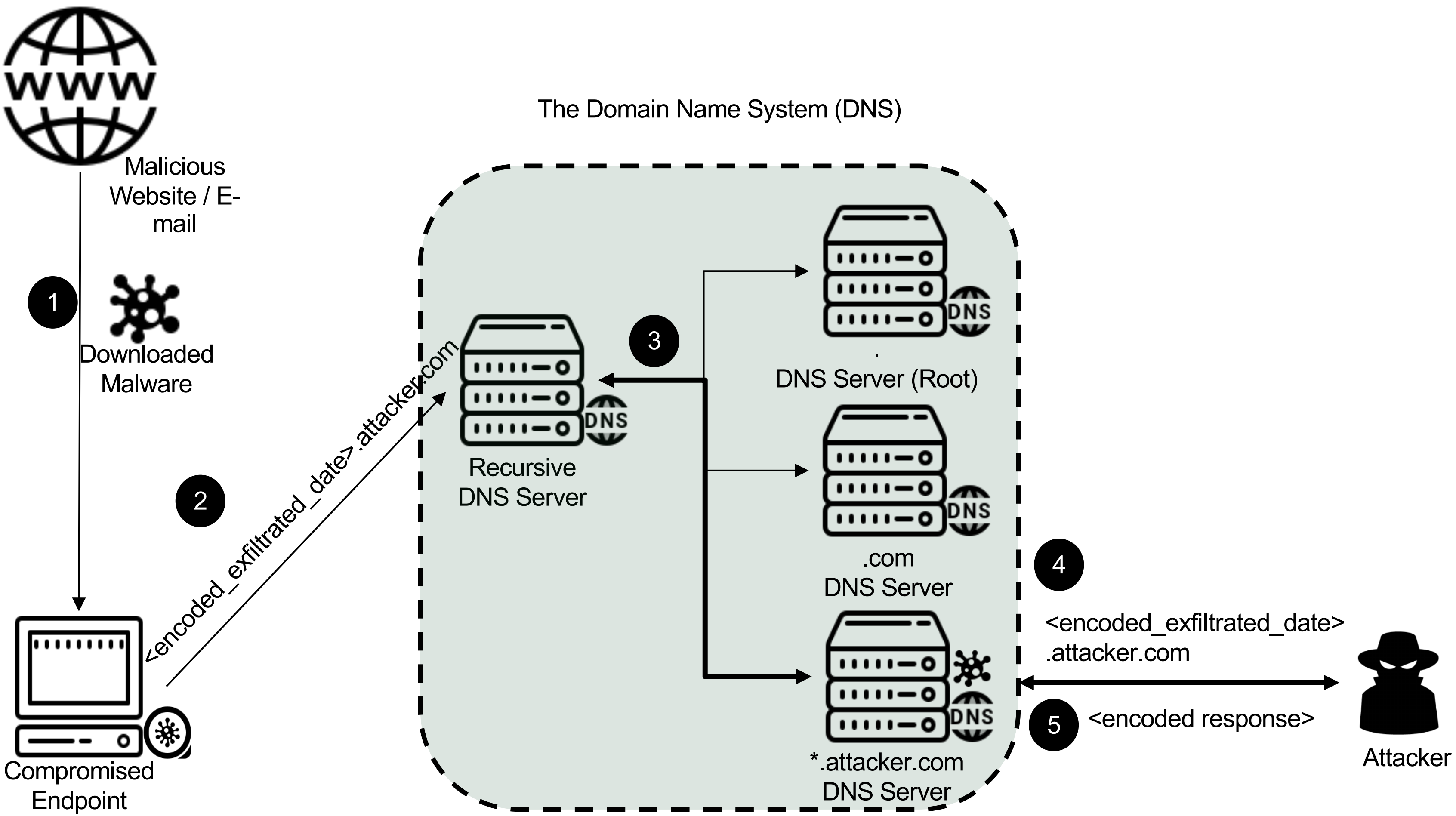


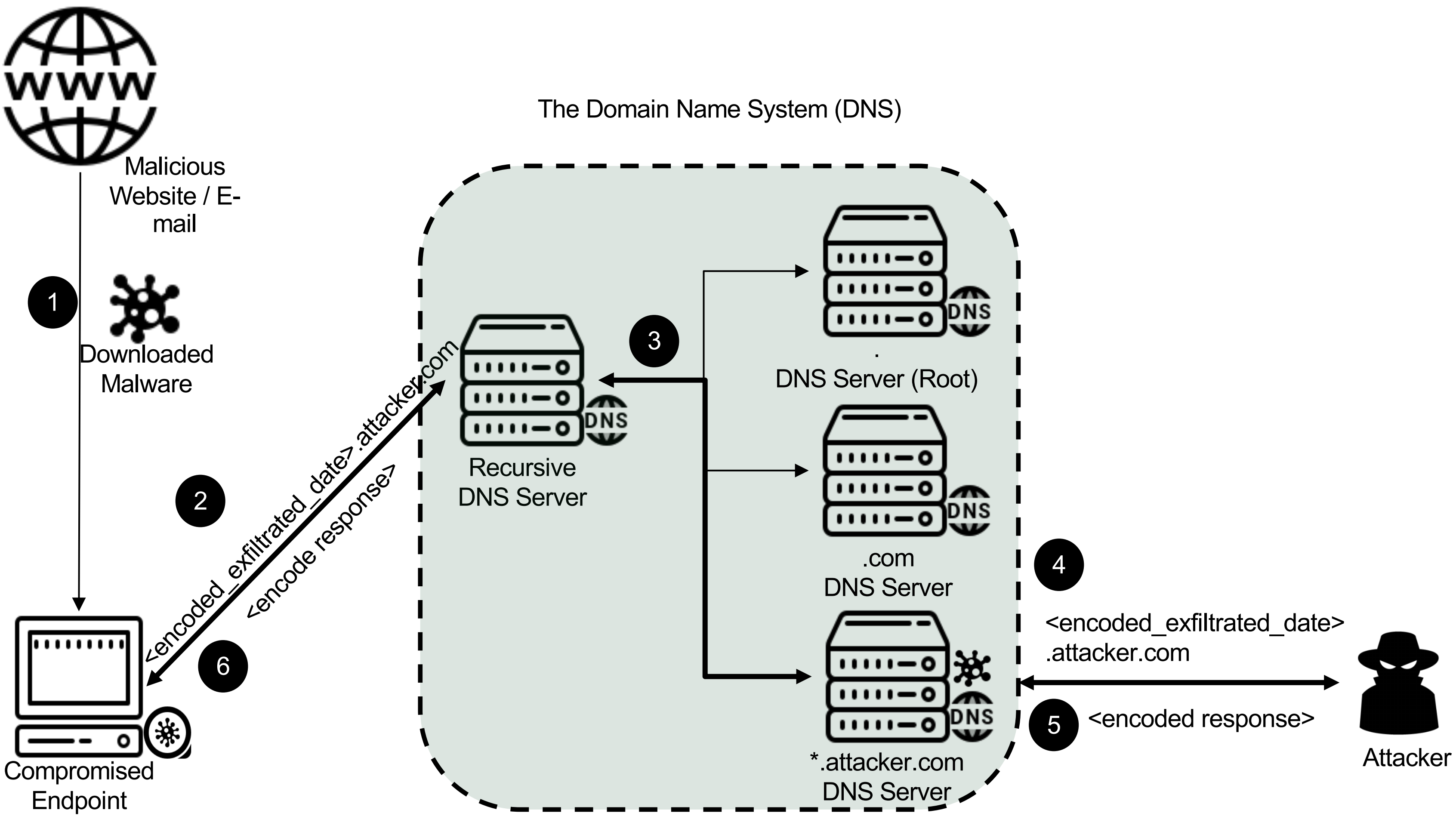
Malicious Website / E-mail
1
Downloaded Malware
2
<encoded_exfiltrated_date>.attacker.com
DNS
Recursive DNS Server
Compromised Endpoint



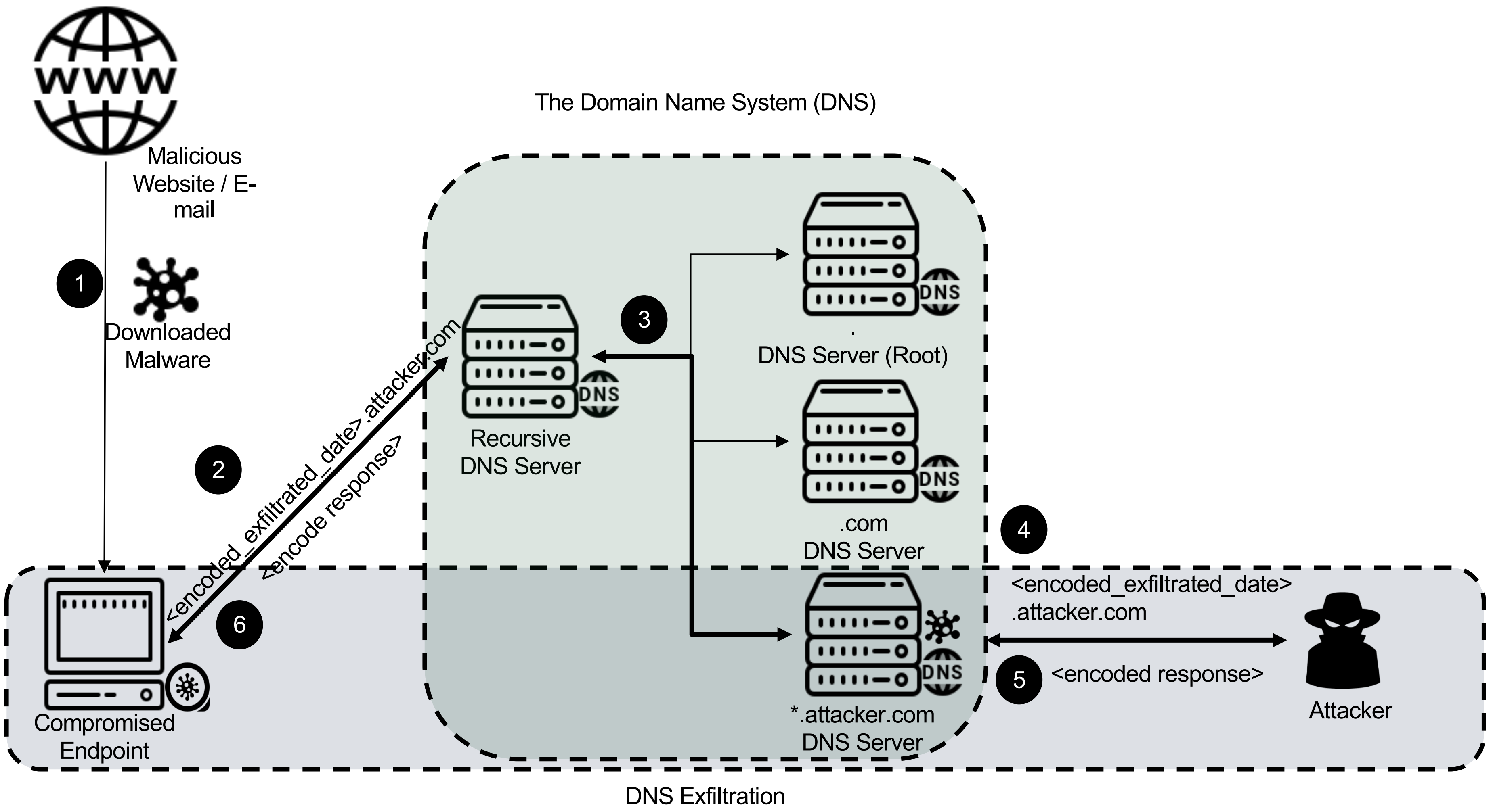
The Domain Name System (DNS)
Malicious Website / E-mail
1
Downloaded Malware
2
<encoded_exfiltrated_date>.attacker.com
3
Recursive DNS Server
DNS
. DNS Server (Root)
.com DNS Server
*.attacker.com DNS Server
Compromised Endpoint


The Domain Name System (DNS)
Malicious Website / E-mail
1
Downloaded Malware
2
<encoded_exfiltrated_date>.attacker.com
Recursive DNS Server
3
.
DNS Server (Root)
.com DNS Server
*.attacker.com DNS Server
4
<encoded_exfiltrated_date> .attacker.com
Compromised Endpoint
Attacker
DNS


The Domain Name System (DNS)
Malicious Website / E-mail
1
Downloaded Malware
2
<encoded_exfiltrated_date>.attacker.com
Compromised Endpoint
Recursive DNS Server
3
.
DNS Server (Root)
.com
DNS Server
*.attacker.com
DNS Server
4
<encoded_exfiltrated_date>
.attacker.com
5
<encoded response>
Attacker



The Domain Name System (DNS)
Malicious Website / E-mail
1
Downloaded Malware
2
<encoded_exfiltrated_date>.attacker.com
<encode response>
6
Compromised Endpoint
Recursive DNS Server
3
. DNS Server (Root)
.com DNS Server
*.attacker.com DNS Server
4
<encoded_exfiltrated_date> .attacker.com
5
<encoded response>
Attacker


The Domain Name System (DNS)
Malicious Website / E-mail
1
Downloaded Malware
2
<encoded_exfiltrated_date>.attacker.com
<encode response>
Recursive DNS Server
3
.
DNS Server (Root)
.com
DNS Server
4
<encoded_exfiltrated_date>
.attacker.com
5
<encoded response>
*.attacker.com
DNS Server
Attacker
6
Compromised Endpoint
DNS Exfiltration

# Our Paper: A New Method of DNS Exfiltration

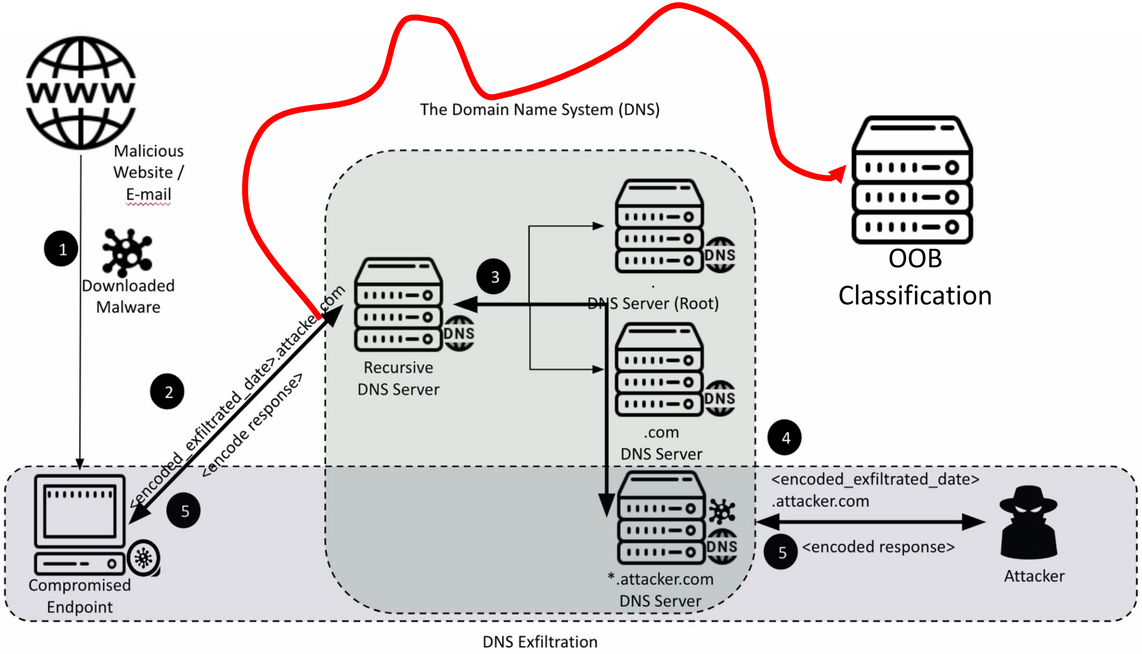
- Applicable in Real-time
  - **X15 faster than SOTA [Ahmed2019]**, on a DNS Server benchmark
  - **Inline speed processing** of >600k queries/sec

- Tested at Scale, on Real-World Data
  - *Ziza* 2023, 35M DNS queries, Largest Public Dataset
  - Akamai 2023, 255B DNS queries, 750 Orgs, **Largest Ever Evaluated**

- Simple Implementation
  - Designed to be easily implemented on standard BIND DNS servers
  - Now being deployed on Akamai world-wide DNS network
  - Open source: https://shorturl.at/goUW5

# Prior Work isn't Designed for Real-Time

- Prior work relies on OOB feature collection and classification limiting real-time detection. Why?
  - DNS is a critical service, which must not be slowed down
  - Statistical methods calculating the baseline and history



# Information-based Heavy Hitters

- Basic Idea for Real-time:
  - Perform the detection directly on the DNS resolver for every new DNS query
  - Quantity the amount of information transmitted in DNS queries for every registered domain
  - Raise an alert if the amount exceeds a threshold

- Challenge:
  - Requires **memory and computation linear to the number of DNS queries**
  - Under an attack can be an overwhelming amount for a DNS resolver

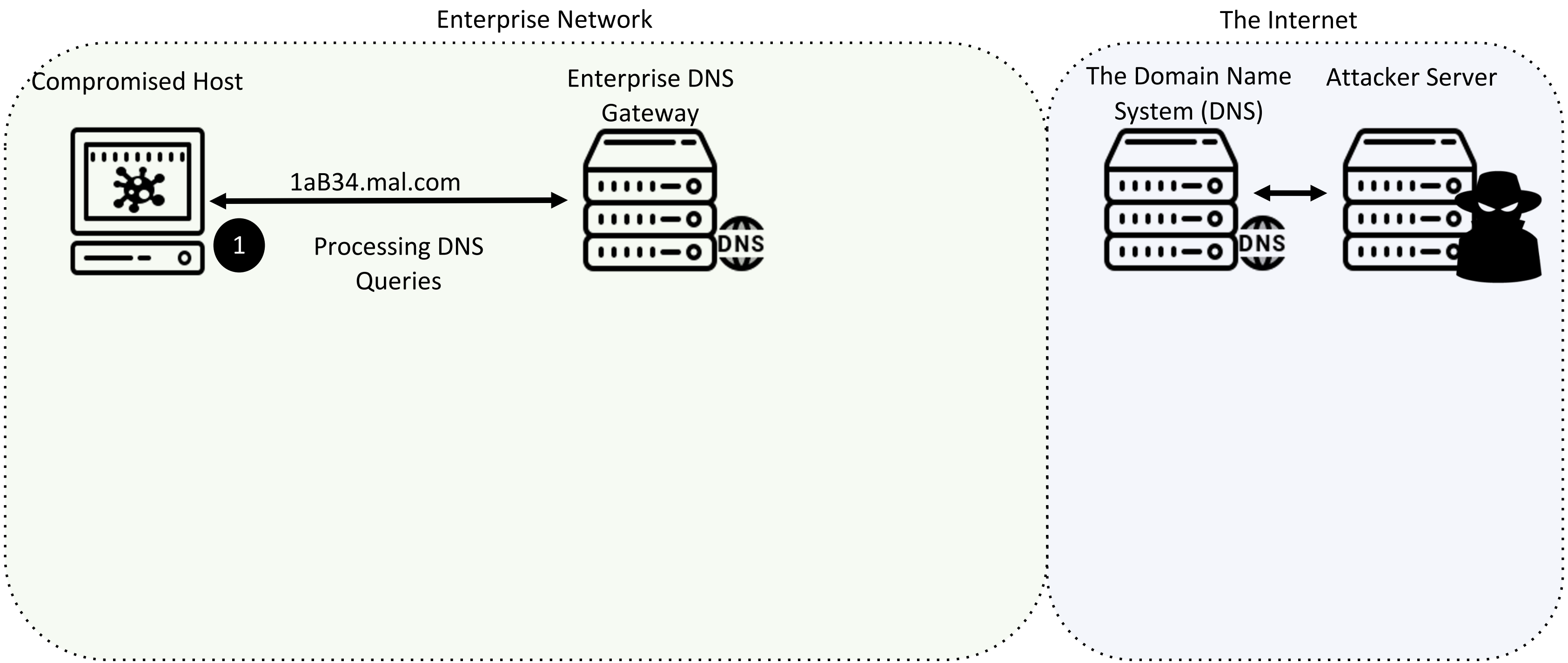
- Solution:
  - Approximation using probabilistic cardinality estimation algorithms
  - **Constant memory and computation (*)**, to suit DNS resolvers
  - Modeled as a weighted variation of the distinct heavy hitter detection problem [Venkataraman 2004, Afek 2016]

# Information-based Heavy Hitters

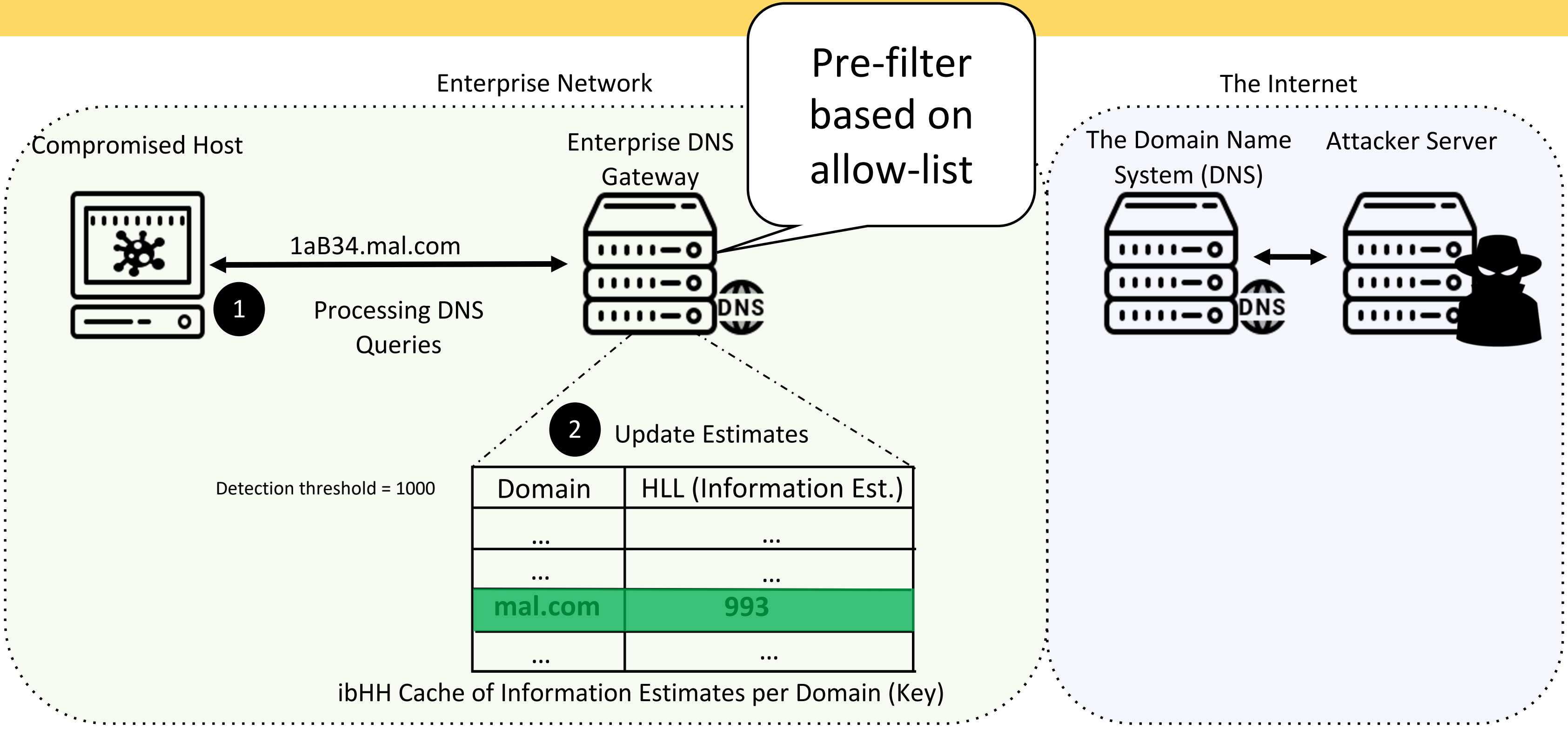
- The **cache** data-structure behind the method:
  - Behaves as a priority queue of “suspected domains”
  - A fixed number of entries (K)
  - Higher K values improves accuracy
  - Every entry consists of a counting estimation
- At every point in time:
  - The cache consists the K most “suspected” domains
  - Higher information transmitted is more suspicious
- For every new DNS query:
  - Quantify the amount of information for the domain
  - If quantification is more “suspected” than currently monitored domains, update the cache by popping the least “suspected” domain
- Resets every constant time to reduce threshold of entry (e.g., 2 minutes)

| Key (domain) | Value |
|---|---|
| Example_1.com | Seed: 0.9<br>HLL++ instance |
| … | … |
| Example_k.com | Seed: 0.45<br>HLL++ instance |

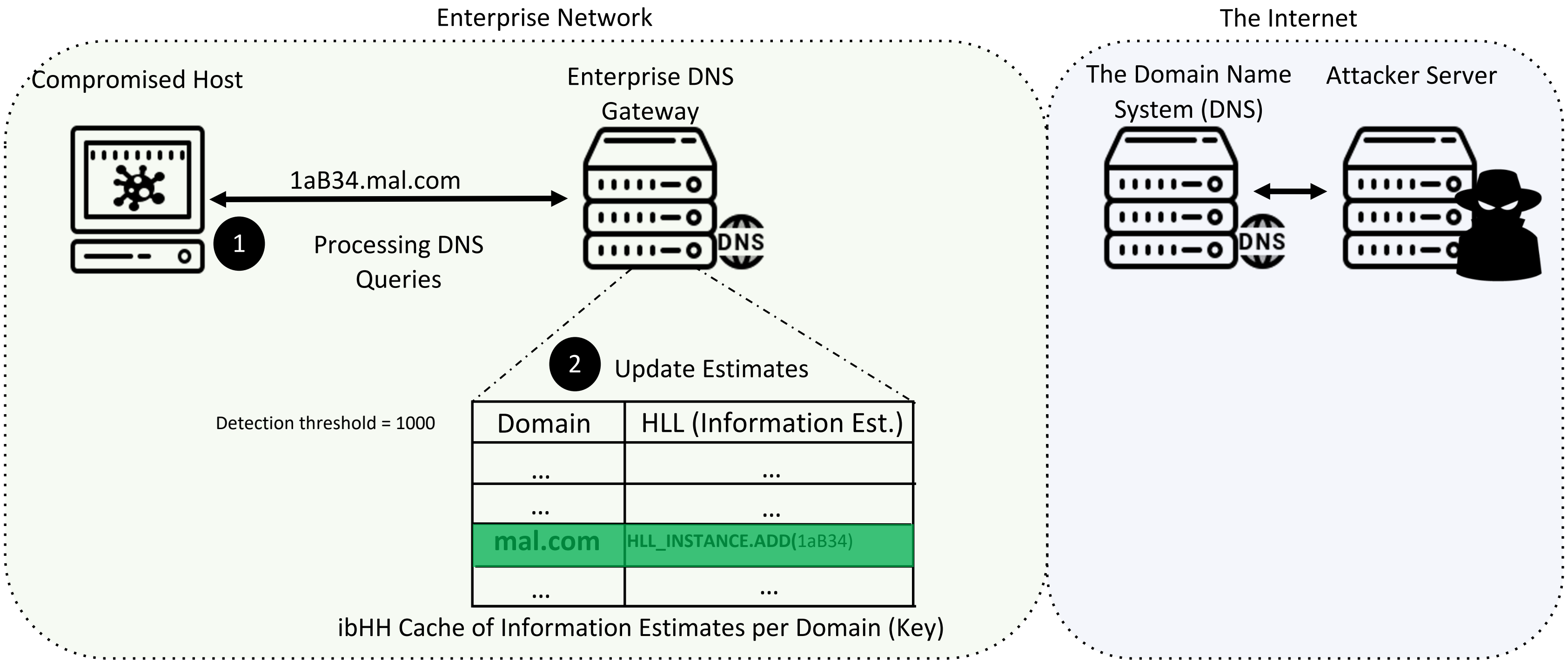
# ibHH in Action



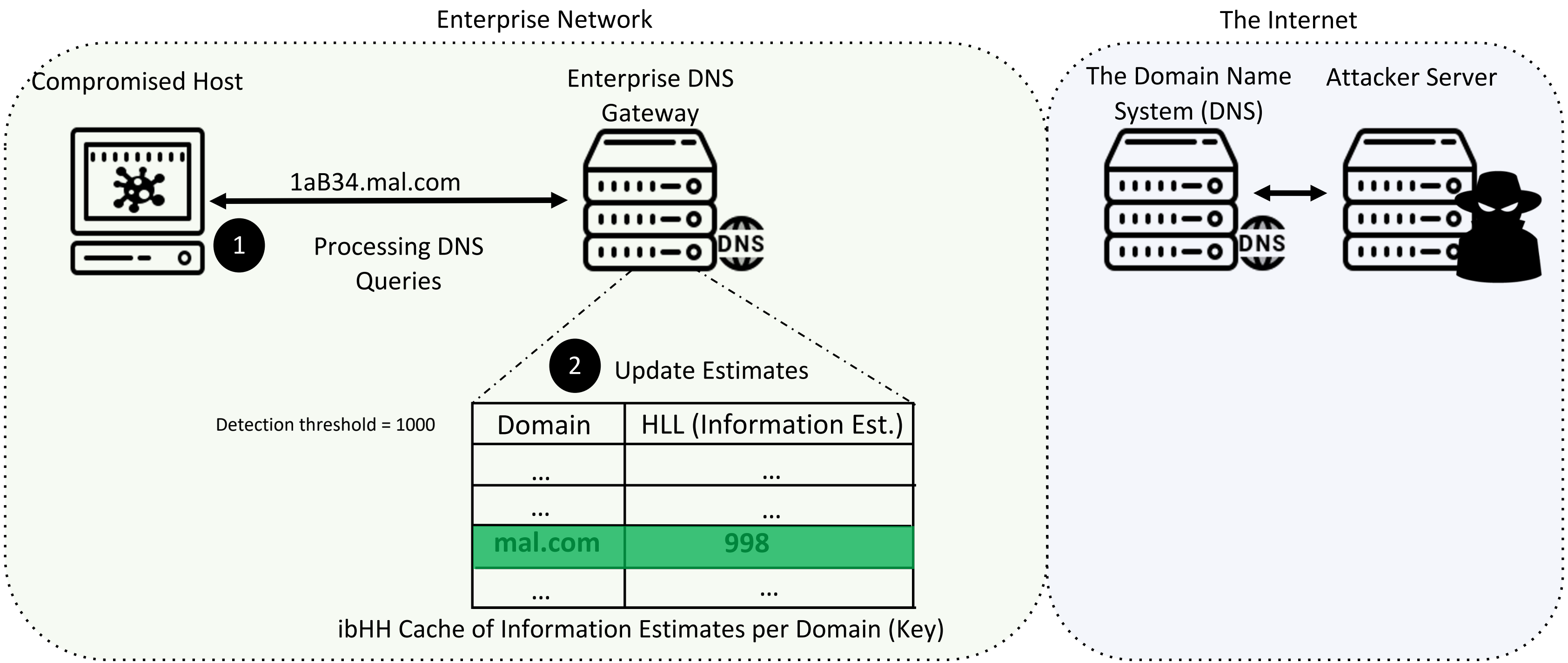
# ibHH in Action



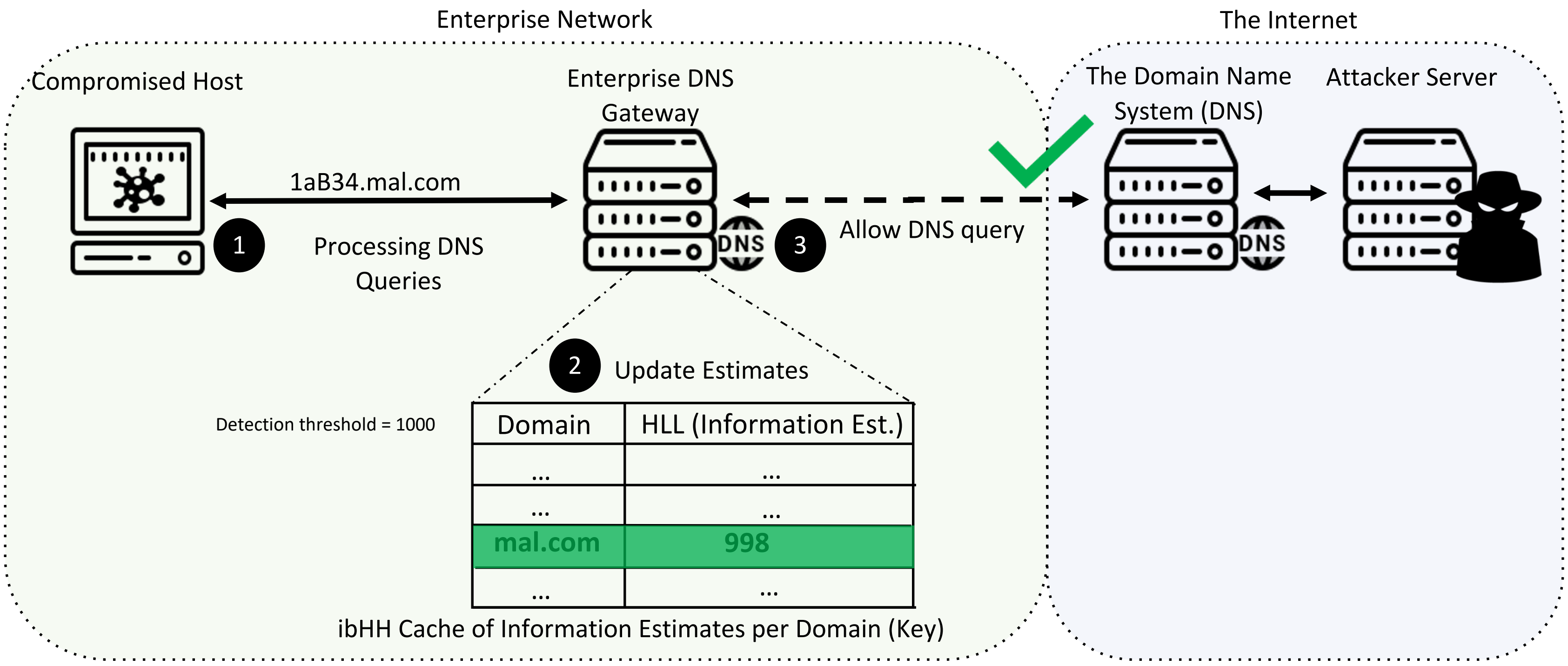
# ibHH in Action



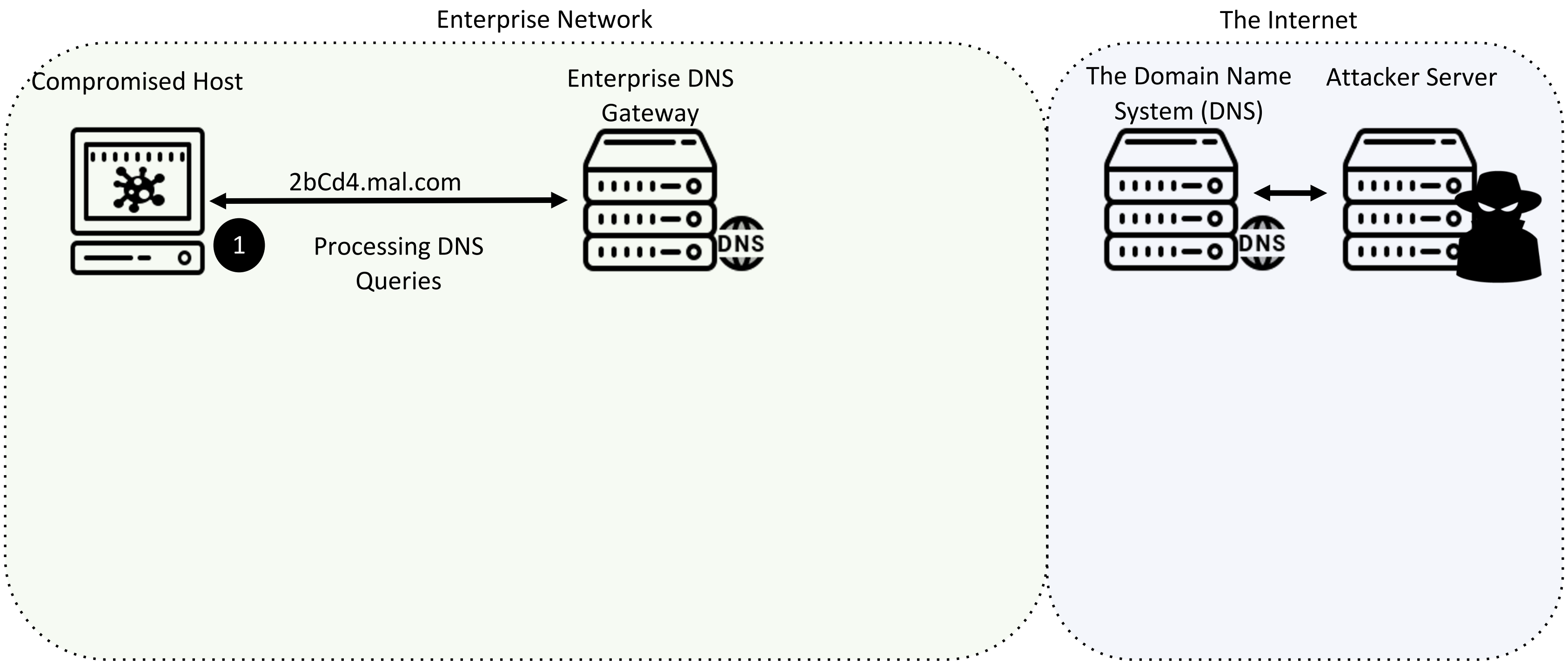
# ibHH in Action



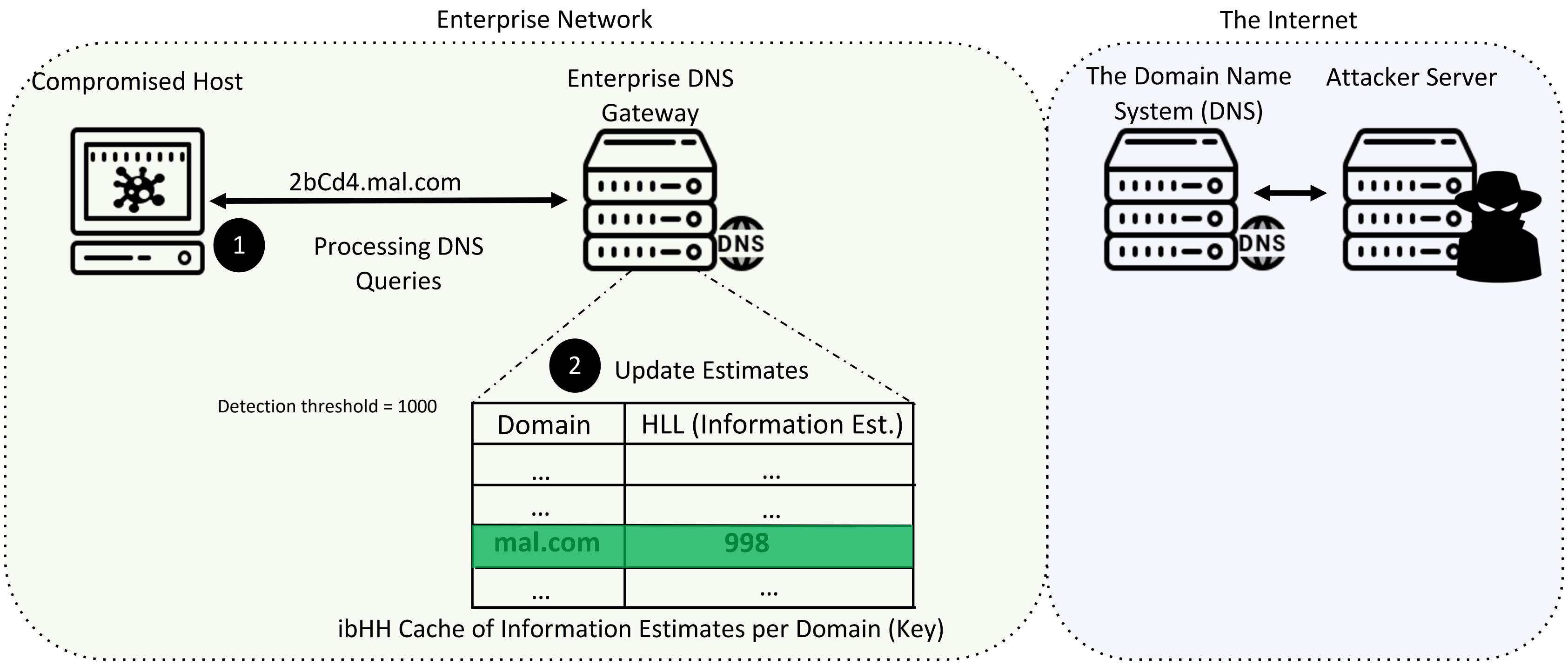
# ibHH in Action



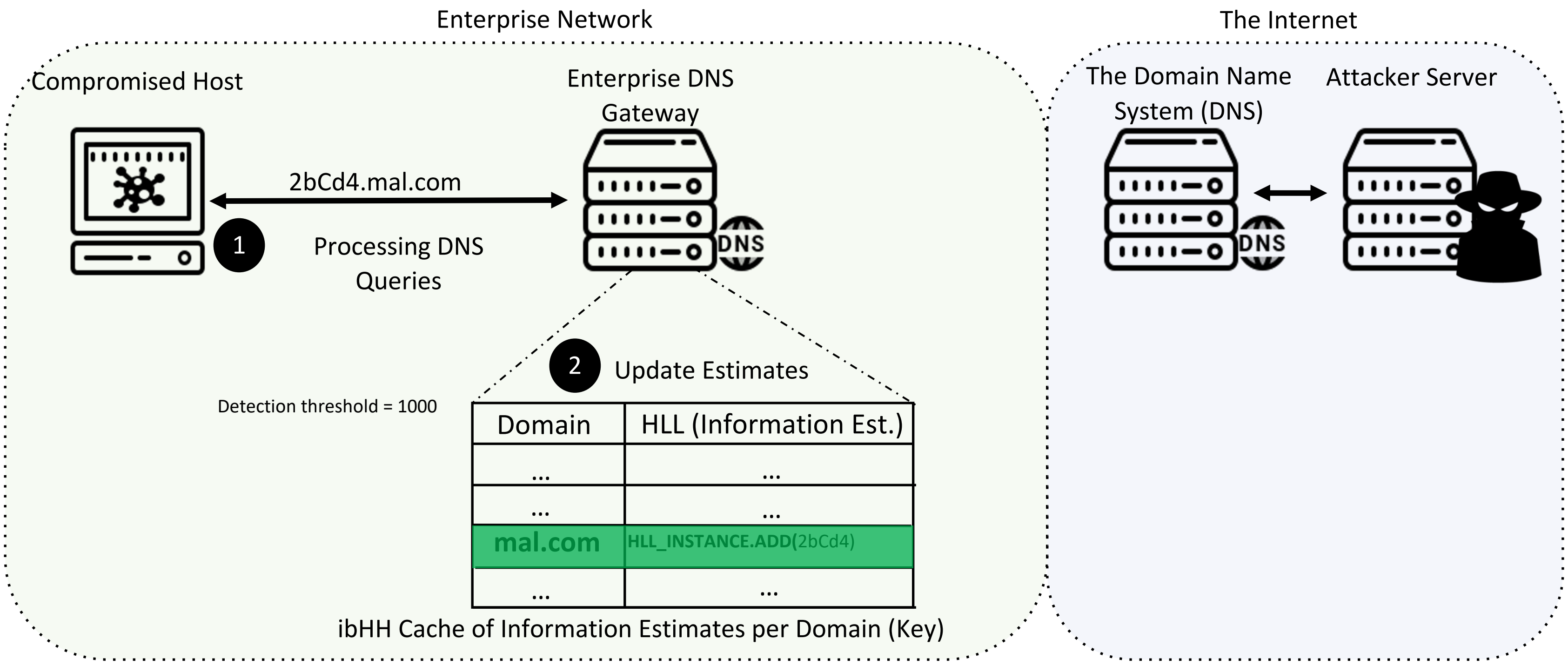
# ibHH in Action



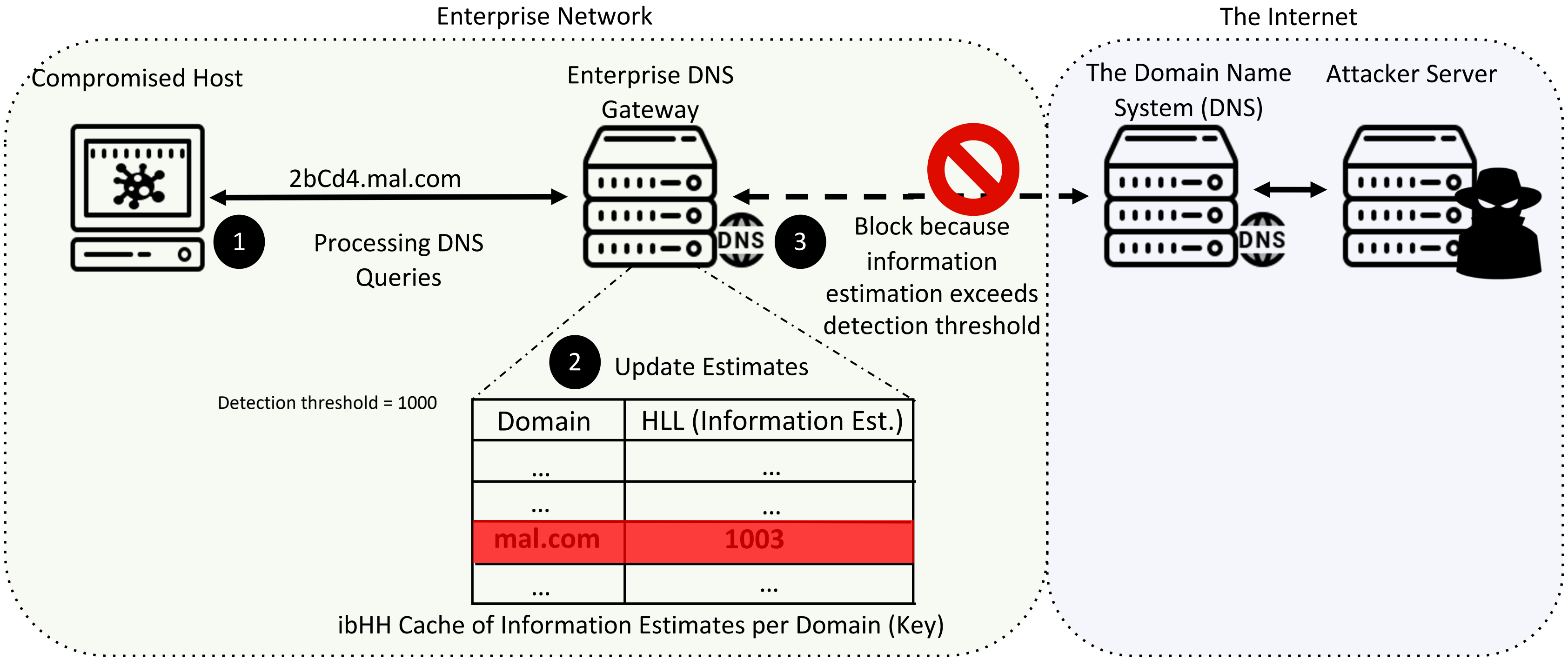
# ibHH in Action



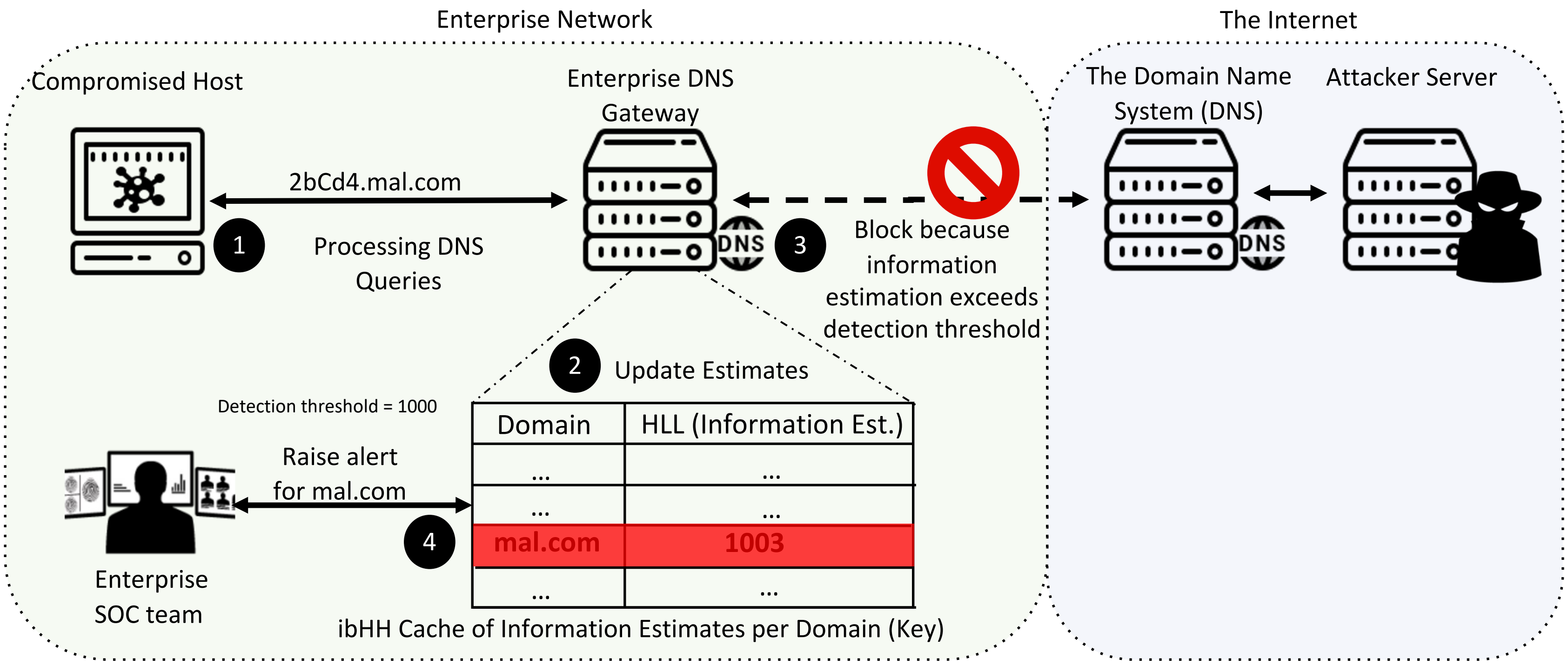
# ibHH in Action



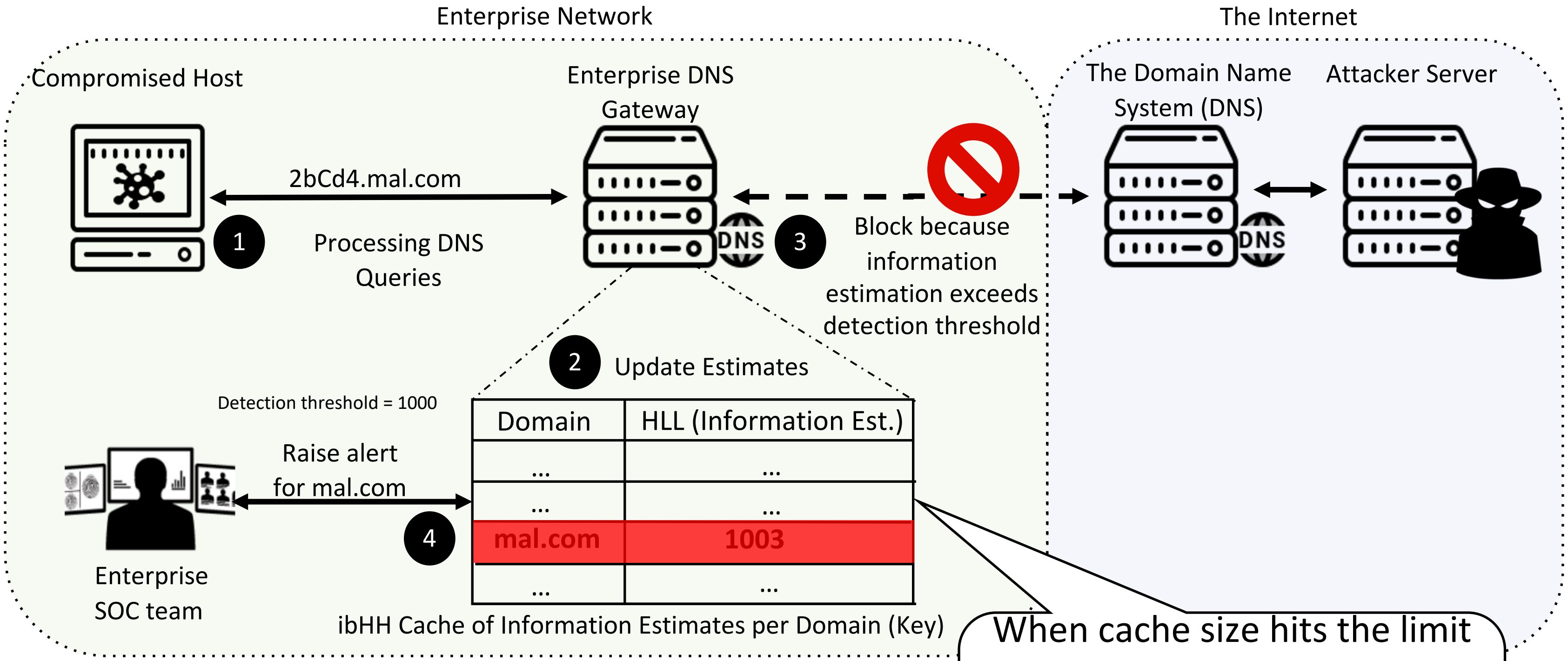
# ibHH in Action



# ibHH in Action



# ibHH in Action




When cache size hits the limit and a new domain needs to be inserted, the least *information-heavy* domain is evicted

# Distinguishing Benign from Malicious Cases

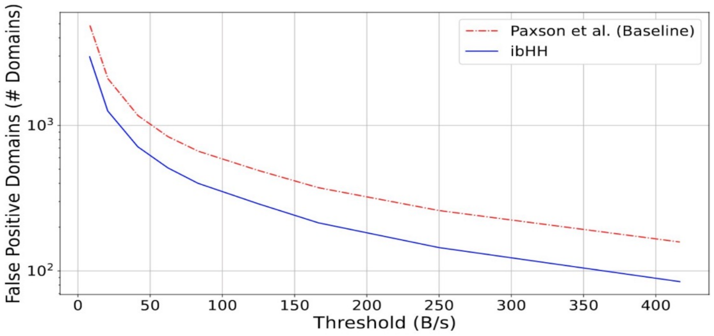
- **Domains that transmit a high amount of information aren’t necessarily malicious**
  - For instance: DNSBL, UGC [Nadler2022]
  - There’s no clear-cut method to telling malice based on DNS data alone

- **Use of Global Allow Lists**:
  - Ignore domains included in Alexa top 1M and TRANCO
  - A popular go-to for reducing false alerts in cybersecurity

- **Peace-time/War-time**:
  - Run the ibHH algorithm in the enterprise network to identify benign “information heavy hitter” domains, collect them into a list to be used as an allow-list
  - Our evaluation shows that >90% of the distinct domains observed on 750 enterprise organizations over a week are observed in the first day

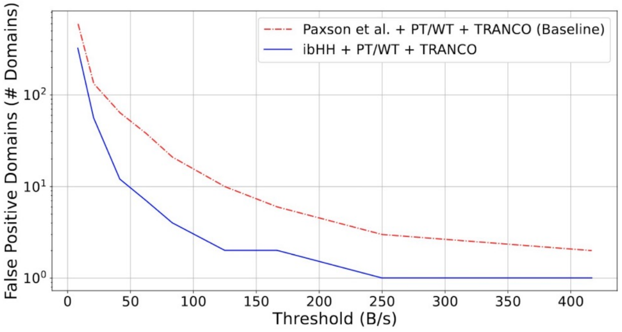
# Setting the Alert Threshold

- “Human readable”: The approximate number of bytes transmitted to raise an alert
- Trade-off between number of false alerts to sensitivity
- Recommended setting: tune in peace-time to obtain an acceptable false positive rate




(b) False positive domains

Fig. 3: Parameter tuning without allowlists.




(b) False positive domains

Fig. 6: Parameter tuning with TRANCO & peacetime allowlist.

# Evaluation Datasets and Compared Methods

- Datasets:

| Dataset Name | # DNS Queries | # Unique 2LD | # Enterprise Organizations | # Client Hosts | Collection period |
|---|---|---|---|---|---|
| $DS_f$ | 50B | 43M | 753 | N/A | 8 Days |
| $DS_p$ | 5B | 668K | 223 | 129K | 8 Days |
| $Ziza$ | 35M | 12.8K | N/A | 35K | 26 Hours |
| $DS_r$ | **255B** | **463M** | 753 | N/A | 21 Days |

- Compared methods:

| Method | Summary | Real-time capable | Year |
|---|---|---|---|
| ibHH | Real-time Information estimation | ✓ | 2024 |
| Paxson (Paxson et al.) | Information estimation | ✗ | 2013 |
| IF (Nadler et al.) | Traffic analysis, isolation forest, 6 features | ✗ | 2019 |
| RT-IF (Ahmed et al.) | Query analysis, isolation forest, 8 features | ✓ | 2019 |

# Evaluation Methodology

- 4 compared methods trained under different acceptable FPR: 1/100, 1/1000, 1/10,000, 1/100,000
- Methodology inspired by [Nadler2019, Daihes2021]
- Split the datasets across time into 3 parts: Train, Peace-time generation, Test
- Injected synthetic DNS exfiltration traffic into the test dataset:
  - 1% (1,300) of the client hosts are “infected”
  - Iodine (open-source DNS tunneling software)
  - FrameworkPOS
  - Backdoor.Win32.Denis
- Measuring the TPR and FPR of each method
  - Based on the count of **registered domains** alerts
  - TRANCO top 1m allow-list was used for all methods

# Results: ibHH outperforms SOTA

TABLE V: Comparison of the evaluated methods based on the TPR and FPR.

| Method | Dataset | FPR=0.01 | | | | FPR=0.001 | | | | FPR=0.0001 | | | | FPR=0.00001 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $TD^1$ | FPR | TPR | $DER^1$ | $TD^1$ | 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 | 1300 | 0 | **1.0** | 275 |
| ibHH | $DS_p + F$ | 1743 | **0.0038** | **1.0** | 0.7 | 1430 | **0.001** | **1.0** | 5 | 1298 | <0.001 | **0.98** | 65 | 1280 | 0 | **0.97** | 275 |
| | $DS_p + D$ | 1728 | **0.0037** | **1.0** | 0.7 | 1417 | **0.001** | **1.0** | 5 | 1252 | <0.001 | **0.98** | 65 | 1214 | 0 | 0.92 | 275 |
| | $ZIZA$ | 65 | **0.005 (62)** | **1.0 (3)** | 0.6 | 12 | **0.0007 (9)** | 1.0 (3) | 4 | 4 | 0.000085 (1) | **1.0 (3)** | 15 | N/A | N/A | N/A | N/A |
| | $DS_p + I$ | 3015 | 0.007 | **1.0** | | 2132 | 0.0012 | **1.0** | | 1342 | <0.001 | **1.0** | | 1300 | 0 | **1.0** | |
| IF | $DS_p + F$ | 3015 | 0.007 | 0.99 | N/A | 2085 | 0.0012 | 0.96 | N/A | 1267 | <0.001 | **0.98** | N/A | 1279 | 0 | **0.97** | N/A |
| | $DS_p + D$ | 3015 | 0.007 | 0.98 | | 2058 | 0.0012 | 0.94 | | 1240 | <0.001 | 0.97 | | 1183 | 0 | 0.91 | |
| | $ZIZA$ | 143 | 0.012 (140) | **1.0 (3)** | | 24 | 0.0017 (22) | 0.67 (2) | | 1 | **0.0 (0)** | 0.33 (1) | | N/A | N/A | N/A | |
| | $DS_p + I$ | 3200 | 0.008 | **1.0** | | 2659 | 0.014 | **1.0** | | 1314 | <0.001 | **1.0** | | 1250 | 0 | 0.96 | |
| RT-IF | $DS_p + F$ | 3214 | 0.008 | **1.0** | N/A | 2631 | 0.014 | 0.98 | N/A | 1107 | <0.001 | 0.85 | **N/A** | 0 | 0 | 0 | N/A |
| | $DS_p + D$ | 3170 | 0.008 | 0.98 | | 2599 | 0.014 | 0.95 | | 1039 | <0.001 | 0.8 | | 0 | 0 | 0 | |
| | $ZIZA$ | 122 | 0.01 (119) | **1.0 (3)** | | 21 | 0.015 (19) | 0.67 (2) | | 0 | **0.0 (0)** | 0.0 (0) | | N/A | N/A | N/A | |
| | $DS_p + I$ | 1927 | 0.0041 | **1.0** | 0.9 | 1771 | 0.0023 | **1.0** | 12 | 1314 | <0.001 | **1.0** | 70 | 1300 | 0 | **1.0** | 300 |
| Paxson | $DS_p + F$ | 1927 | 0.0041 | **1.0** | 0.9 | 1771 | 0.0023 | **1.0** | 12 | 1249 | <0.001 | 0.96 | 70 | 1270 | 0 | 0.96 | 300 |
| | $DS_p + D$ | 1927 | 0.0041 | 0.98 | 0.9 | 1771 | 0.0023 | **1.0** | 12 | 1230 | <0.001 | 0.95 | 70 | 932 | 0 | 0.72 | 300 |
| | $ZIZA$ | 87 | 0.0071 (84) | **1.0 (3)** | 1 | 14 | 0.0009 (11) | **1.0 (3)** | 6 | 3 | 0.000085 (1) | 0.67 (2) | 32 | N/A | N/A | N/A | N/A |

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

# Longitudinal Analysis on Real-World Traffic

- Running on large-scale, real-world traffic of 750 enterprise organizations
  - 255B DNS queries
  - 21 days of traffic

- 2 Real-world attacks detected in **real-time**
  - Open-source DNS tunneling tool
  - Low-throughput attack simulation by a cybersecurity company

- 3 out of 4 methods detected the 2 attacks, ibHH with the least number of FPs (13% less than second-best method)

TABLE VI: Real-world evaluation results.

| Method | FP Domains | TP Domains | FP Queries | TP Queries | $DER$ |
|---|---|---|---|---|---|
| ibHH | **15** | **2** | 2,043 | 17,441 | 6 |
| IF | 31 | **2** | 57,125 | 17,820 | N/A |
| RT-IF | 20 | 1 | 5,093 | 12,391 | N/A |
| Paxson | 17 | **2** | 2,677 | 15,570 | 11 |

# Conclusions & Future Work

- Simple and effective real-time DNS exfiltration detection method
- Designed to be deployed on DNS resolver for real-time detection
- DNS exfiltration detection capabilities outperform SOTA
- Future work:
  - Currently being deployed on Akamai’s DNS resolvers
  - Longitudinal analysis over a year

# Conclusions & Future Work

- Simple and effective real-time DNS exfiltration detection method
- Designed to be deployed on DNS resolver for real-time detection
- DNS exfiltration detection capabilities outperform SOTA
- Future work:
    - Currently being deployed on Akamai’s DNS resolvers
    - Longitudinal analysis over a year

Questions?

yarinoz@post.bgu.ac.il

Source Code