Anomaly Detection of DHCP Starvation Attacks Using a Probabilistic Approach

Radu-Constantin Onuțu

Computer Science Department
University of Bucharest

Introduction

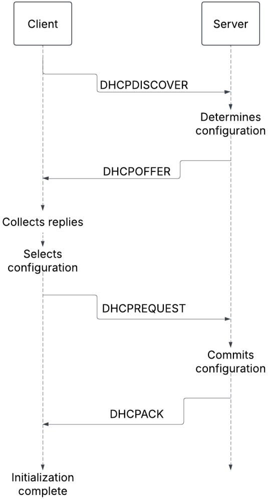
- Reference: Tripathi, Nikhil, and Neminath Hubballi. "A probabilistic anomaly detection scheme to detect DHCP starvation attacks." 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). IEEE, 2016.
- How does DHCP work and how can someone exploit it?

Introduction

- Reference: Tripathi, Nikhil, and Neminath Hubballi. "A probabilistic anomaly detection scheme to detect DHCP starvation attacks." 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). IEEE, 2016.
- How does DHCP work and how can someone exploit it?

DHCP Overview

- DHCP is a protocol for providing IP addresses to devices on a network.



DHCP Starvation Attack

There are 2 main types of DHCP Starvation Attacks:

- Classical DHCP Starvation Attack
- Induced DHCP Starvation Attack

DHCP Starvation Attack

There are 2 main types of DHCP Starvation Attacks:

- Classical DHCP Starvation Attack
- Induced DHCP Starvation Attack

DHCP Starvation Attack

There are 2 main types of DHCP Starvation Attacks:

- Classical DHCP Starvation Attack
- Induced DHCP Starvation Attack

Probabilistic method to detect DHCP Starvation Attack - Training Phase

Algorithm 1: Training

Data: ΔT - Window period in hours

d - Training period in days

Result: PD - Probability Distribution

$W \leftarrow \frac{24}{\Delta T}$

Create an array $COUNT[1, \ldots, W]$

for $day = 1$ **to** d **do**

  for $window = 1$ **to** W **do**

    $EventCount \leftarrow 0$

    for $t = t_{start}$ **to** $t_{start} + \Delta T$ **do**

      $EventCount \leftarrow EventCount + 1$

    $COUNT[window] \leftarrow COUNT[window] + EventCount$

Probabilistic method to detect DHCP Starvation Attack - Training Phase

Algorithm 2: ProbEstimate

$Sum \leftarrow 0$
for $i = 1$ **to** W **do**
  $Sum \leftarrow Sum + COUNT[i]$
for $i = 1$ **to** W **do**
  $PD_i \leftarrow \frac{COUNT[i]}{Sum}$
return PD

Probabilistic method to detect DHCP Starvation Attack - Testing Phase

Algorithm 3: Testing

Data: ΔT - Window period in hours

Sum - Total number of occurrences of type *Event* during training phase

PD - Probability Distribution of type *Event*

d - Training Period in Days

Result: Starvation Attack Detection

while *Not interrupted* **do**

  $Event\_Count \leftarrow 0$

  for $t = t^{test}_{start}$ **to** $t^{test}_{start} + \Delta T$ **do**

    $Event \leftarrow$ New Event of Type *Event* Detected

    $Event\_Count \leftarrow Event\_Count + 1$

  $Event\_Count\_train \leftarrow$ **GetCount**$(t^{test}_{start}, t^{test}_{end}, PD, Sum, d)$

  if $Event\_Count \geq Event\_Count\_train + \beta$ **then**

    Starvation Attack detected

Probabilistic method to detect DHCP Starvation Attack - Testing Phase

Algorithm 4: GetCount($t^{test}_{start}, t^{test}_{end}, PD, Sum, d$)

$PD_{test} \leftarrow$ Retrieved probability of type *Event* generated from training phase
$AvgSum \leftarrow \frac{Sum}{d}$
$Event\_Count\_train \leftarrow PD_{test} \times AvgSum$
return $Event\_Count\_train$

Example: Training Phase

- Training period: $d = 3$ days.
- Window period: $\Delta T = 0.5$ hours (48 total time windows).
- **Window 1:** 10:00–10:30

Window	Time	Day 1	Day 2	Day 3
1	10:00–10:30	40	50	30

Example: Training Phase

- Training period: $d = 3$ days.
- Window period: $\Delta T = 0.5$ hours (48 total time windows).
- **Window 1:** 10:00–10:30

Window	Time	Day 1	Day 2	Day 3
1	10:00–10:30	40	50	30

Example: Training Phase

Step 1: Compute Total Count per Window:

- **Window 1 (10:00-10:30):** $40 + 50 + 30 = 120$

Step 2: Compute the Overall Total:

$$\text{Sum} = 480.$$

Step 3: Calculate the Probability Distribution:

$$PD_i = \frac{\text{COUNT}[i]}{\text{Sum}}.$$

Thus, the probability of an event happening in that time frame is:

$$PD_1 = \frac{120}{480} = 0.25,$$

Example: Training Phase

Step 1: Compute Total Count per Window:

- **Window 1 (10:00-10:30):** $40 + 50 + 30 = 120$

Step 2: Compute the Overall Total:

$$\text{Sum} = 480.$$

Step 3: Calculate the Probability Distribution:

$$PD_i = \frac{\text{COUNT}[i]}{\text{Sum}}.$$

Thus, the probability of an event happening in that time frame is:

$$PD_1 = \frac{120}{480} = 0.25,$$

Example: Training Phase

Step 1: Compute Total Count per Window:

- **Window 1 (10:00-10:30):** $40 + 50 + 30 = 120$

Step 2: Compute the Overall Total:

$$\text{Sum} = 480.$$

Step 3: Calculate the Probability Distribution:

$$PD_i = \frac{\text{COUNT}[i]}{\text{Sum}}.$$

Thus, the probability of an event happening in that time frame is:

$$PD_1 = \frac{120}{480} = 0.25,$$

Example: Training Phase

Step 1: Compute Total Count per Window:

- **Window 1 (10:00-10:30):** $40 + 50 + 30 = 120$

Step 2: Compute the Overall Total:

$$\text{Sum} = 480.$$

Step 3: Calculate the Probability Distribution:

$$PD_i = \frac{\text{COUNT}[i]}{\text{Sum}}.$$

Thus, the probability of an event happening in that time frame is:

$$PD_1 = \frac{120}{480} = 0.25,$$

Example: Testing Phase

Step 1: Compute the Expected Event Count:

$$Event\_Count\_train = PD_{test} \times AvgSum(\frac{480}{3}) = 0.25 \times 160 = 40.$$

Step 2: Testing Observation:

- Let the threshold $\beta = 20$.
- Suppose during the testing window (10:00–10:30) we observe $Event\_Count = 500$ events.

Step 3: **Decision Criterion:** The attack detection condition is:

$$\text{Attack detected if } Event\_Count \geq Event\_Count\_train + \beta.$$

$$500 \geq 40 + 20.$$

Example: Testing Phase

Step 1: Compute the Expected Event Count:

$$Event\_Count\_train = PD_{test} \times AvgSum(\frac{480}{3}) = 0.25 \times 160 = 40.$$

Step 2: Testing Observation:

- Let the threshold $\beta = 20$.
- Suppose during the testing window (10:00–10:30) we observe $Event\_Count = 500$ events.

Step 3: Decision Criterion: The attack detection condition is:

$$\text{Attack detected if } Event\_Count \geq Event\_Count\_train + \beta.$$

$$500 \geq 40 + 20.$$

Example: Testing Phase

Step 1: Compute the Expected Event Count:

$$Event\_Count\_train = PD_{test} \times AvgSum(\frac{480}{3}) = 0.25 \times 160 = 40.$$

Step 2: Testing Observation:

- Let the threshold $\beta = 20$.
- Suppose during the testing window (10:00–10:30) we observe $Event\_Count = 500$ events.

Step 3: Decision Criterion: The attack detection condition is:

$$\text{Attack detected if } Event\_Count \geq Event\_Count\_train + \beta.$$

$$500 \geq 40 + 20.$$