Where The Wild Things Are: Brute-Force SSH Attacks In The Wild And How To Stop Them

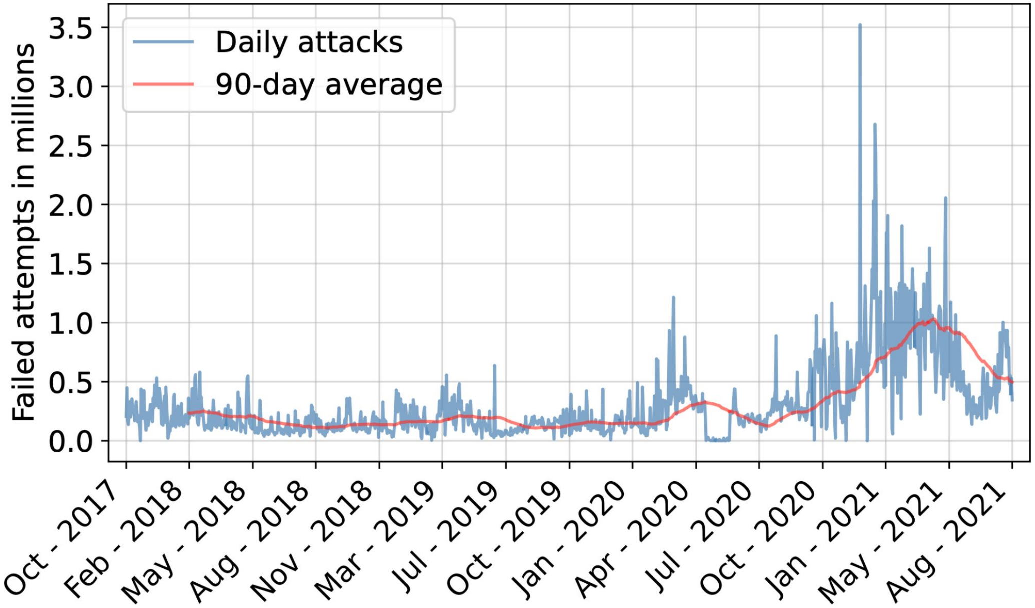
Sachin Kumar Singh, Shreeman Gautam, Cameron Cartier, Sameer Patil, Robert Ricci

University of Utah

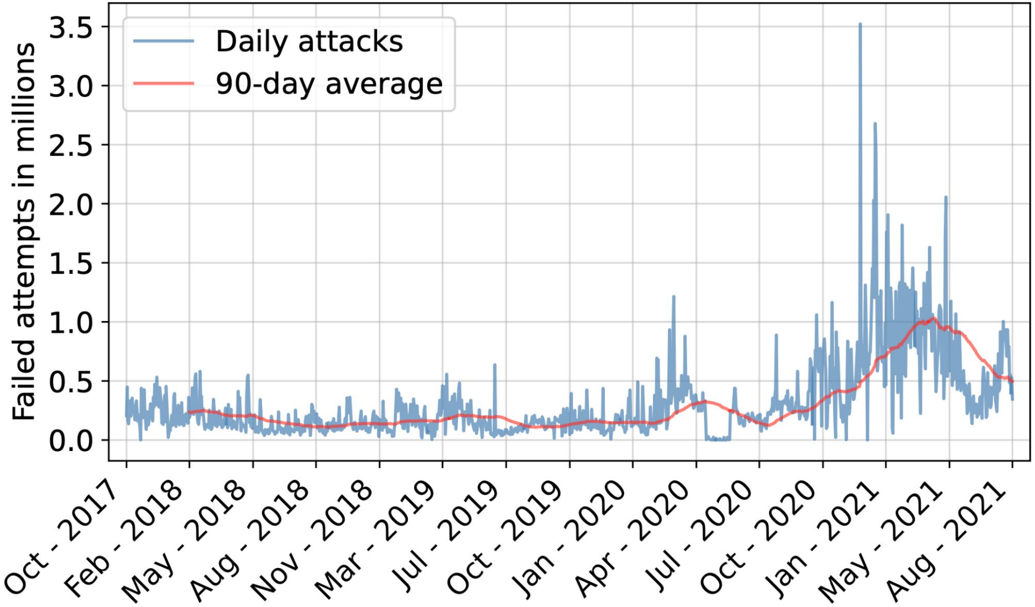
KAHLERT SCHOOL OF COMPUTING
THE UNIVERSITY OF UTAH

SSH Brute Force Attacks (BFAs) in the Wild

SSH Brute Force Attacks (BFAs) in the Wild

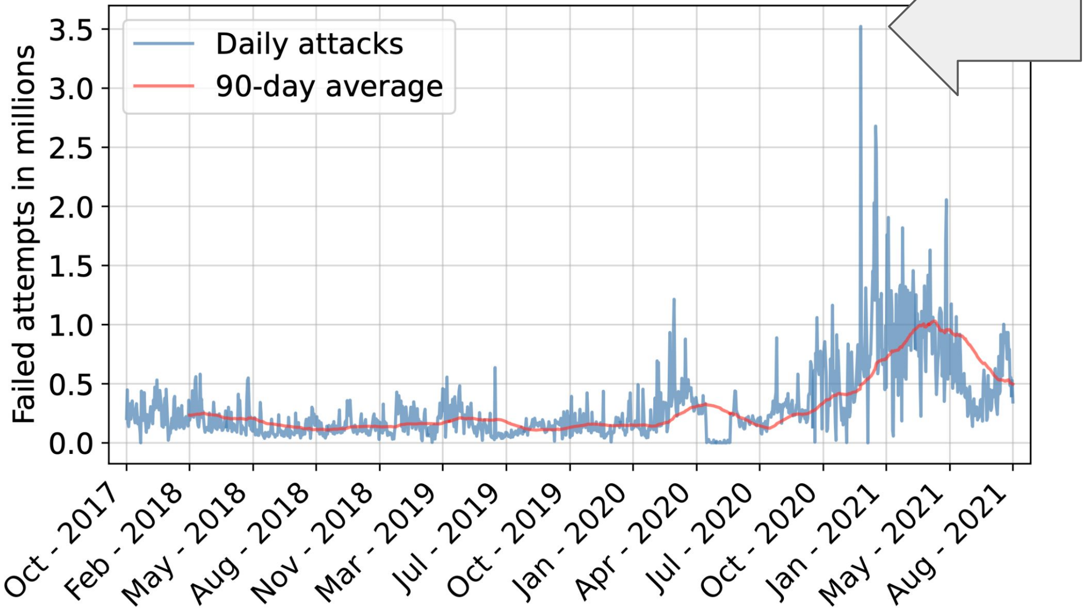


SSH Brute Force Attacks (BFAs) in the Wild



381 million failed brute force attempts

SSH Brute Force Attacks (BFAs) in the Wild

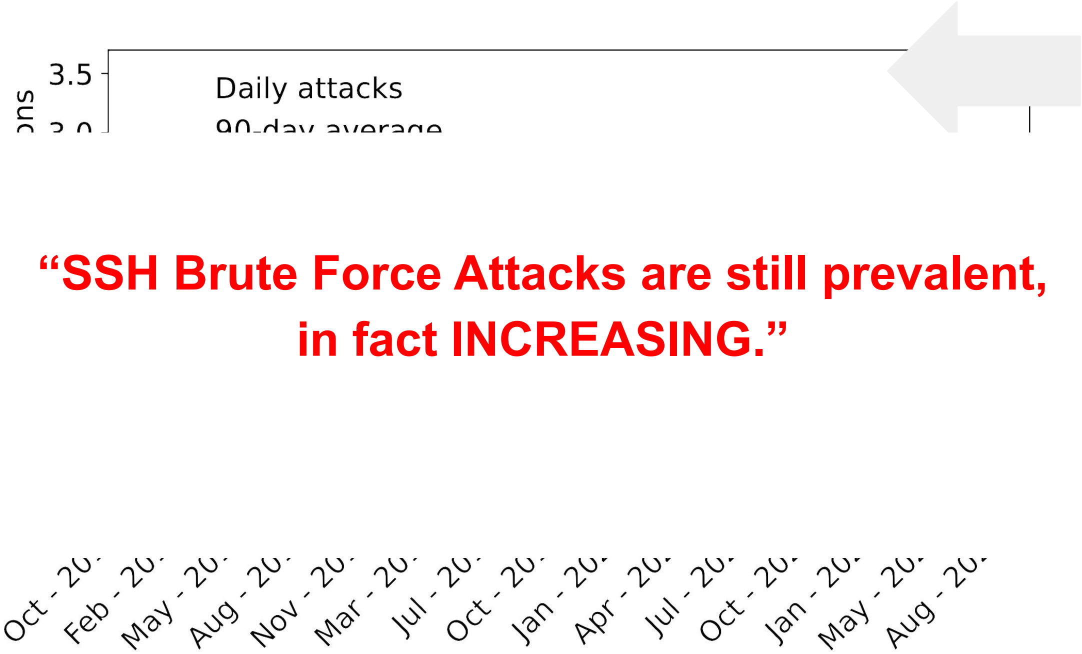


Peak 3.5 million a day

381 million failed brute force attempts

SSH Brute Force Attacks (BFAs) in the Wild

Peak 3.5 million a day



"SSH Brute Force Attacks are still prevalent, in fact INCREASING."

381 million failed brute force attempts

Data Collection

Data Collection

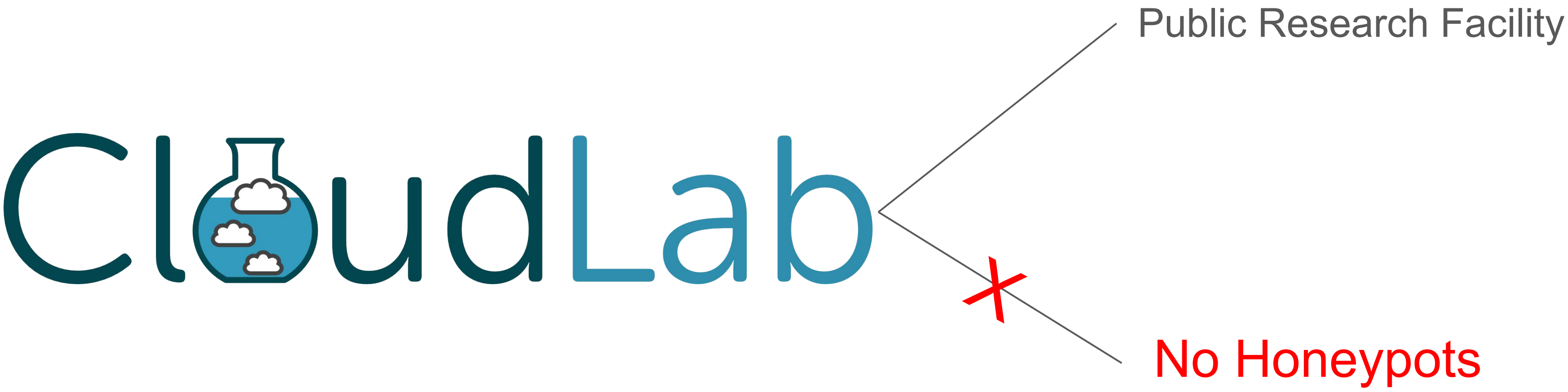
CloudLab

Data Collection

CloudLab

Public Research Facility

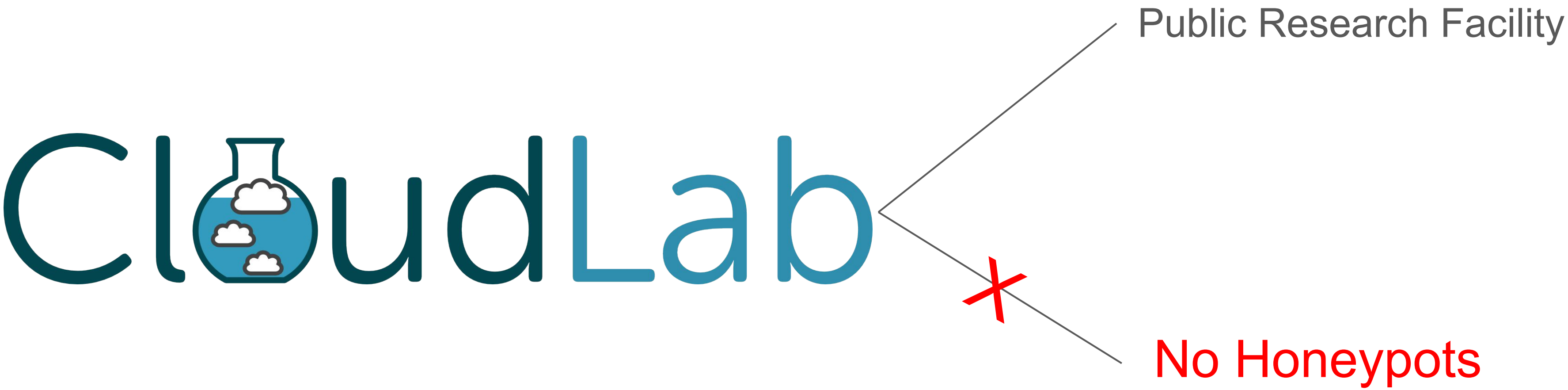
Data Collection



Public Research Facility

No Honeypots

Data Collection



Legitimate Users & Attackers

Data Collection

Public Research Facility

“Our unique data aided the development of blocking.”

Legitimate Users & Attackers

Data Collection

Public Research Facility

“Our unique data aided the development of blocking.”

“Provide the means to evaluate effectiveness”

Legitimate Users & Attackers

Ingredients for a SSH Brute Force Attack

Ingredients for a SSH Brute Force Attack

- Target Machine (CloudLab Nodes)

Ingredients for a SSH Brute Force Attack

- Target Machine (CloudLab Nodes)
  - ~500 Nodes

Ingredients for a SSH Brute Force Attack

- Target Machine (CloudLab Nodes)
  - ~500 Nodes
  - Attacker pick nodes **RANDOMLY**

Ingredients for a SSH Brute Force Attack

- Target Machine (CloudLab Nodes)
  - ~500 Nodes
  - Attacker pick nodes **RANDOMLY**
- Host Machine (Source IP)

Ingredients for a SSH Brute Force Attack

- Target Machine (CloudLab Nodes)
  - ~500 Nodes
  - Attacker pick nodes **RANDOMLY**
- Host Machine (Source IP)
  - ~800K IPs

Ingredients for a SSH Brute Force Attack

- Target Machine (CloudLab Nodes)
  - ~500 Nodes
  - Attacker pick nodes RANDOMLY
- Host Machine (Source IP)
  - ~800K IPs
- Guessing Vector (*{username,password}* pairs)

Ingredients for a SSH Brute Force Attack

- Target Machine (CloudLab Nodes)
  - ~500 Nodes
  - Attacker pick nodes RANDOMLY
- Host Machine (Source IP)
  - ~800K IPs
- Guessing Vector (*{username,password}* pairs)
  - ~277K unique usernames

Ingredients for a SSH Brute Force Attack

- Target Machine (CloudLab Nodes)
    - ~500 Nodes
    - Attacker pick nodes RANDOMLY
- Host Machine (Source IP)
    - ~800K IPs
- Guessing Vector (*{**username**,password}* pairs)
    - ~277K unique usernames

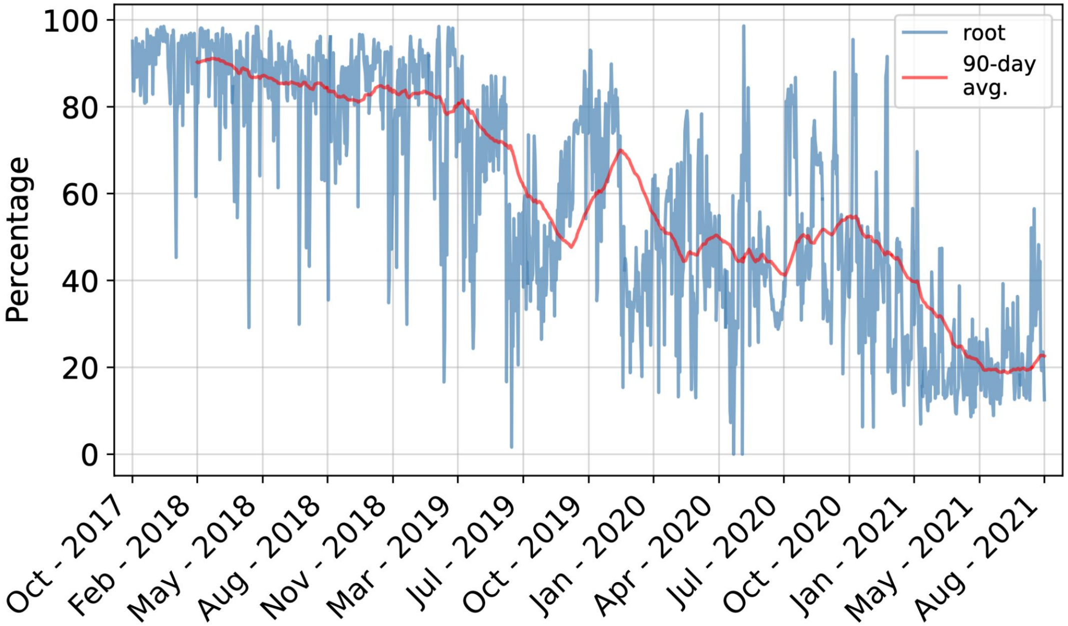
Username in Guessing Vector

Username in Guessing Vector

- *root*

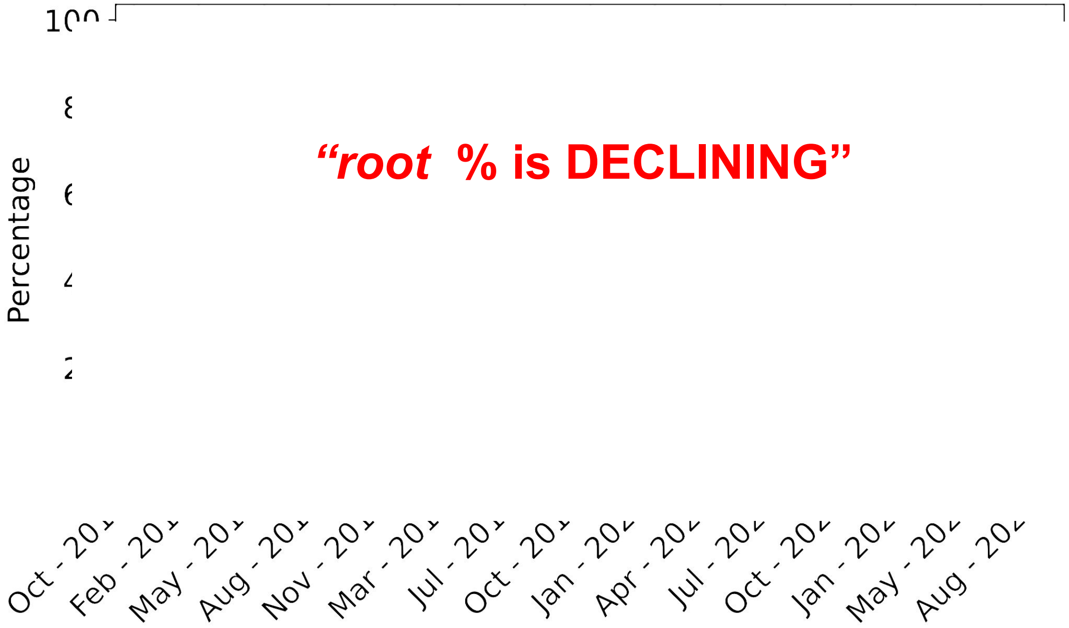
Username in Guessing Vector

- *root*



Username in Guessing Vector

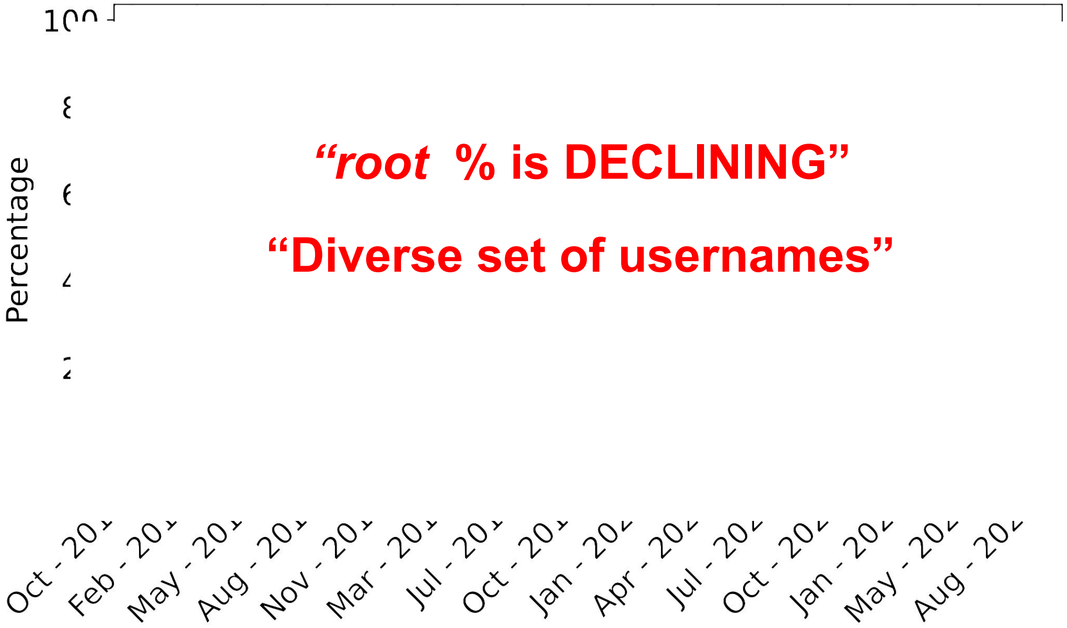
- *root*



“*root* % is DECLINING”

Username in Guessing Vector

- *root*



“*root* % is DECLINING”

“Diverse set of usernames”

- Are there patterns in the usernames utilized by attackers?

- Are there patterns in the usernames utilized by attackers?

- Can these patterns be fingerprinted for effective blocking?

Username Set in Guessing Vector

Attacker →

Username Set **in Guessing Vector**

Attacker → Guessing Vector →

Username Set in Guessing Vector

Attacker → Guessing Vector →({username-1},

{username-2},

{username-3},

.

.

.

.

{username-n})

Username Set in Guessing Vector

Attacker → Guessing Vector →({username-1},

{username-2},

{username-3},

.

.

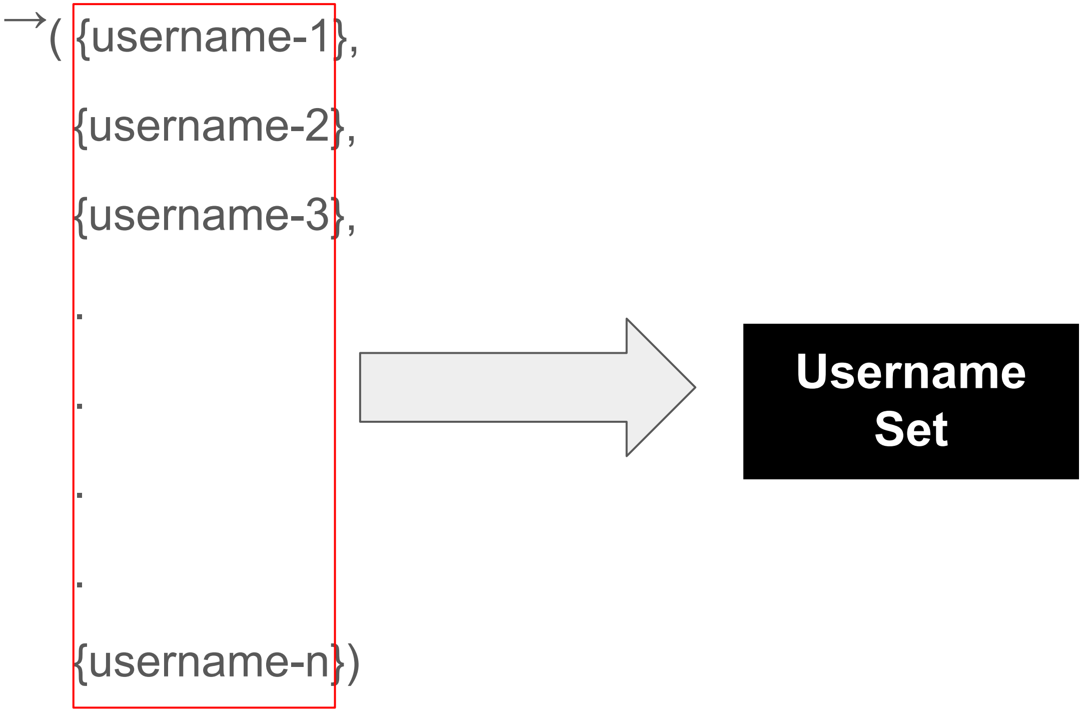
.

.

{username-n})

Username Set in Guessing Vector

Attacker → Guessing Vector →



({username-1},

{username-2},

{username-3},

.

.

.

.

{username-n})

Username Set

Username Set in Guessing Vector

Attacker_1 →

Attacker_2 →

Attacker_3 →

Attacker_4 →

.

.

.

Attacker_n →

Username Set in Guessing Vector

Attacker_1 → **Username Set A**

Attacker_2 →

Attacker_3 →

Attacker_4 → **Username Set A**

.

.

.

Attacker_n →

Username Set in Guessing Vector

Attacker_1 → Username Set A

Attacker_2 → **Username Set B**

Attacker_3 →

Attacker_4 → Username Set A

.

.

.

Attacker_n →

Username Set in Guessing Vector

Attacker_1 → Username Set A

Attacker_2 → Username Set B

Attacker_3 → **Username Set C**

Attacker_4 → Username Set A

.

.

.

Attacker_n → **Username Set C**

Username Set in Guessing Vector

Attacker_1 → Username Set A

Attacker_2 → Username Set B

Attacker_3 → Username Set C

Attacker_4 → Username Set A

.

.

.

Attacker_n → Username Set C

Username Set A

Username Set in Guessing Vector

Attacker_1 → Username Set A

Attacker_2 → Username Set B

Attacker_3 → Username Set C

Attacker_4 → Username Set A

.

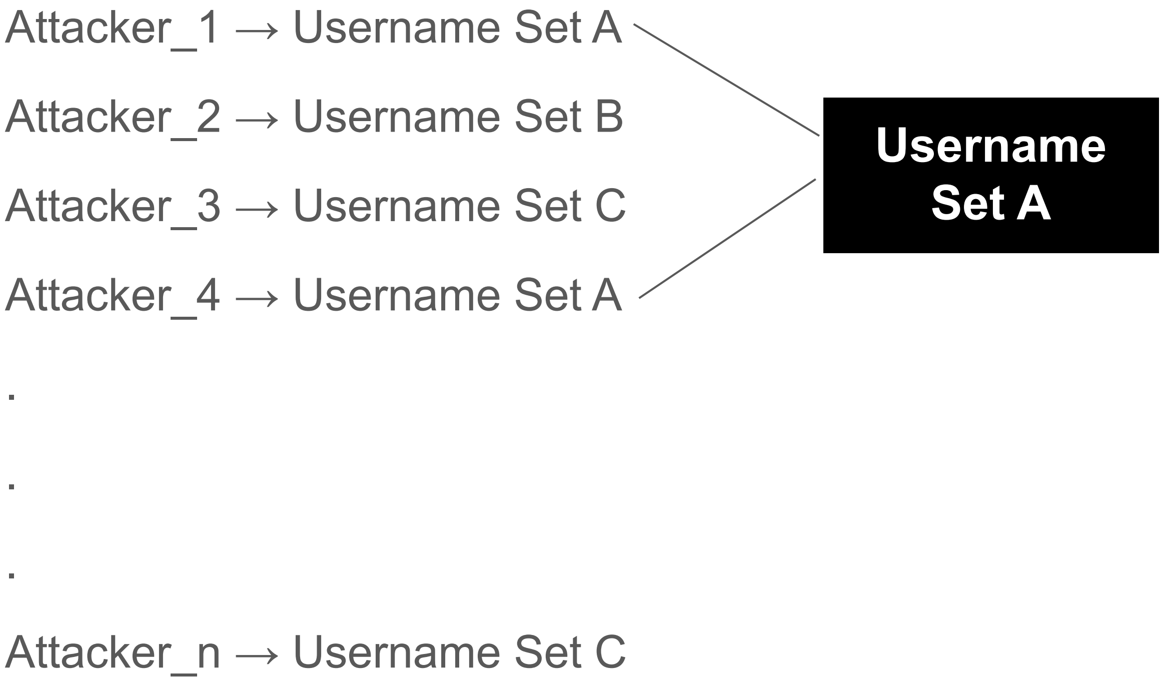
.

.

Attacker_n → Username Set C

Username Set A

Username Set in Guessing Vector



Attacker_1 → Username Set A

Attacker_2 → Username Set B

Attacker_3 → Username Set C

Attacker_4 → Username Set A

.

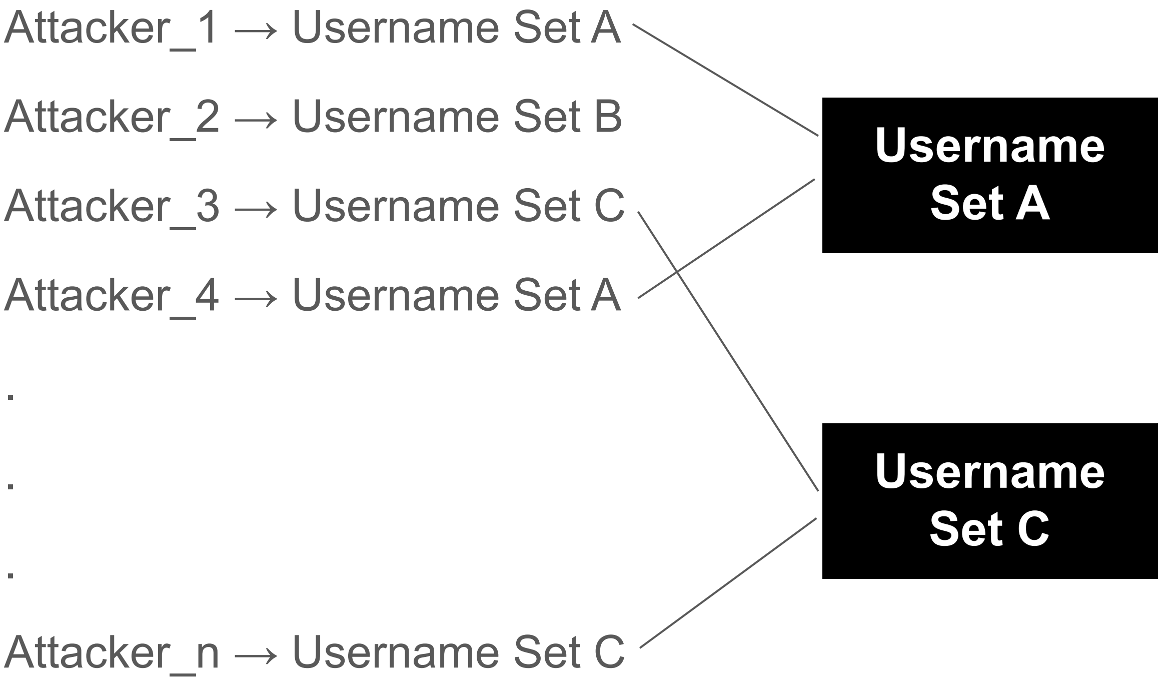
.

.

Attacker_n → Username Set C

Username Set A

Username Set in Guessing Vector



Attacker_1 → Username Set A

Attacker_2 → Username Set B

Attacker_3 → Username Set C

Attacker_4 → Username Set A

.

.

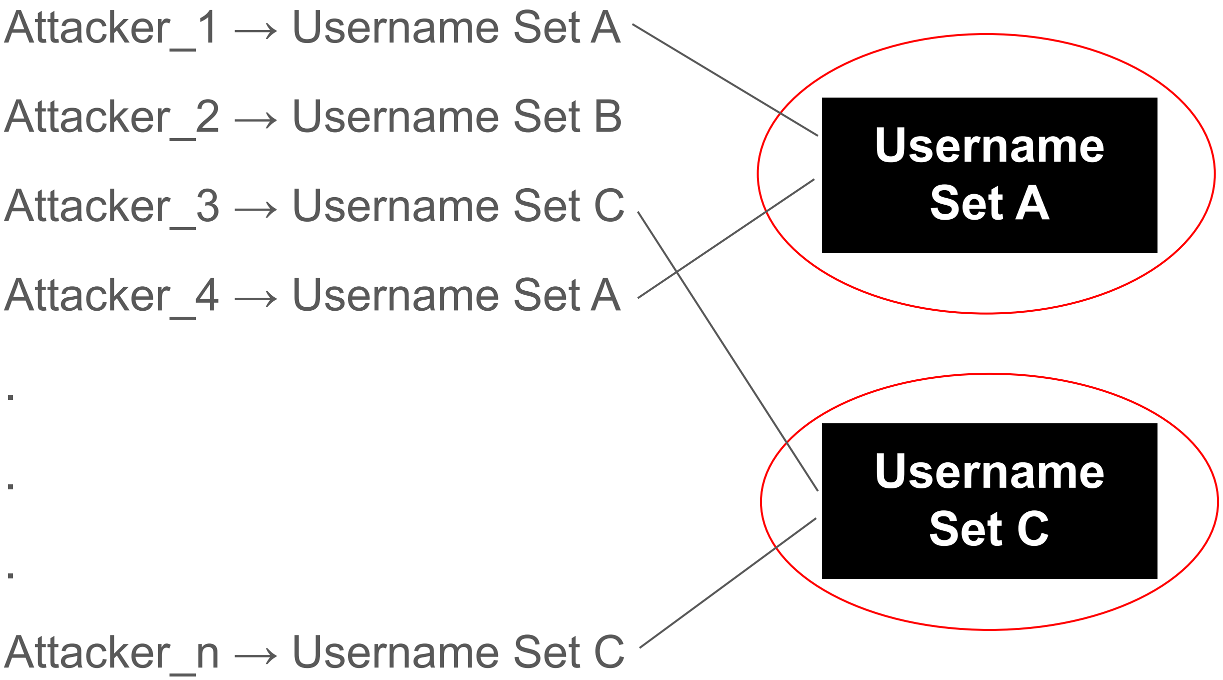
.

Attacker_n → Username Set C

Username Set A

Username Set C

Username Set in Guessing Vector



Attacker_1 → Username Set A

Attacker_2 → Username Set B

Attacker_3 → Username Set C

Attacker_4 → Username Set A

.

.

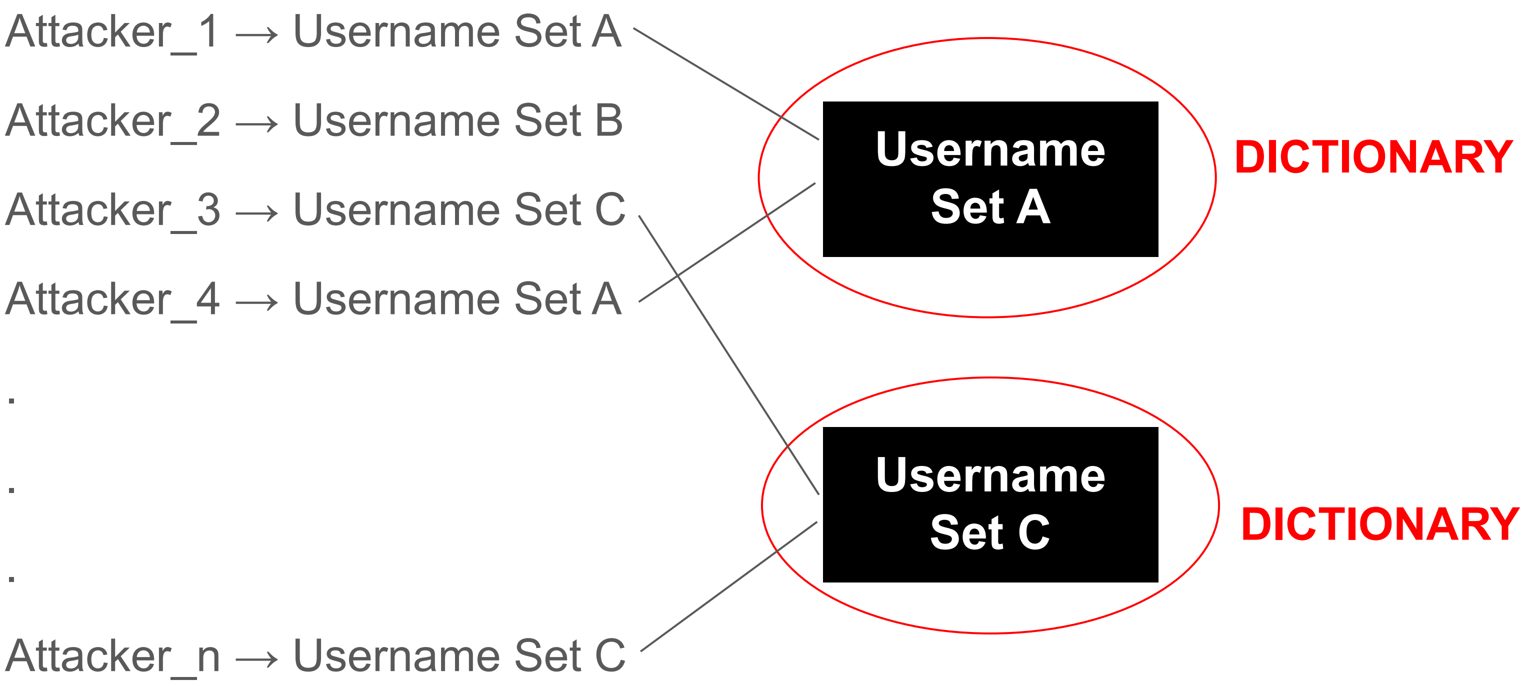
.

Attacker_n → Username Set C

Username Set A

Username Set C

Username Set in Guessing Vector



Attacker_1 → Username Set A

Attacker_2 → Username Set B

Attacker_3 → Username Set C

Attacker_4 → Username Set A

.

.

.

Attacker_n → Username Set C

Username Set A

DICTIONARY

Username Set C

DICTIONARY

Username Set in Guessing Vector

Attacker_1 → Username Set A

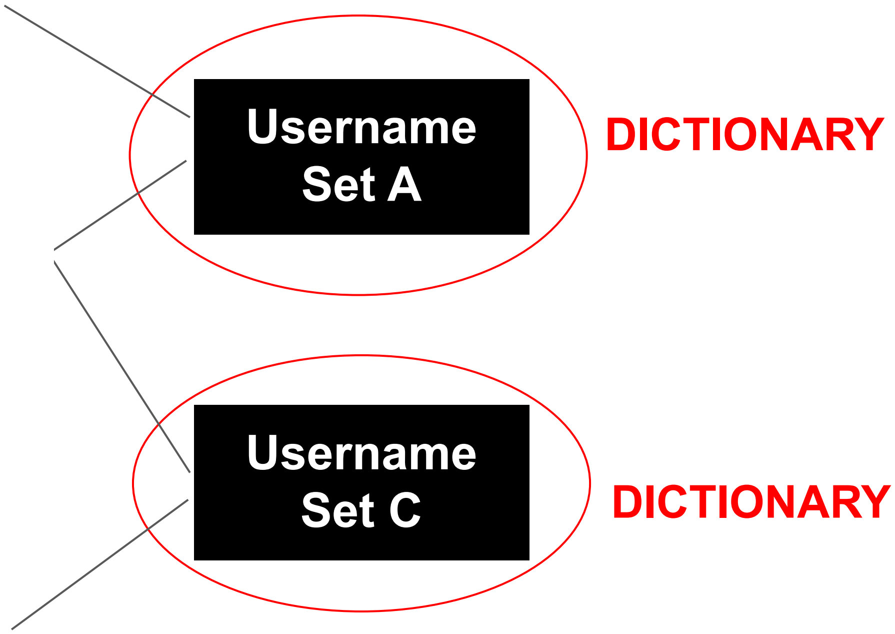
Attacker_2 → Username Set B

- **64% attackers use dictionary**

.

.

Attacker_n → Username Set C



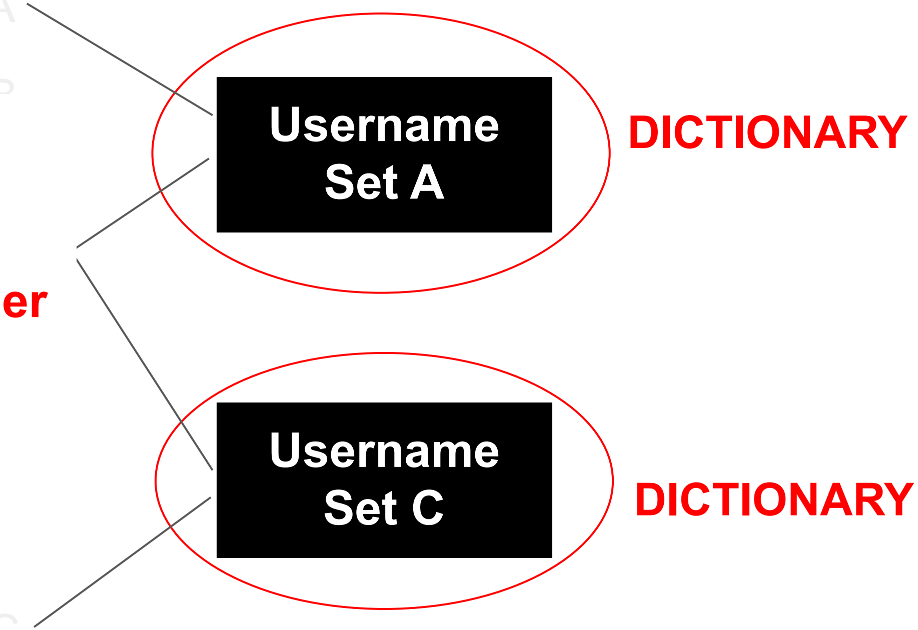
Username Set in Guessing Vector

Attacker_1 → Username Set A

Attacker_2 → Username Set B

- **64% attackers use dictionary**
- **94% of the attackers user at least one username from a dictionary**

Attacker_n → Username Set C



Username Set A

DICTIONARY

Username Set C

DICTIONARY

Dictionary Based Blocking (DBB)

Dictionary Based Blocking (DBB)

- We create a Username Blocking List (UBL) by combining all dictionaries.

Dictionary Based Blocking (DBB)

- We create a Username Blocking List (UBL) by combining all dictionaries.
- We perform local sanitation to eliminate usernames from the Username Blocking List that are locally valid.

Dictionary Based Blocking (DBB)

- We create a Username Blocking List (UBL) by combining all dictionaries.
- We perform local sanitation to eliminate usernames from the Username Blocking List that are locally valid.
- Any IP that attempts a failed login with a username present in the Username Blocking List is subsequently blocked.

Dictionary Based Blocking (DBB)

- We create a Username Blocking List (UBL) by combining all dictionaries.
- We perform local sanitation to eliminate usernames from the Username Blocking List that are locally valid.
- Any IP that attempts a failed login with a username present in the Username Blocking List is subsequently blocked.

- **64% attackers use dictionary**
- **94% of the attackers user at least one username from a dictionary**

- How DBB performs in % Attacks Blocked and False Positives?

- How DBB performs in % Attacks Blocked and False Positives?

- Does the characteristics of Dictionary Based Blocking generalize?

Evaluating Dictionary Based Blocking (DBB)

Evaluating Dictionary Based Blocking (DBB)

- We simulated DBB on three different sites data (A,B,C) over ten weeks.

Evaluating Dictionary Based Blocking (DBB)

- We simulated DBB on three different sites data (A,B,C) over ten weeks.
- DBB effectively blocked over 99.3% of BFAs across all sites with only ~14 false positives per site.

Performance of DBB across multiple sites

Performance of DBB across multiple sites

- For three sites (A,B,C), we checked whether Username Blocking List (UBL) created at one site are effective at other sites.

Performance of DBB across multiple sites

- For three sites (A,B,C), we checked whether Username Blocking List (UBL) created at one site are effective at other sites.

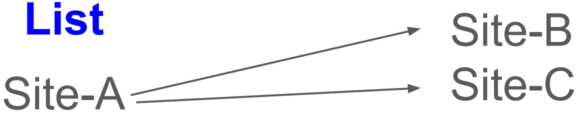
Username Blocking List

Site-A

Performance of DBB across multiple sites

- For three sites (A,B,C), we checked whether Username Blocking List (UBL) created at one site are effective at other sites.

Username Blocking List



Site-A

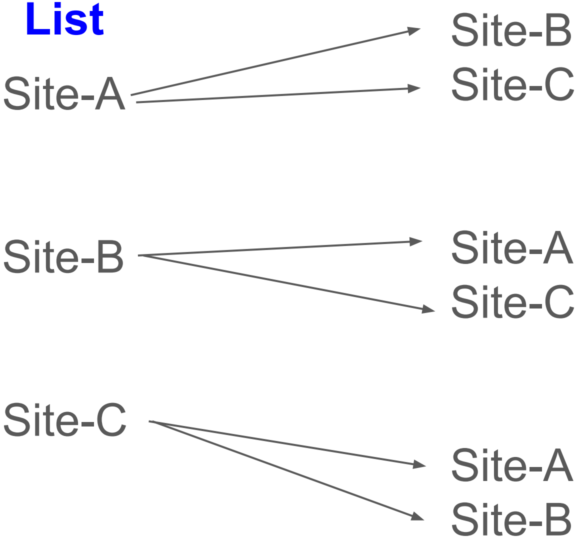
Site-B

Site-C

Performance of DBB across multiple sites

- For three sites (A,B,C), we checked whether Username Blocking List (UBL) created at one site are effective at other sites.

Username Blocking List



Site-A → Site-B, Site-C

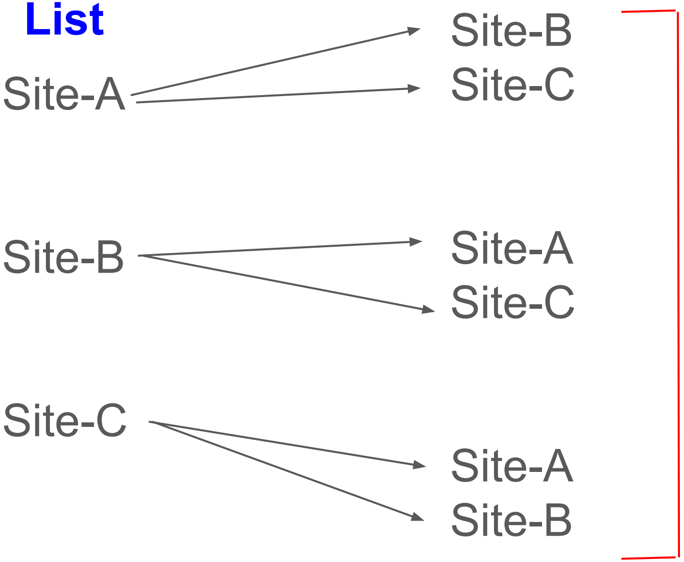
Site-B → Site-A, Site-C

Site-C → Site-A, Site-B

Performance of DBB across multiple sites

- For three sites (A,B,C), we checked whether Username Blocking List (UBL) created at one site are effective at other sites.

Username Blocking List



Site-A → Site-B, Site-C

Site-B → Site-A, Site-C

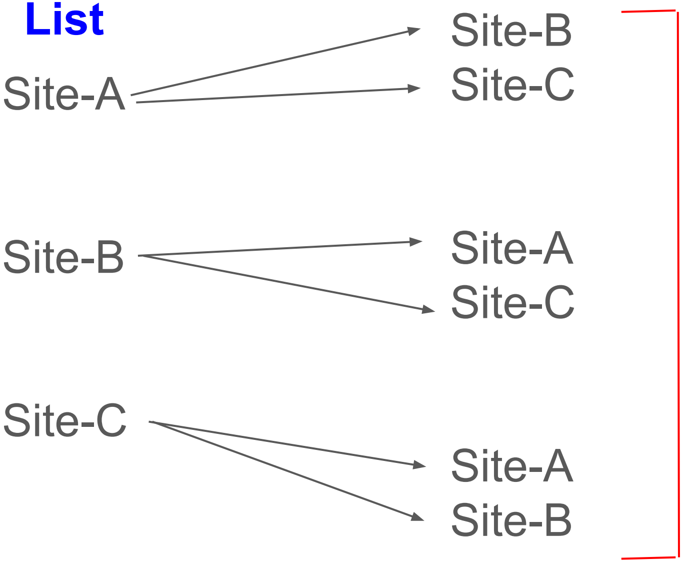
Site-C → Site-A, Site-B

Blocked 99.4% of Attacks

Performance of DBB across multiple sites

- For three sites (A,B,C), we checked whether Username Blocking List (UBL) created at one site are effective at other sites.

Username Blocking List



Site-A → Site-B, Site-C

Site-B → Site-A, Site-C

Site-C → Site-A, Site-B

Blocked 99.4% of Attacks

13 False Positives

Dictionary Based Blocking (DBB)

“Dictionary Based Blocking (DBB) does generalize”

Dictionary Based Blocking (DBB)

“Dictionary Based Blocking (DBB) does generalize”

“High Blocking Rate with Low False Positives”

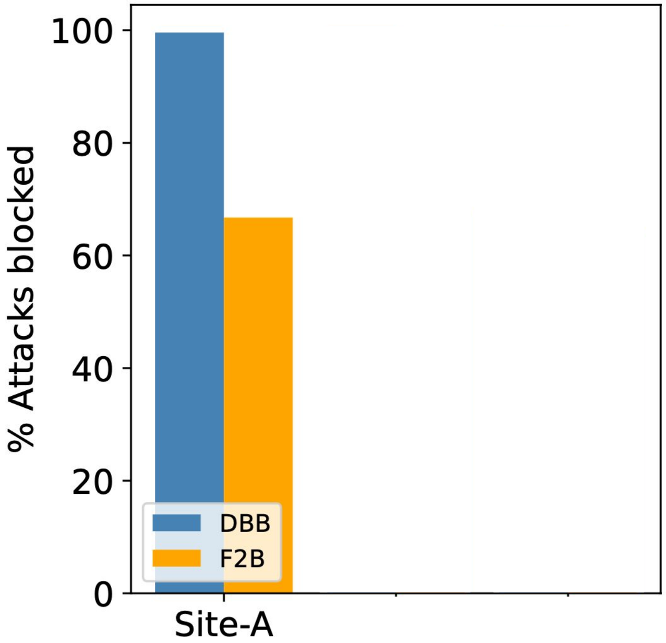
Evaluating DBB: DBB and Fail2ban

Evaluating DBB: DBB and Fail2ban

- Default settings for DBB and Fail2ban.

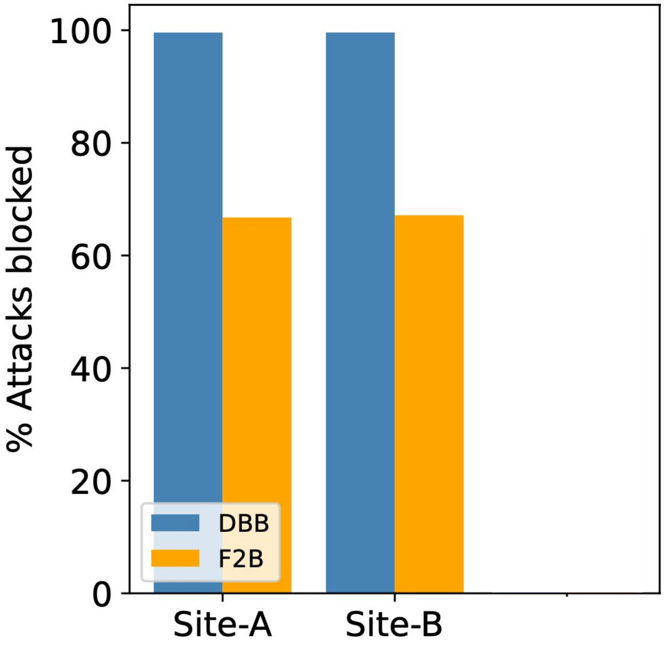
Evaluating DBB: DBB and Fail2ban

- Default settings for DBB and Fail2ban.



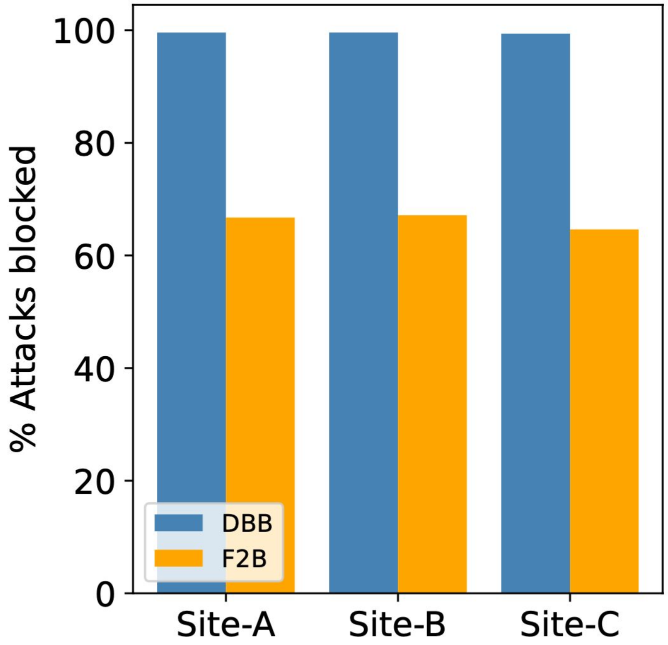
Evaluating DBB: DBB and Fail2ban

- Default settings for DBB and Fail2ban.



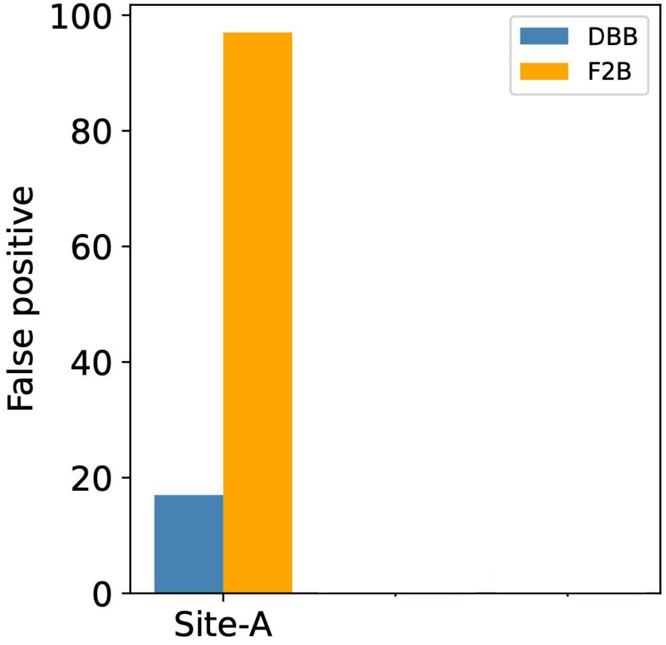
Evaluating DBB: DBB and Fail2ban

- Default settings for DBB and Fail2ban.



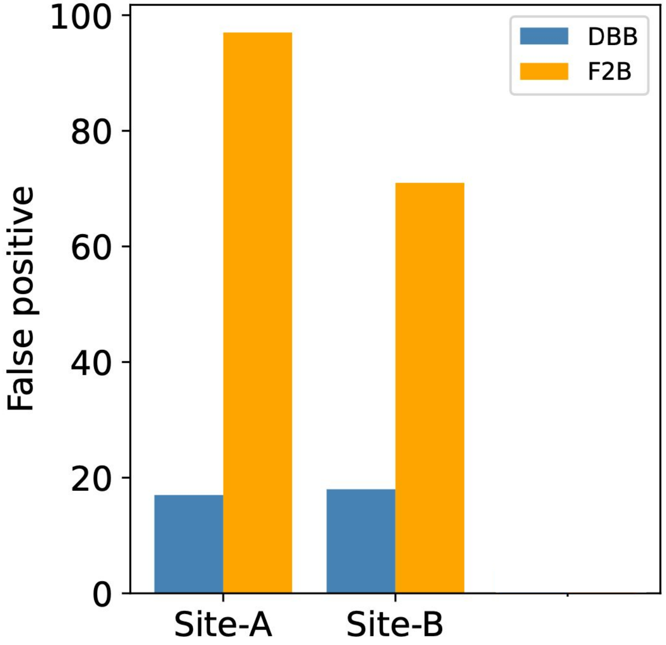
Evaluating DBB: DBB and Fail2ban

- Default settings for DBB and Fail2ban.



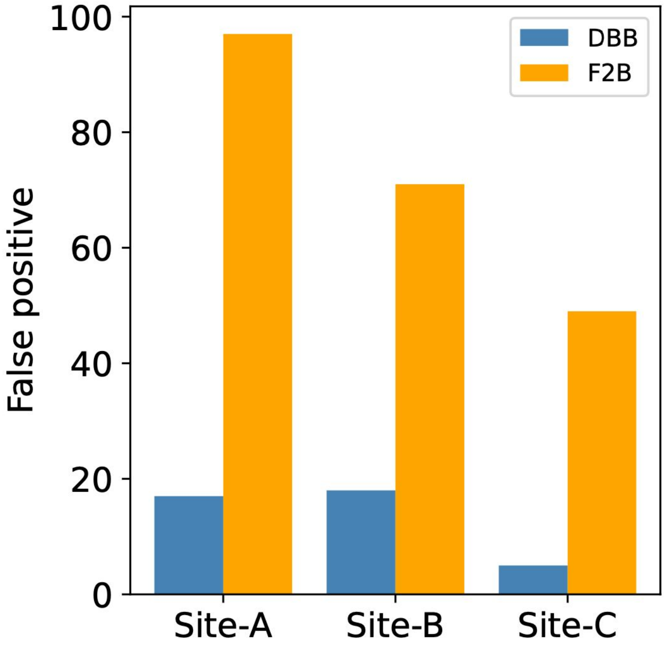
Evaluating DBB: DBB and Fail2ban

- Default settings for DBB and Fail2ban.



Evaluating DBB: DBB and Fail2ban

- Default settings for DBB and Fail2ban.



Evaluating DBB: DBB and Fail2ban

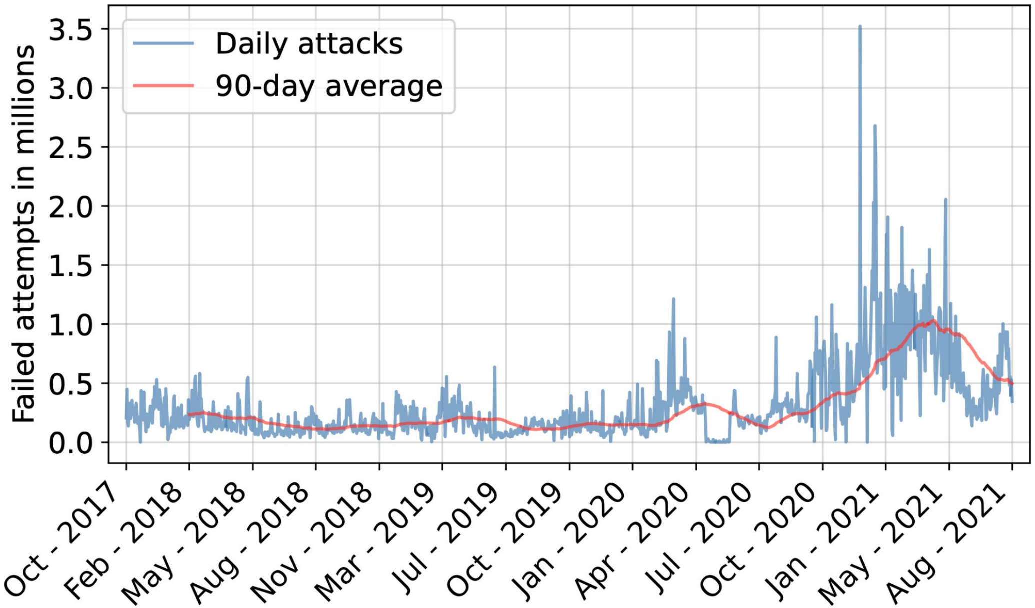
- Default settings for DBB and Fail2ban.

“Dictionary Based Blocking outperforms Fail2ban with huge margin”

Dictionary Based Blocking In Production

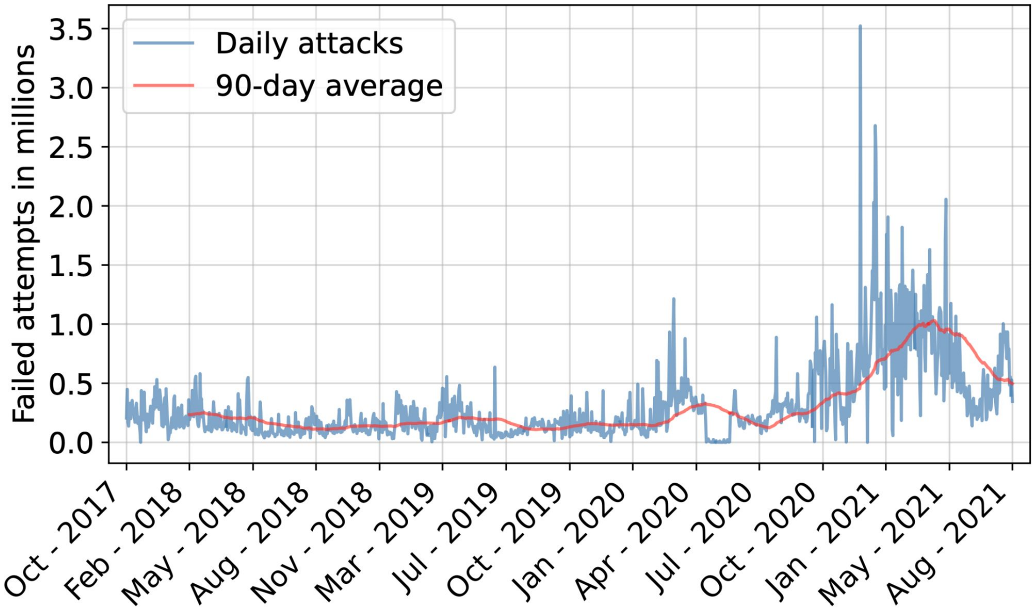
Dictionary Based Blocking In Production

Revisiting SSH Brute Force Attacks in the Wild



Dictionary Based Blocking In Production

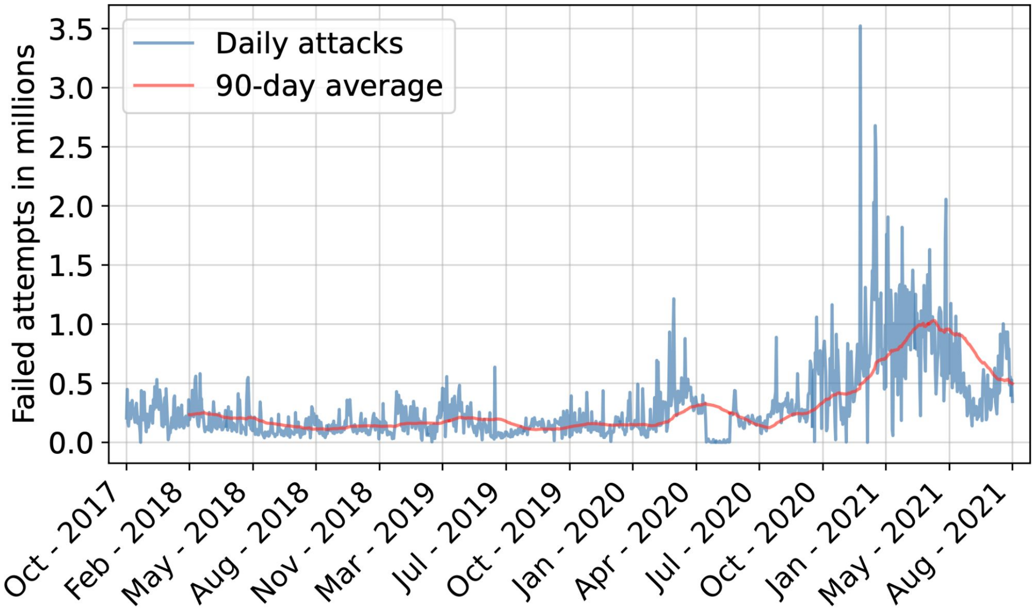
Revisiting SSH Brute Force Attacks in the Wild



→ **After Aug 2021?**

Dictionary Based Blocking In Production

Revisiting SSH Brute Force Attacks in the Wild



→ After Aug 2021?

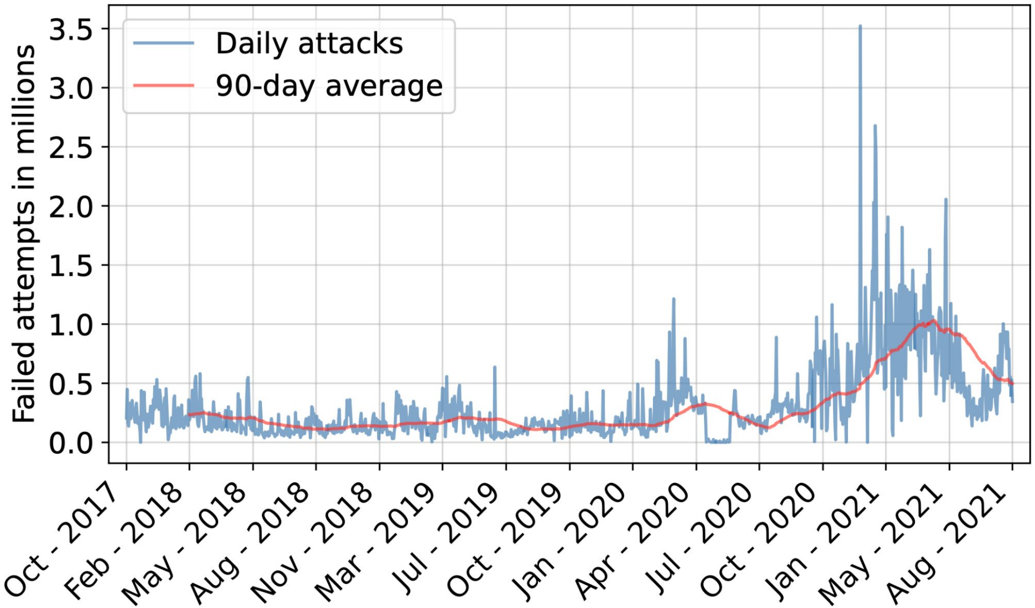
FIREWALL

Periodic updation of filter rules

Aug 2021

Dictionary Based Blocking In Production

Revisiting SSH Brute Force Attacks in the Wild



→ **After Aug 2021?**

FIREWALL

Periodic updation of filter rules

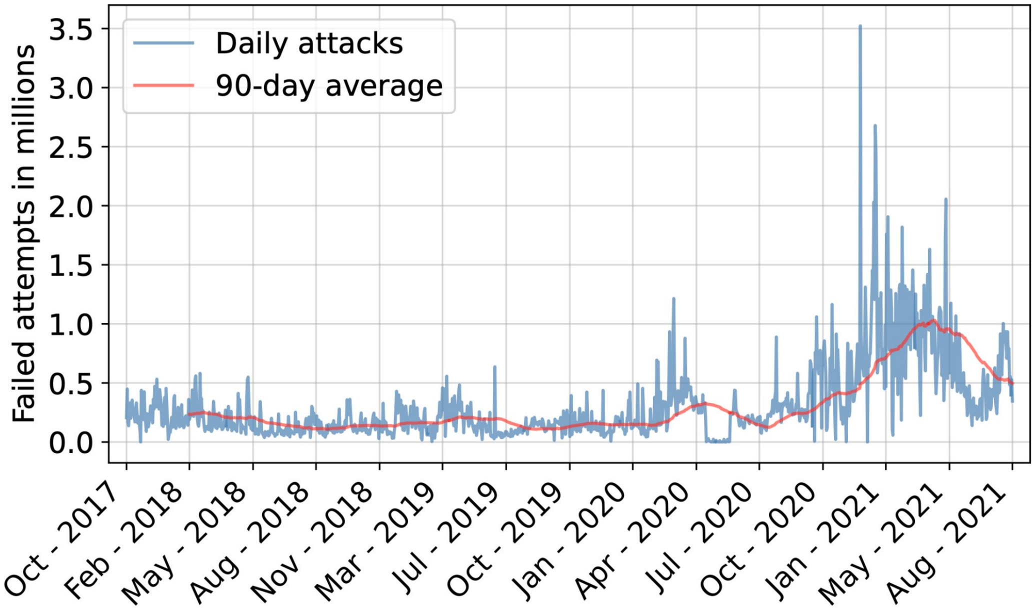
Unfiltered

Filtered

Aug 2021

Dictionary Based Blocking In Production

Revisiting SSH Brute Force Attacks in the Wild



→ After Aug 2021?

FIREWALL

Periodic updation of filter rules

DBB

Unfiltered

Filtered

Aug 2021

Evaluating DBB In Production

- Deployed Dictionary Bases Blocking for three weeks on ~400 nodes.

Evaluating DBB In Production

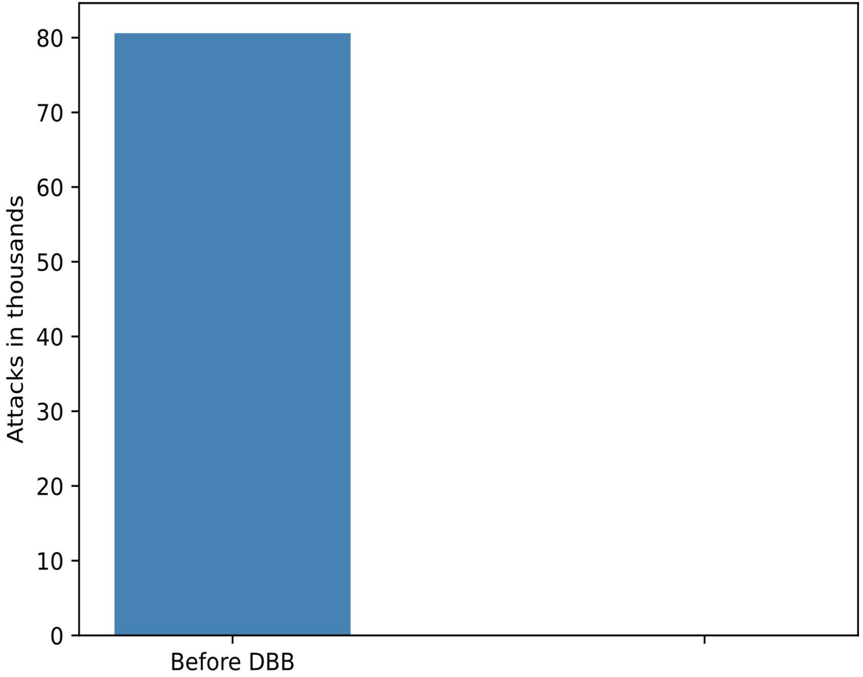
- Deployed Dictionary Bases Blocking for three weeks on ~400 nodes.
- Due to real-time IP blocking and filtered traffic, calculating the exact attack block rate is challenging.

Evaluating DBB In Production

- Deployed Dictionary Bases Blocking for three weeks on ~400 nodes.
- Due to real-time IP blocking and filtered traffic, calculating the exact attack block rate is challenging.
- Evaluate Dictionary Based Blocking effectiveness by comparing attack volumes pre and post-deployment.

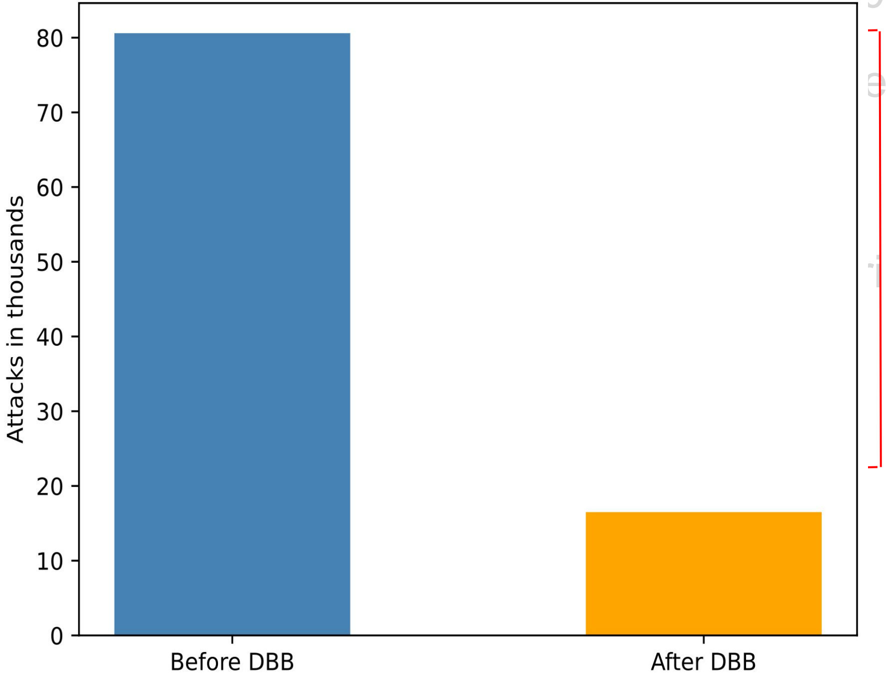
Evaluating DBB In Production

- Deployed Dictionary Bases Blocking for three weeks on ~400 nodes.
- Due to real-time e exact attack block rate is challengin
- Evaluate Dictiona ring attack volumes pre and post-dep



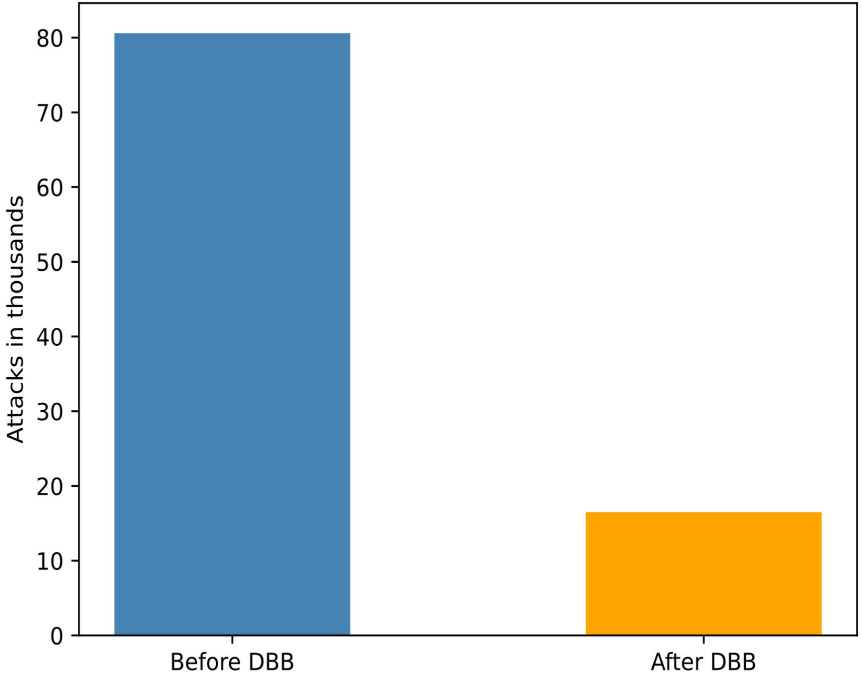
Evaluating DBB In Production

- Deployed Dictionary Bases Blocking for three weeks on ~400 nodes.
- Due to real-time e exact attack block rate is challengin
- Evaluate Dictiona ing attack volumes pre and post-dep



Evaluating DBB In Production

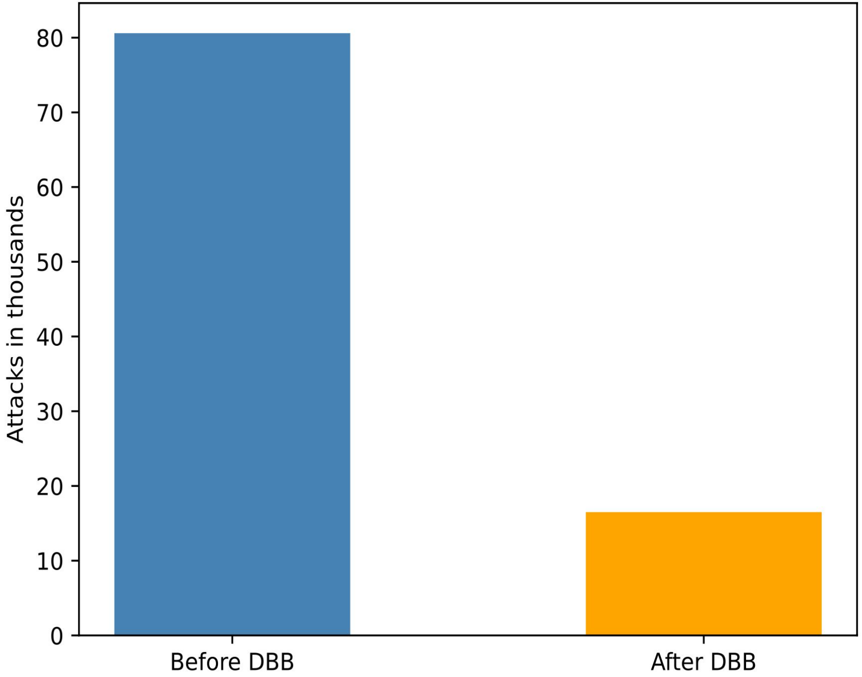
- Deployed Dictionary Bases Blocking for three weeks on ~400 nodes.
- Due to real-time … e exact attack block rate is challengin
- Evaluate Dictiona … ir pre and post-dep



Blocks Four-fifths attacks missed by other defences

Evaluating DBB In Production

- Deployed Dictionary Bases Blocking for three weeks on ~400 nodes.
- Due to real-time e exact attack block rate is challengin
- Evaluate Dictiona ir pre and post-dep



Blocks Four-fifths attacks missed by other defences

DBB had zero FP

- Does the performance comes from the high number of nodes in CloudLab?

- Does the performance comes from the high number of nodes in CloudLab?

Short answer is

- Does the performance comes from the high number of nodes in CloudLab?

Short answer is NO

- Does the performance comes from the high number of nodes in CloudLab?

Short answer is NO

Long answer is

- Does the performance comes from the high number of nodes in CloudLab?

Short answer is NO

Long answer is NO IT DOESN'T

- Does the performance comes from the high number of nodes in CloudLab?

- How many nodes (collectors) are required to perform effective blocking?

Long answer is NO IT DOESN'T

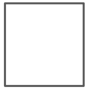
Performance based on number of Collectors

- To examine the effect of the number of collectors on blocking performance, we computed Username Blocking List from various number of collectors.

Performance based on number of Collectors

- To examine the effect of the number of collectors on blocking performance, we computed Username Blocking List from various number of collectors.

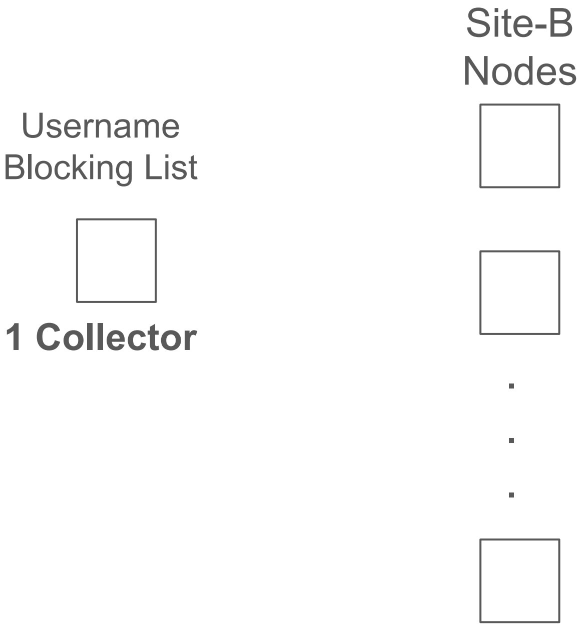
Username Blocking List



1 Collector

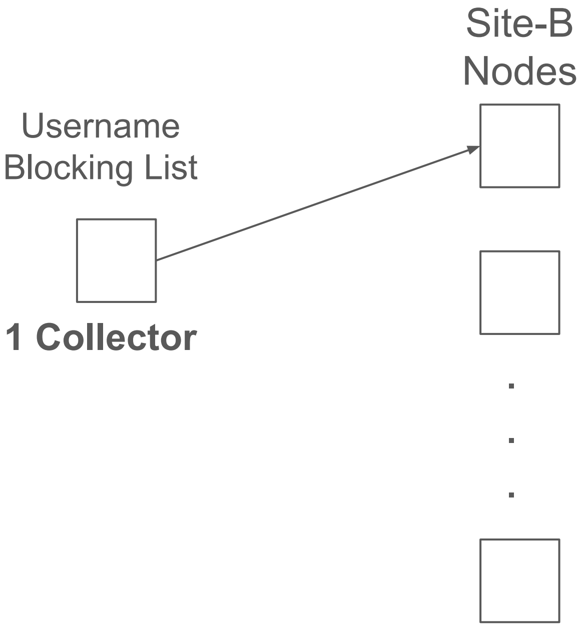
Performance based on number of Collectors

- To examine the effect of the number of collectors on blocking performance, we computed Username Blocking List from various number of collectors.



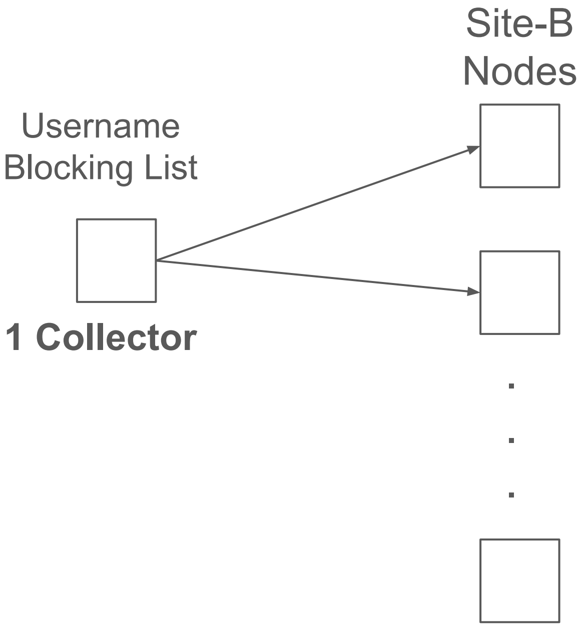
Performance based on number of Collectors

- To examine the effect of the number of collectors on blocking performance, we computed Username Blocking List from various number of collectors.



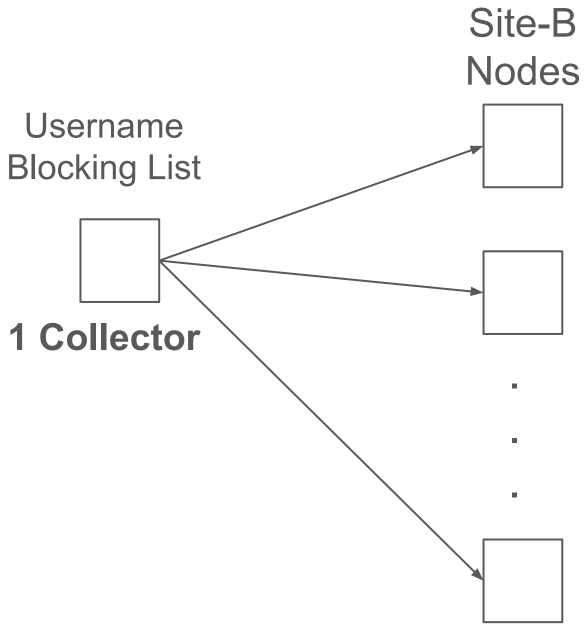
Performance based on number of Collectors

- To examine the effect of the number of collectors on blocking performance, we computed Username Blocking List from various number of collectors.



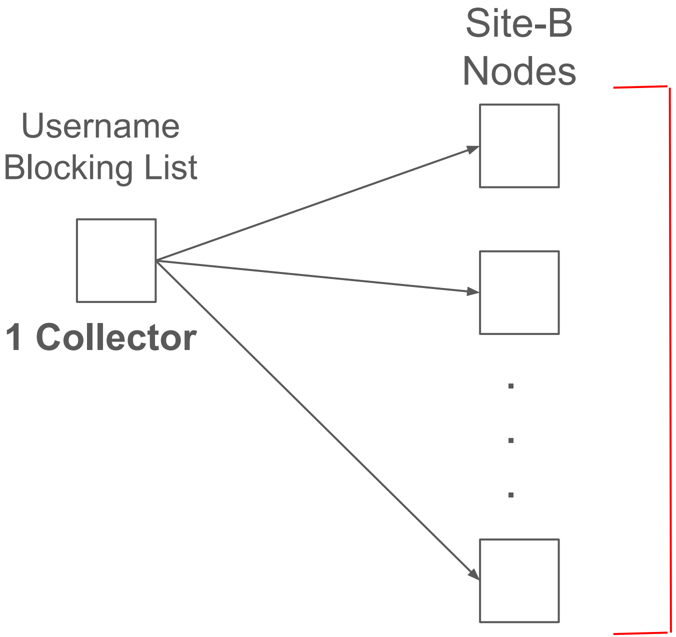
Performance based on number of Collectors

- To examine the effect of the number of collectors on blocking performance, we computed Username Blocking List from various number of collectors.



Performance based on number of Collectors

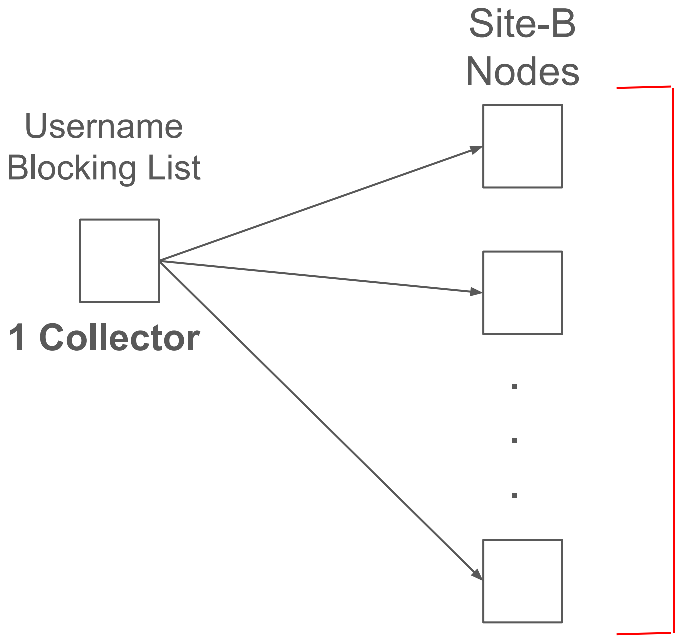
- To examine the effect of the number of collectors on blocking performance, we computed Username Blocking List from various number of collectors.



1 Collector - Blocked Minimum 97.6% attacks

Performance based on number of Collectors

- To examine the effect of the number of collectors on blocking performance, we computed Username Blocking List from various number of collectors.

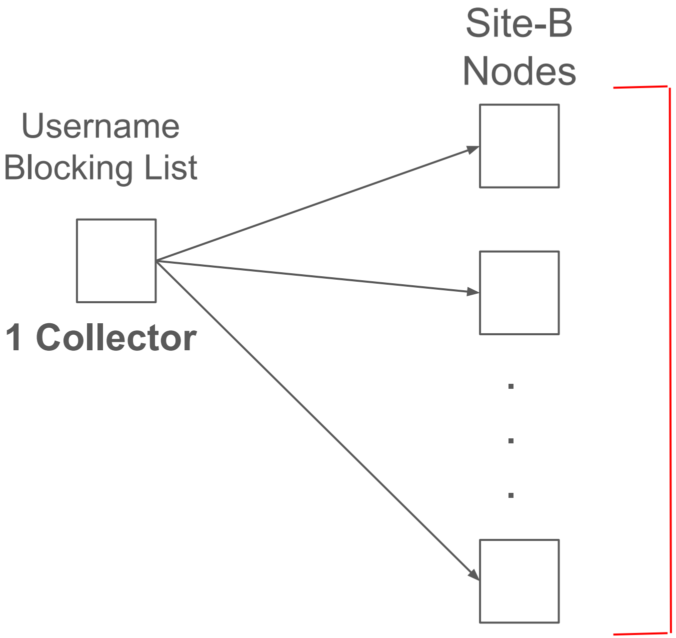


1 Collector - Blocked Minimum 97.6% attacks

2 Collector - Blocked Minimum 98.4% attacks

Performance based on number of Collectors

- To examine the effect of the number of collectors on blocking performance, we computed Username Blocking List from various number of collectors.



1 Collector - Blocked Minimum 97.6% attacks

2 Collector - Blocked Minimum 98.4% attacks

.

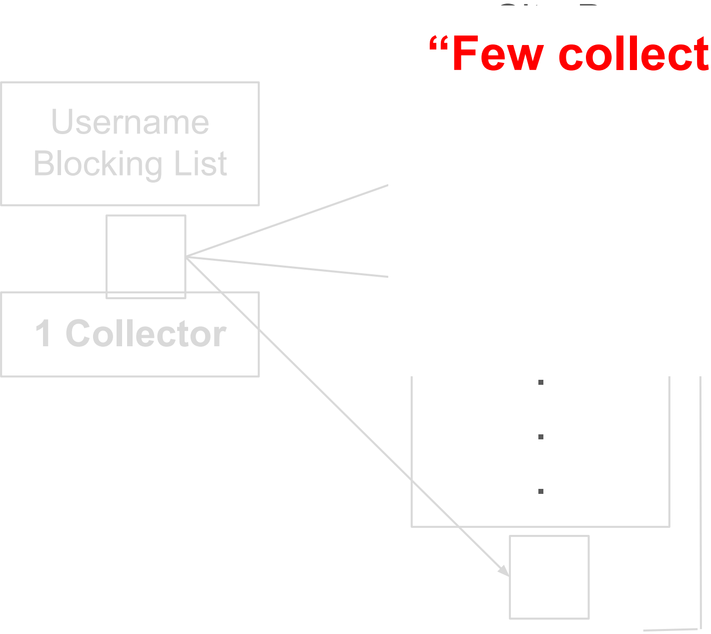
.

6 Collector - Blocked Minimum 99.0% attacks

Performance based on number of Collectors

- To examine the effect of the number of collectors on blocking performance, we computed Username Blocking List from various number of collectors.

“Few collectors can also perform effective blocking”



.6% attacks
.4% attacks

6 Collector - Blocked Minimum 99.0% attacks

Conclusion

Conclusion

- Analysis aided our development of blocking mechanism.

Conclusion

- Analysis aided our development of blocking mechanism.

- Dictionary Based Blocking is a easy to computer, light-weight mechanism.

Conclusion

- Analysis aided our development of blocking mechanism.
- Dictionary Based Blocking is a easy to computer, light-weight mechanism.
- Dictionary Based Blocking outperform existing mechanism by significant margin.

Conclusion

- Analysis aided our development of blocking mechanism.
- Dictionary Based Blocking is a easy to computer, light-weight mechanism.
- Dictionary Based Blocking outperform existing mechanism by significant margin.

…..

Paper has more insights.

Questions

…