

Nizar Akbar Meilani

System Administrator

by ■ positive technologies

```
# Create buttons for keys
keys = ['Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P']
in the following keys:
    button = tk.Button(root, text=key, width=5,
command=lambda k=key: key_pressed(k))
    button.pack(side=tk.LEFT)
```

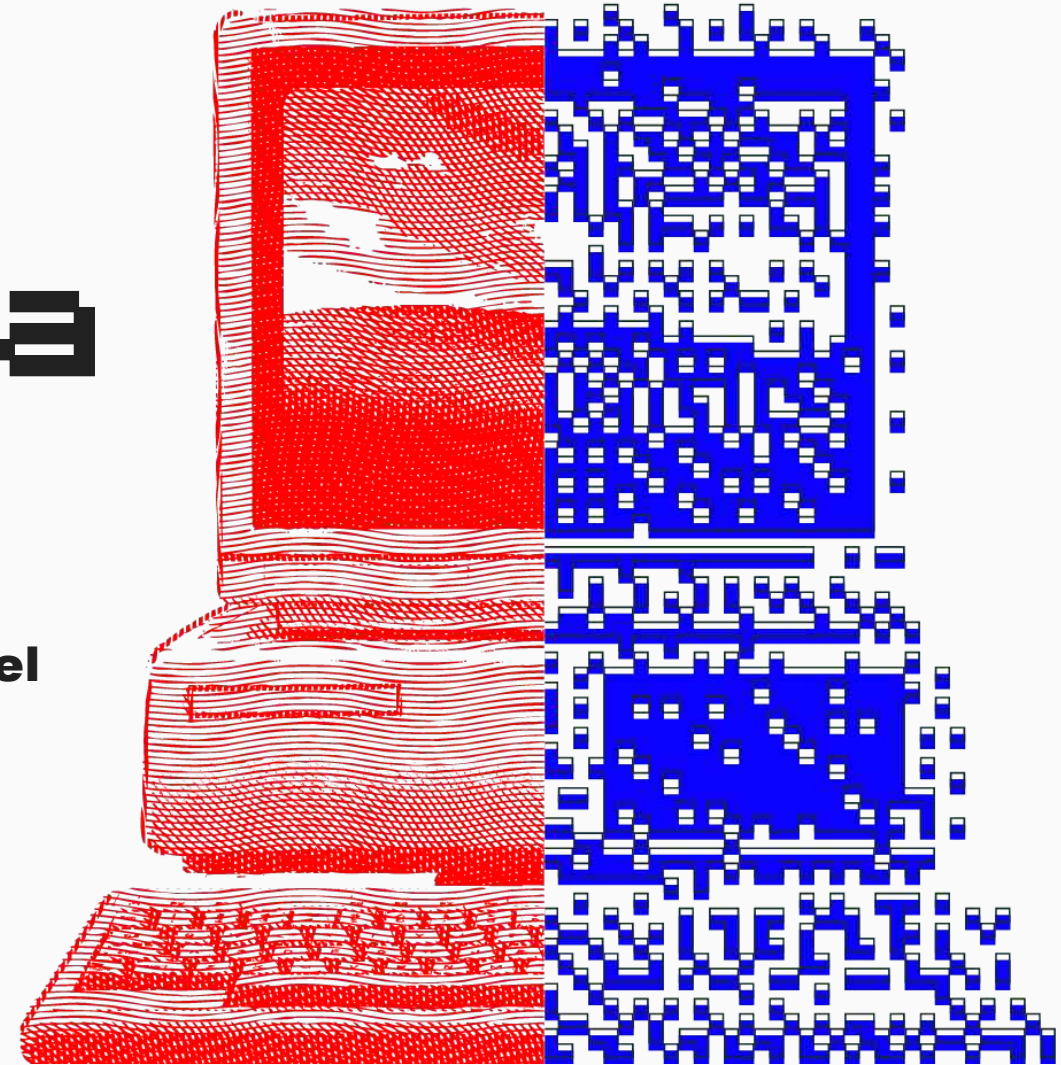
Costly DDoS Protection is a Challenge

- **IPList to BPF Rule Delivers Efficient, Domain-level Mitigation**

positive
Jakarta hack
talks

#PHTalks

```
# Create the main window
root = tk.Tk()
root.title("Simple Keyboard")
```



The speaker



A SysAdmin

- A four years Tech. Support and Two Years SysAdmin
- Who passionates about Linux and little bit Security
- Enthusiast of open-source community and knowledge sharing (still learning and not yet a contributor)

Contacts

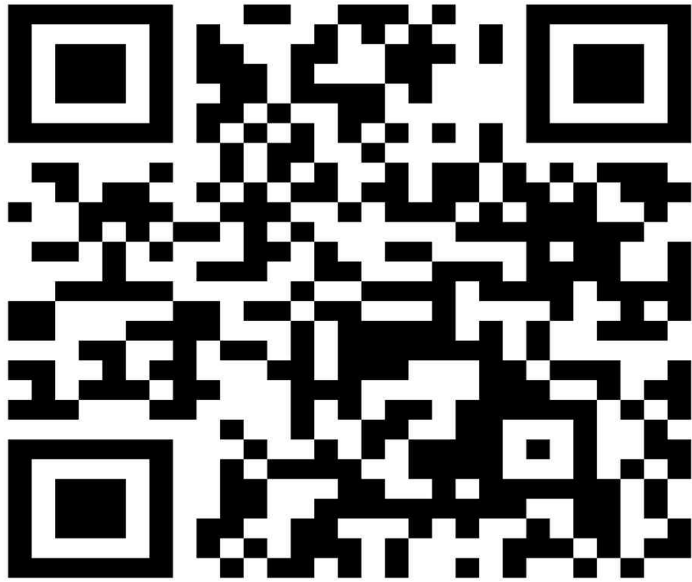


Nizar Akbar Meilani

System
Administrator

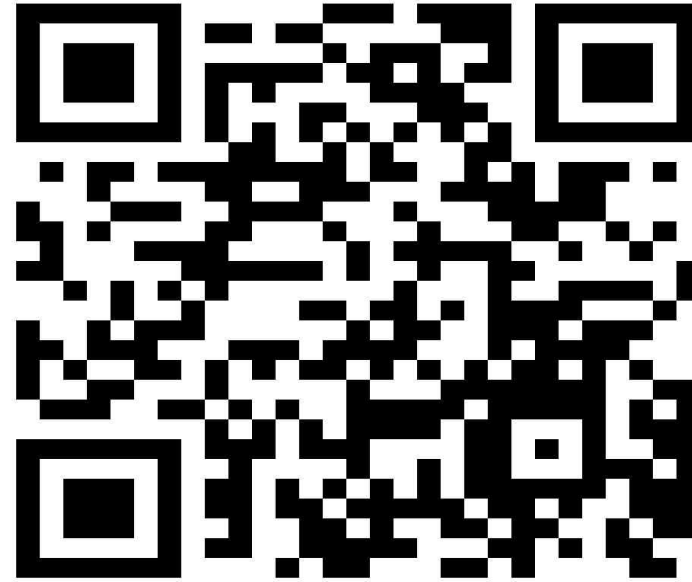
✉ nizarakbarm04@gmail.com

My Other Information



LinkedIn

 <https://s.id/lnkniz>



Website

 Nizarakbar.com

01

by ■ positive technologies

```
def key_pressed(key):  
    print(f"{key} pressed")
```

```
# Run the application  
root.mainloop()
```

Background

- **What is DDoS?**
Is the attack really unintended?

```
# Create buttons for keys  
keys = ['Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P']  
in the following keys:  
    button = tk.Button(root, text=key, width=5,  
        command=lambda k=key: key_pressed(k))  
    button.pack(side=tk.LEFT)
```



#PHTalks

What is DDoS?

Malicious attempt with the intent to disrupt normal traffic at the targeted server, service, or network

Usually done by overwhelming the target with a bunch of request

What OSI Layer DDoS Attack we will focusing?

DDoS could target layer 7, layer 4, or even layer 3

But we only focus on layer 7 currently

Is the attack targeted and has intention?

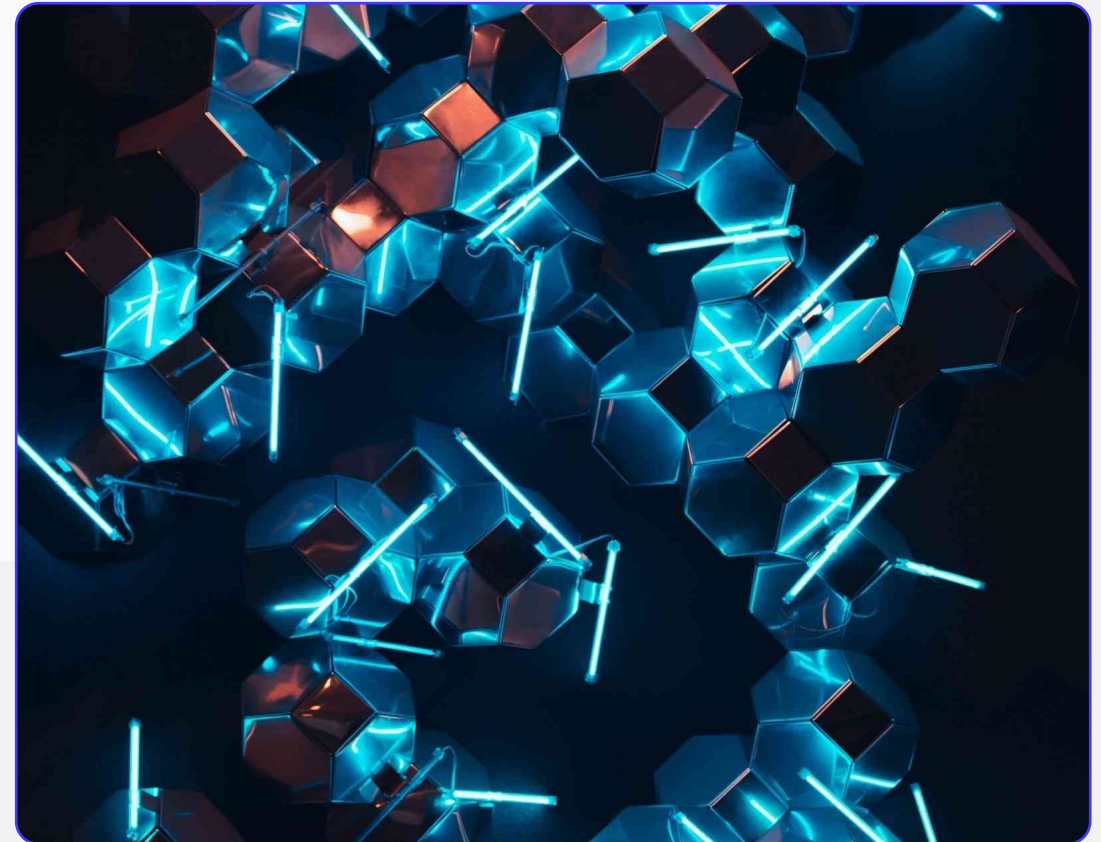
Is the attack targeted?

Some of the attack lately targeted and with specific campaign

What DDoS Attack campaign currently (in Indonesia) focused?

DDoS Attack to governmental, university, and school domain

It has intention for specific cyber attack campaign especially the one currently become trend in Indonesia



Threat Landscape Indonesia & DDoS

The attack generally started with some vulnerability injection, continued to web defacement especially gambling phishing campaign, and escalated to DDoS to all Search Engine Result Page

5780

Web Defacement [1]

4071

Web Defacement related to Gambling
Phishing Campaign [1]

43879

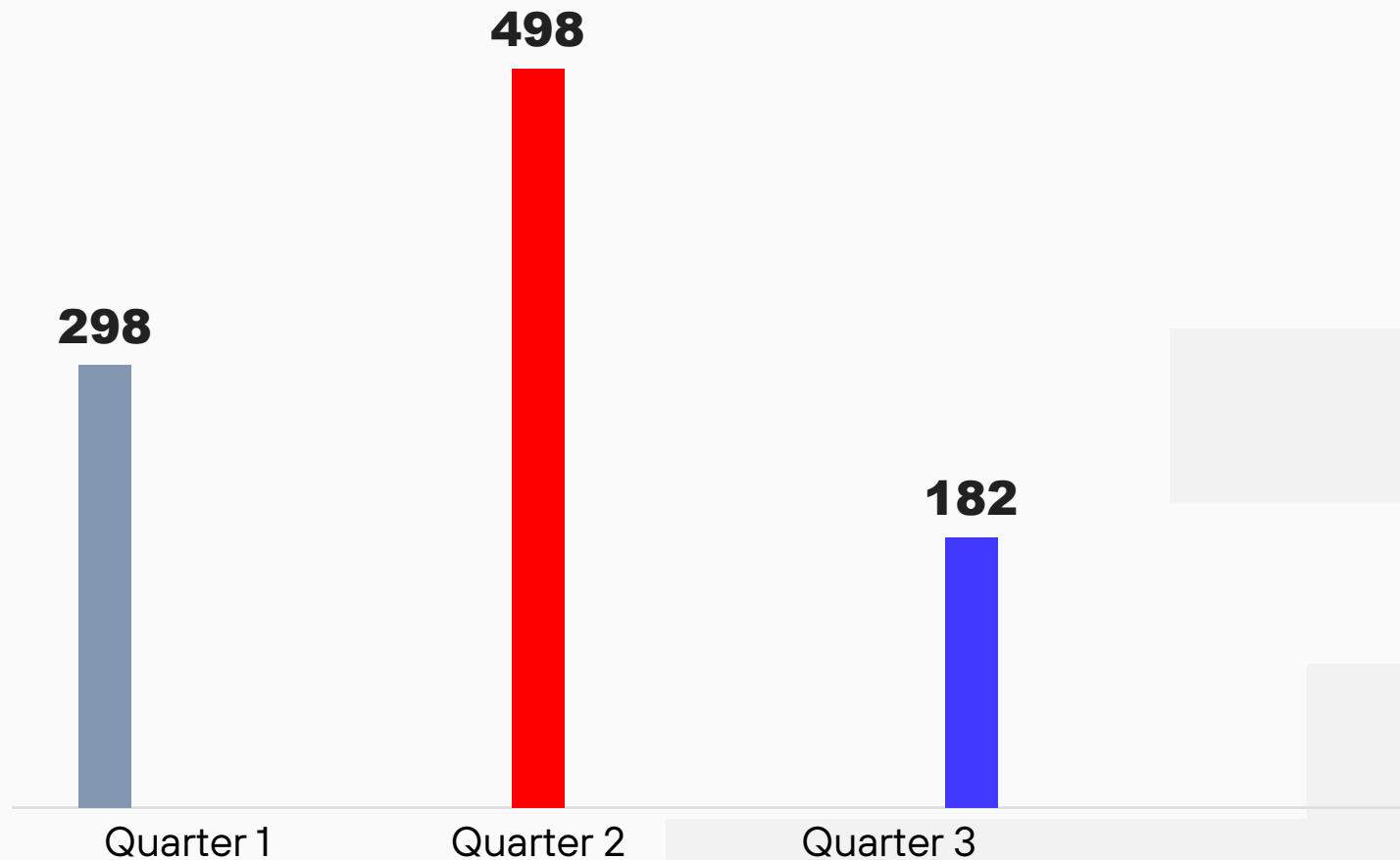
DDoS Incidents [2]

BSSN Security Landscape 2024 [1]

SoC Radar Indonesia Threat Landscape Report 2024 [2]

Our Threat Data

Gambling Phishing Injection Statistic Indonesia at 2025



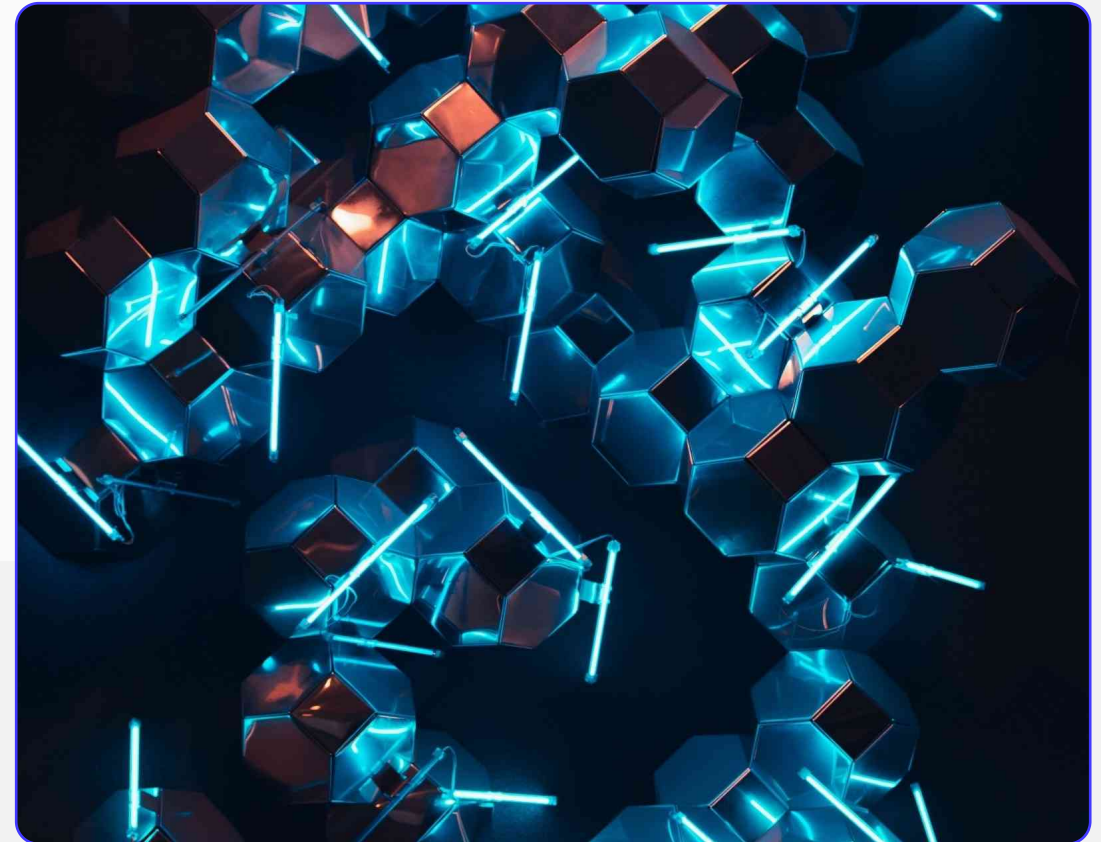
**Gambling
Phishing
Injection
Statistic 2025**

Our own statistic

Who is The Target and the Impact?

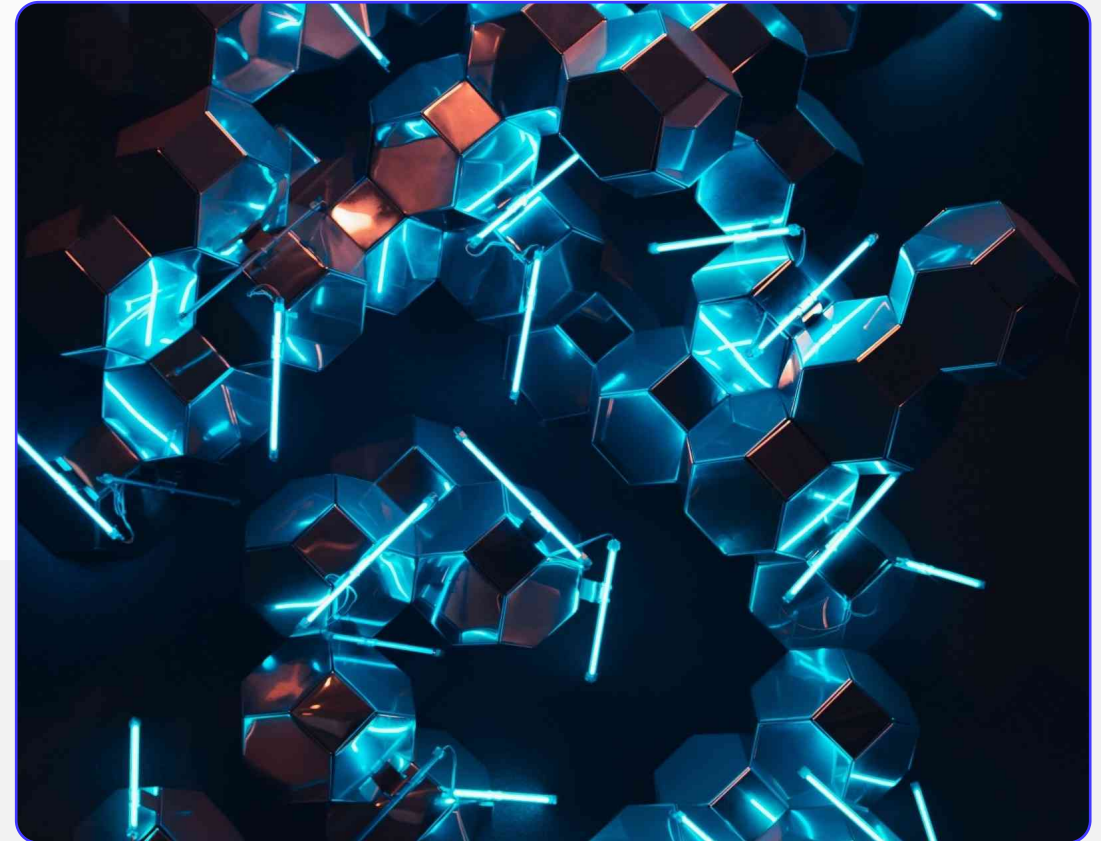
Small government, academic, university organization

- Usually target smaller organization not the larger one
- Not seen as major incident, but spread largely
- Limited financial capability



Unseen and not Impressive Threat Activity

- Can be seen as not typical APT(Advanced Persistent Threat) or unknown threat currently
- They target trusted small organization to evade detection from komdigi
Combining SERP to gain popularity at indexing
- Have a competition with DDoS as the possibility of multiple group
- The increasing of this threat made it important for protection



03

by ■ positive technologies

```
engine = pyttsx3.init()
```

The Tech

```
import pyttsx3
```

- **Combining past Tech that gaining impact lately**

```
message = "Hello, human! This is your  
computer speaking. Hack the planet!"
```



#PHTalks

eBPF : Technology from the Past and Impactful Lately

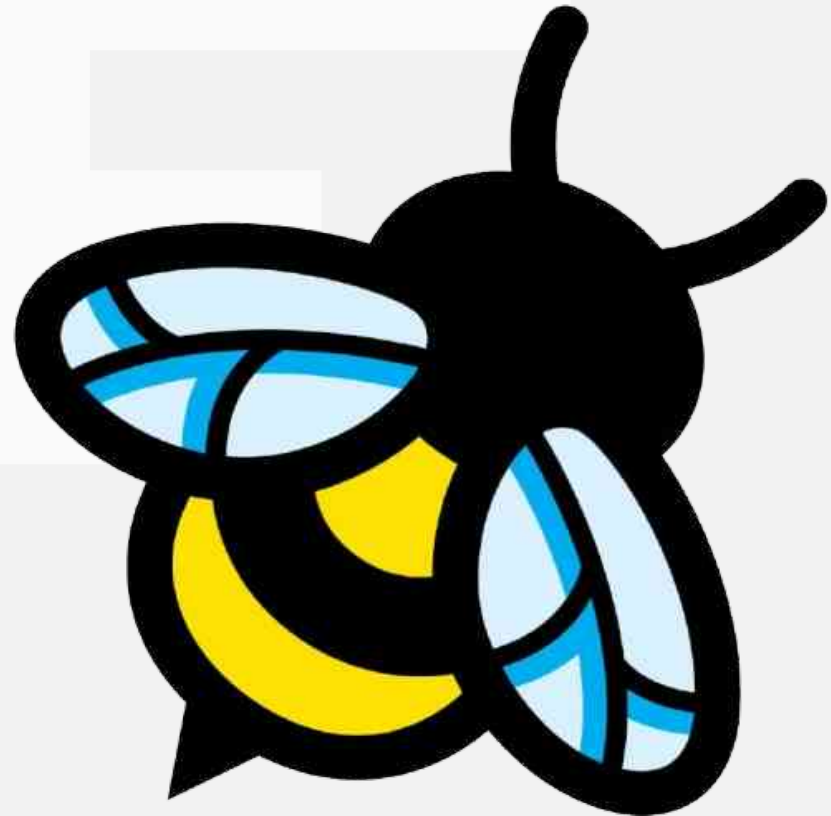
Linux Kernel Space Technologies

Allows extending kernel functionality dynamically (performance tracing, high-performance networking, and detecting & preventing malicious activity)

Used first in tcpdump (1997), improved until 2015, applied to production by Brendon from Netflix ,implemented for Katran Load Balancing of Facebook at 2014

Now

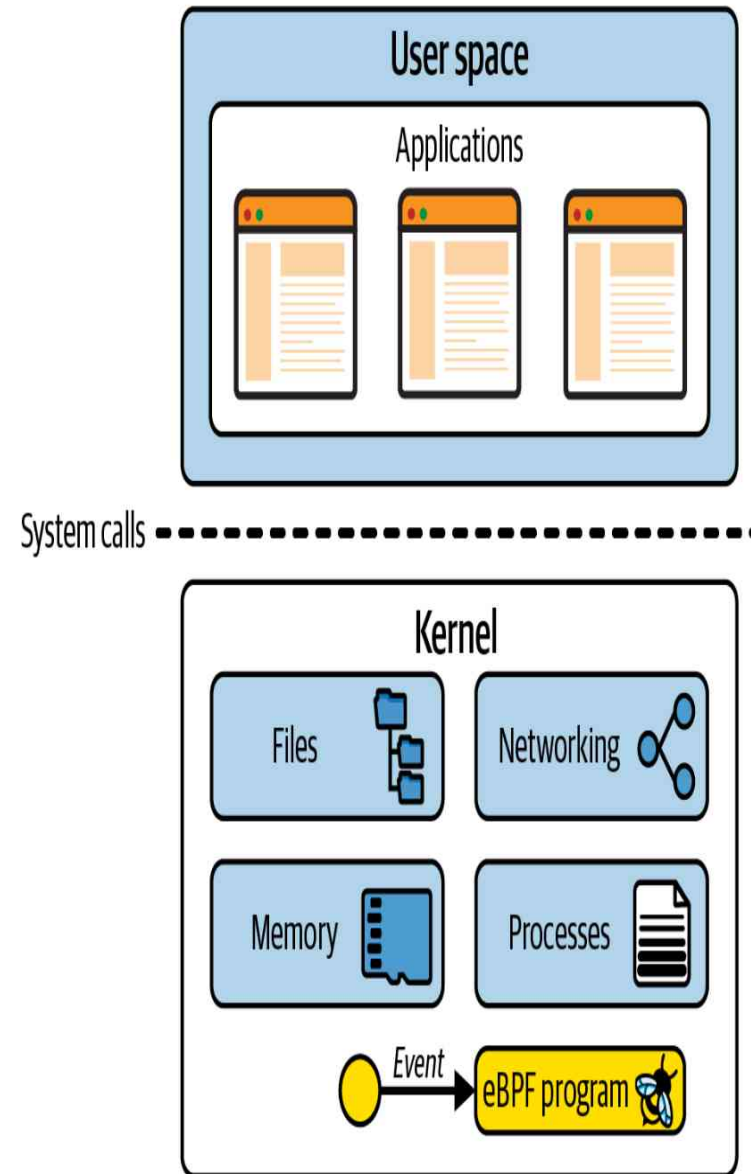
Bring more superpower for others



How it works?

Linux Kernel Space Technologies

It runs inside kernel space based on specific event of files, memory, networking, or processes (dynamically)



eBPF XDP

A program which triggered by the arrival of a network packet in kernel space



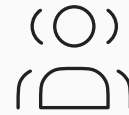
XDP_DROP

Discard the packet immediately



XDP_TX

Forward the packet (which may have been modified) to the same network interface that received it



XDP_REDIRECT

Bypasses the normal network stack and redirects the packet via another NIC to the network



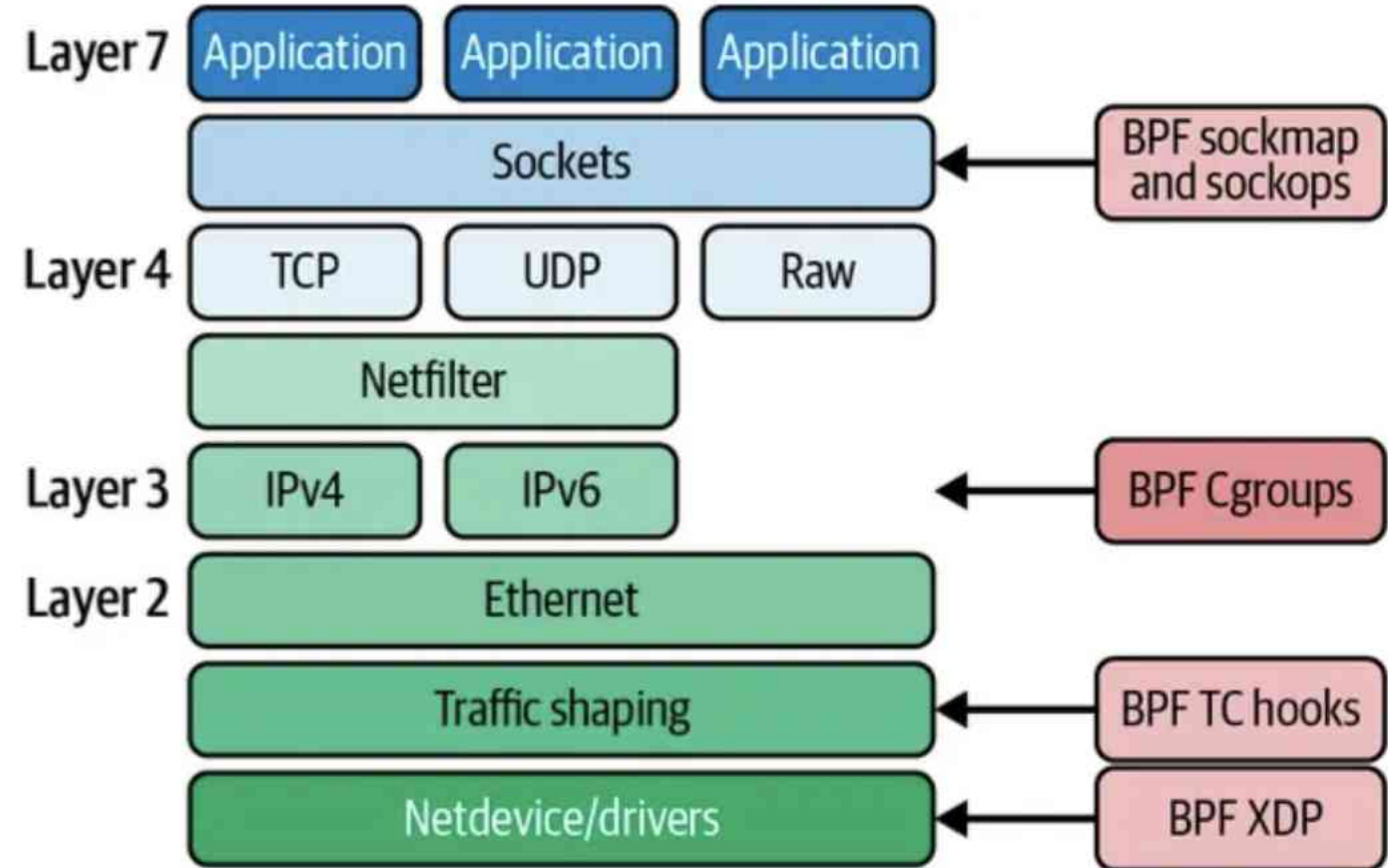
XDP_ABORTED

Packet being dropped, implies error case or something unexpected

The HOOKS

How it hooks the program

Attach to specific interface
Index for XDP



BPF program types hook into various points in the network stack

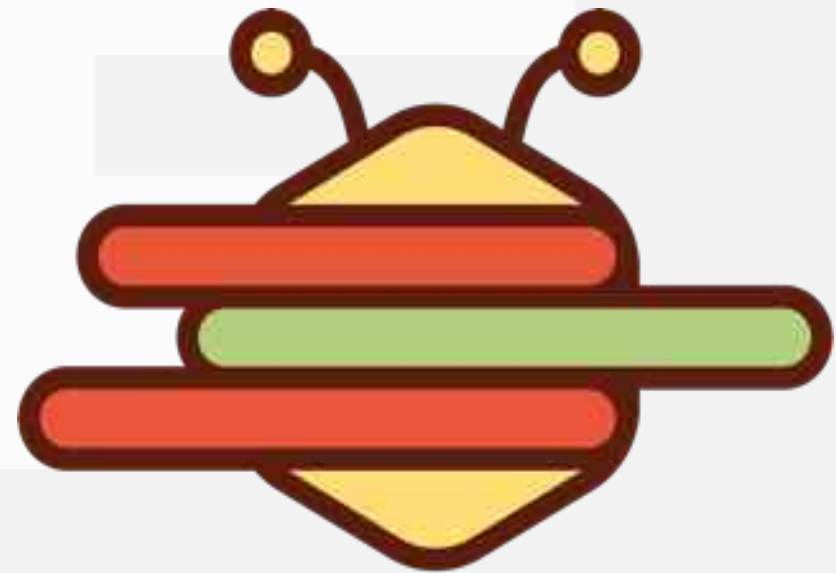
bpfilter

EBPF-based packet filtering framework

Transforms how you control network traffic by leveraging the power of eBPF

Rules translation

Translates filtering rules into optimized BPF program



bpfilter

bpfilter

Key features:

High performance (eBPF's near-native performance), **flexible integration** (could use iptables integration or bpfilter's bfccli command line), **low overhead** (minimal resource consumption with maximized efficiency), **developer friendly** (clean architecture with clear separation of components)



bpfilter

Reverse Proxy WAF

Web Application Firewall as Reverse Proxy

Filtering and monitoring HTTP traffic from Web based attacks, http-flood, and bot abused



SafeLine

03

by ■ positive technologies

```
engine = pytttsx3.init()
```

The Method

```
import pytttsx3
```

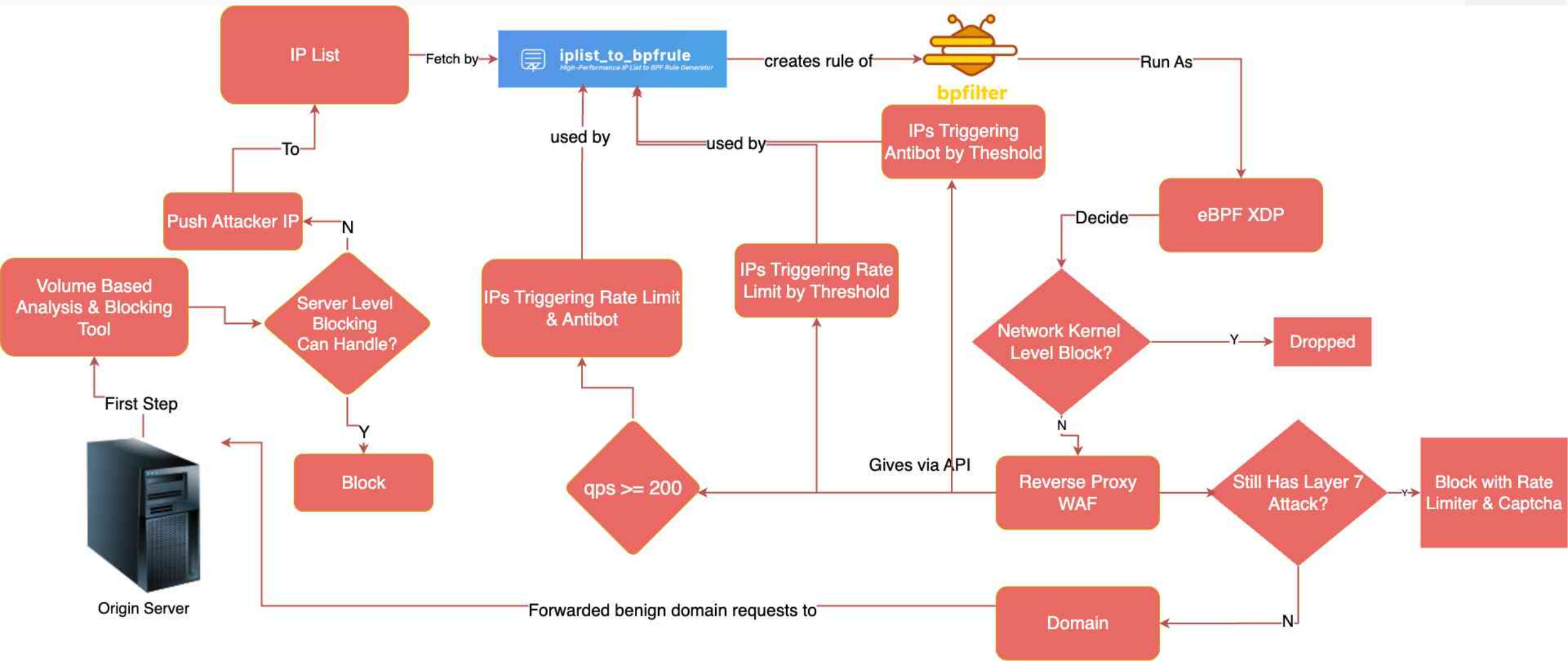
- How it do?



```
message = "Hello, human! This is your  
computer speaking. Hack the planet!"
```

#PHTalks

The Flow



IPLists to XDP Drop Rule

IPs from IPList as First Rule

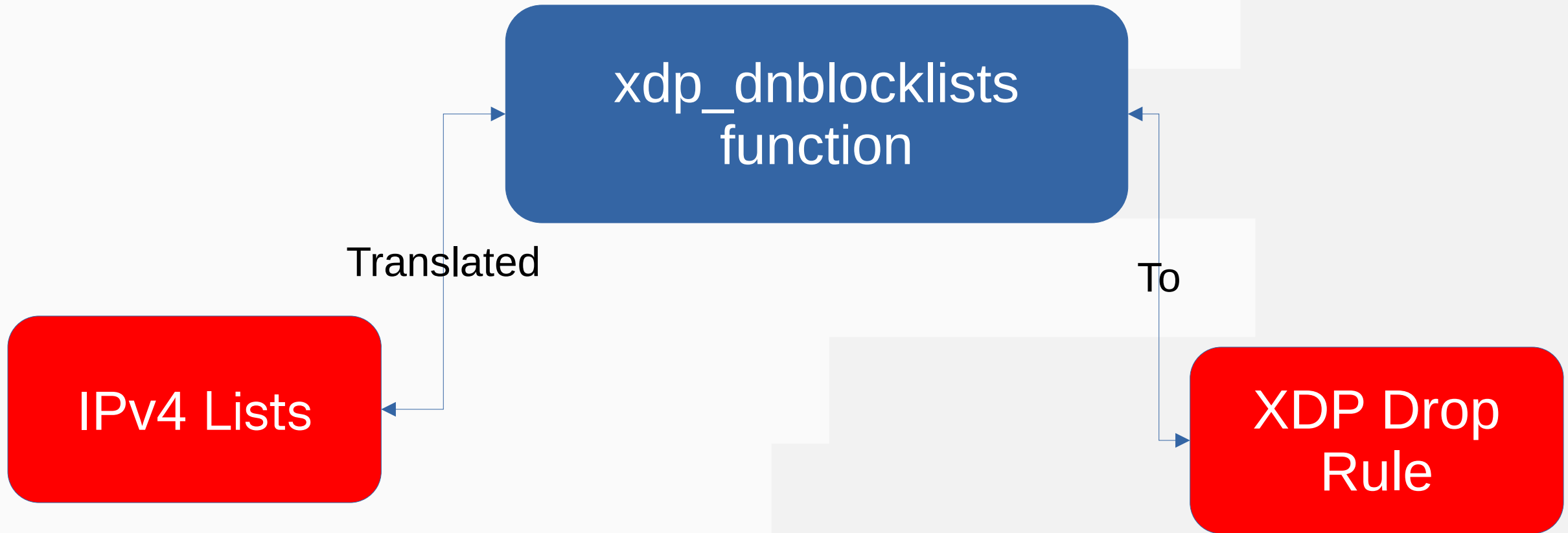
Contains all IP detected by volume based analysis before the domain DNS changed to Reverse WAF Server's IP

It also contains all IP that forwarded to the server and regarded as DDoS request too (by volume based analysis inside the origin server).

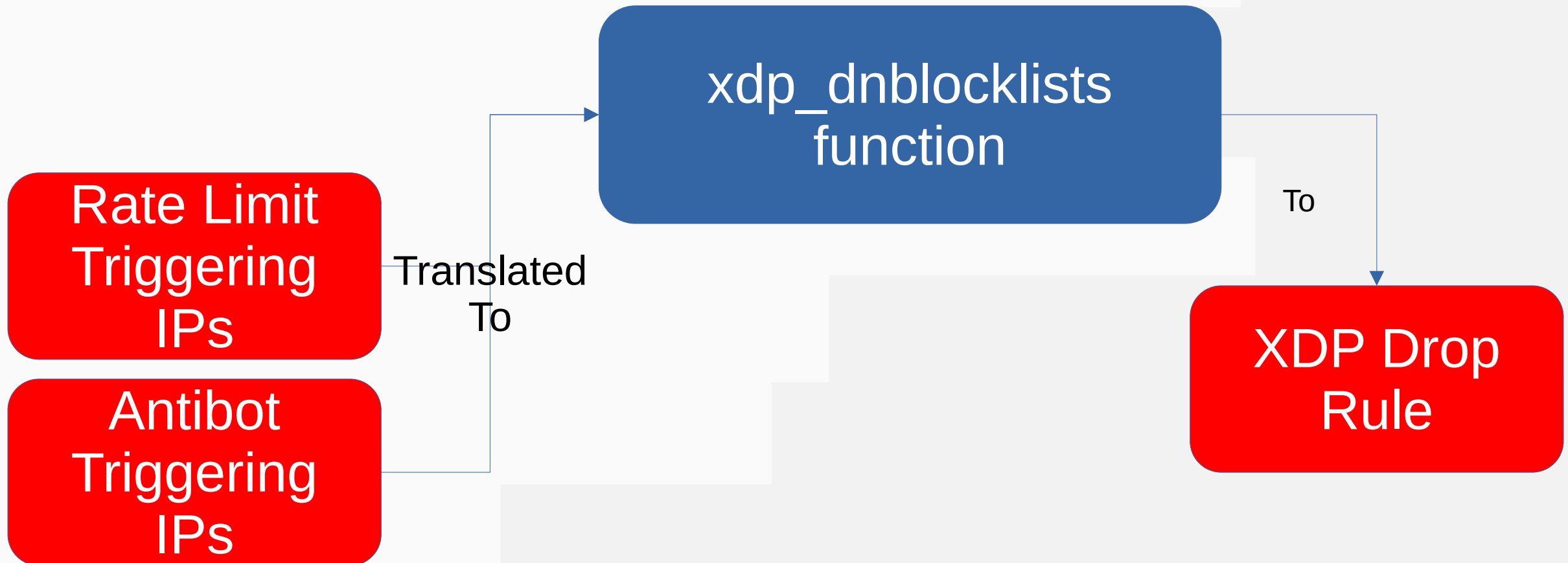
Second Rule

Contains IP from rate limit and antibot triggered event. Normally it is using some threshold in normal condition. And use strict blocking in high qps condition.

First XDP Rule



Second XDP Rule



02

```
with open('/path/to/your/flashdrive/  
myfile.txt', 'w') as file:  
    file.write('Hello, Flash Drive!')
```

by ■ positive technologies

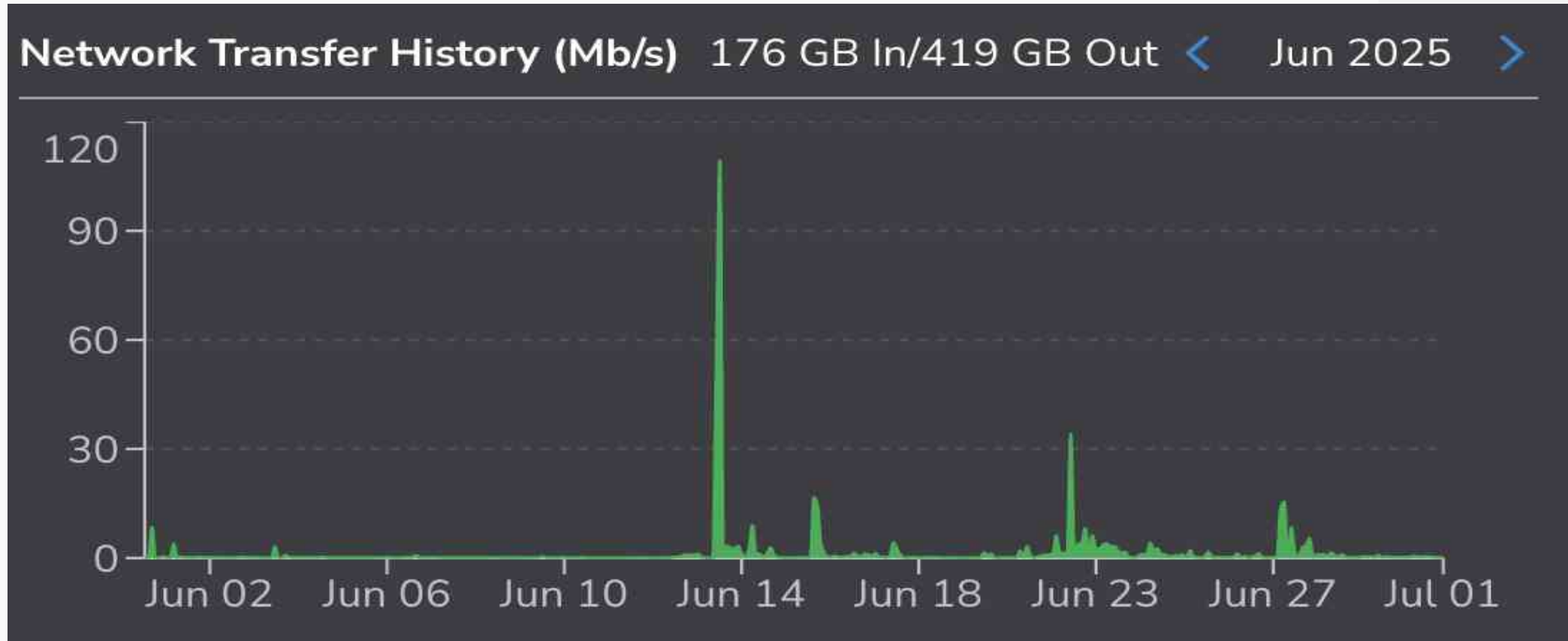
Result

- **The current result**

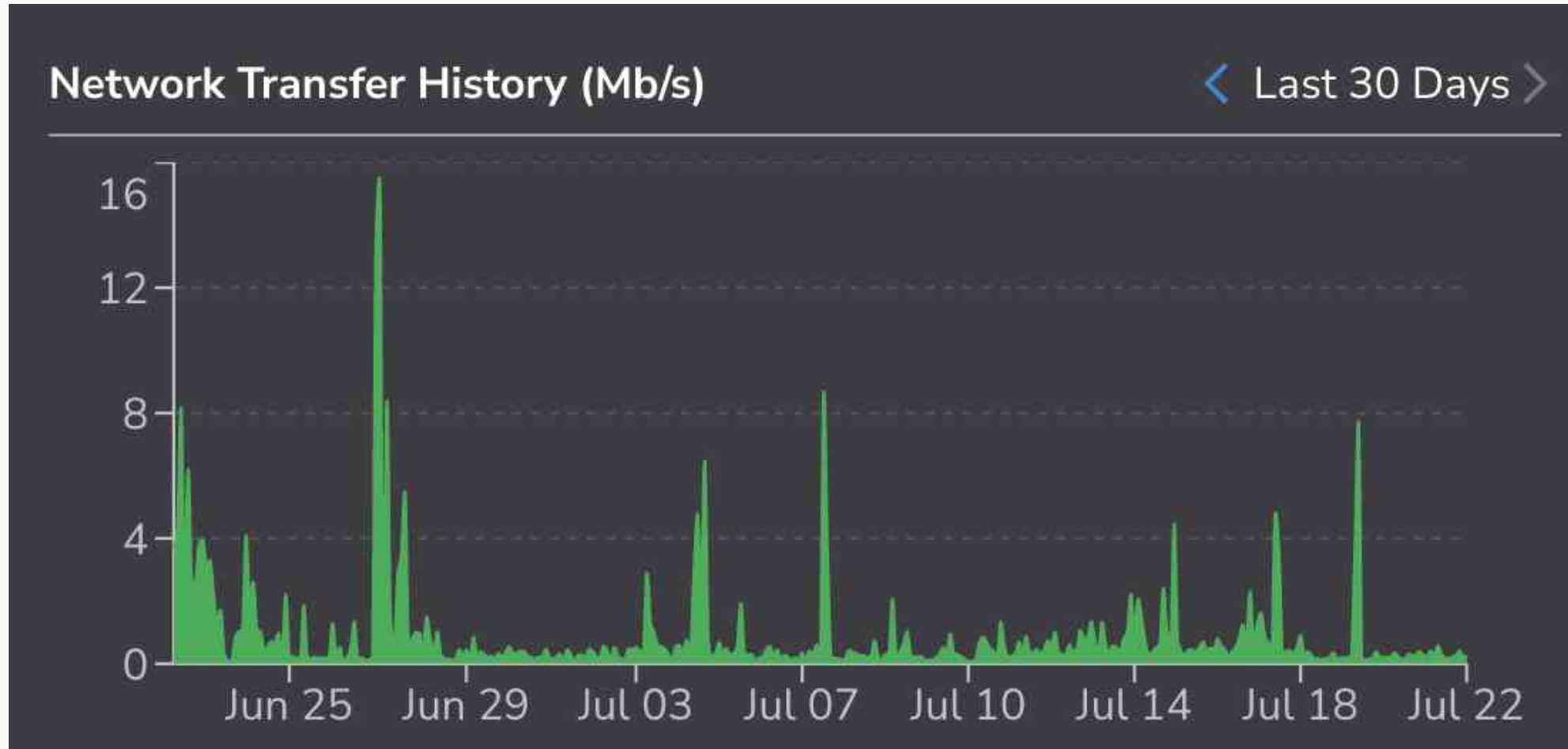
```
with open('/path/to/your/flashdrive/  
myfile.txt', 'r') as file:  
    content = file.read()  
    print(content)
```



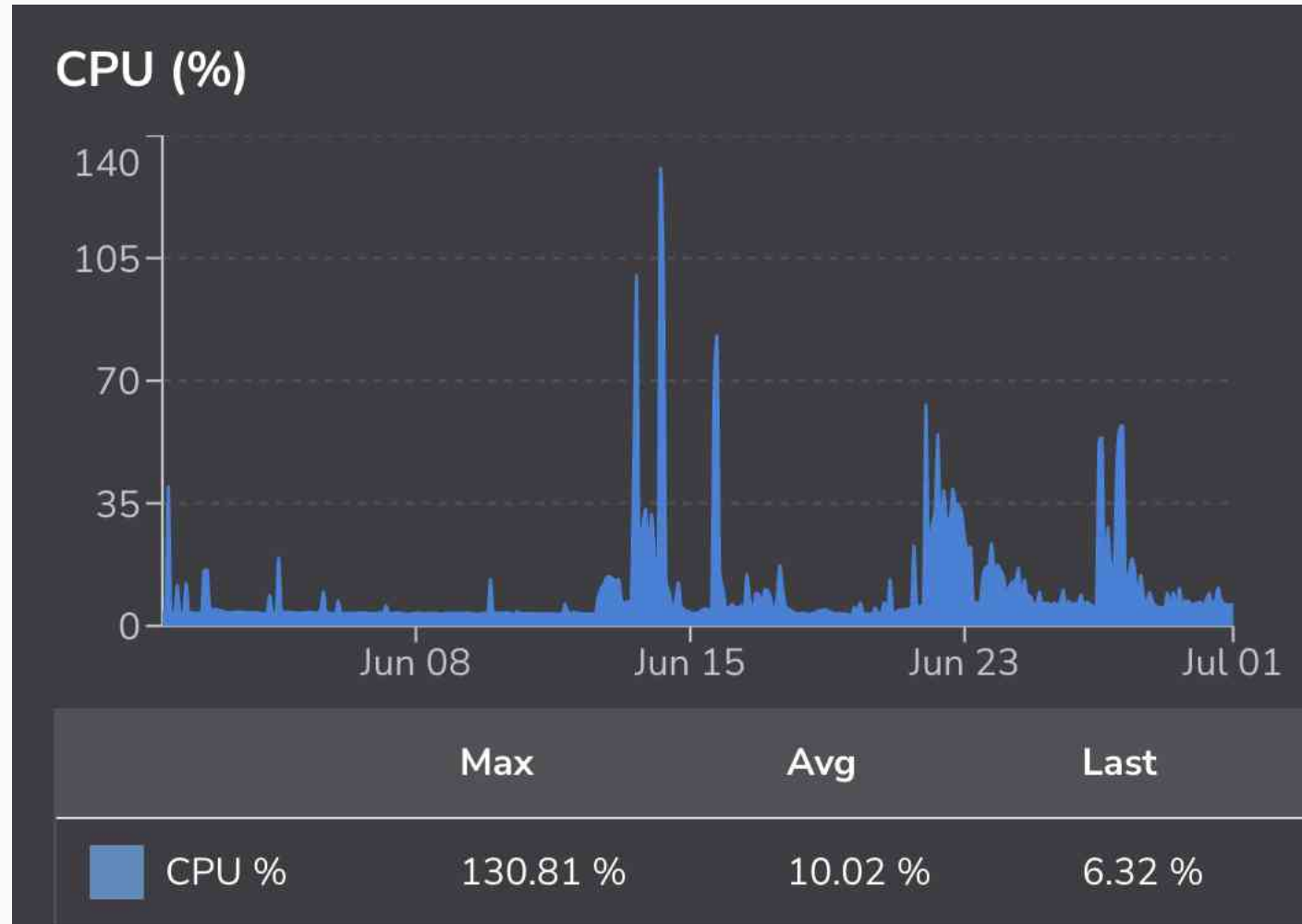
From this



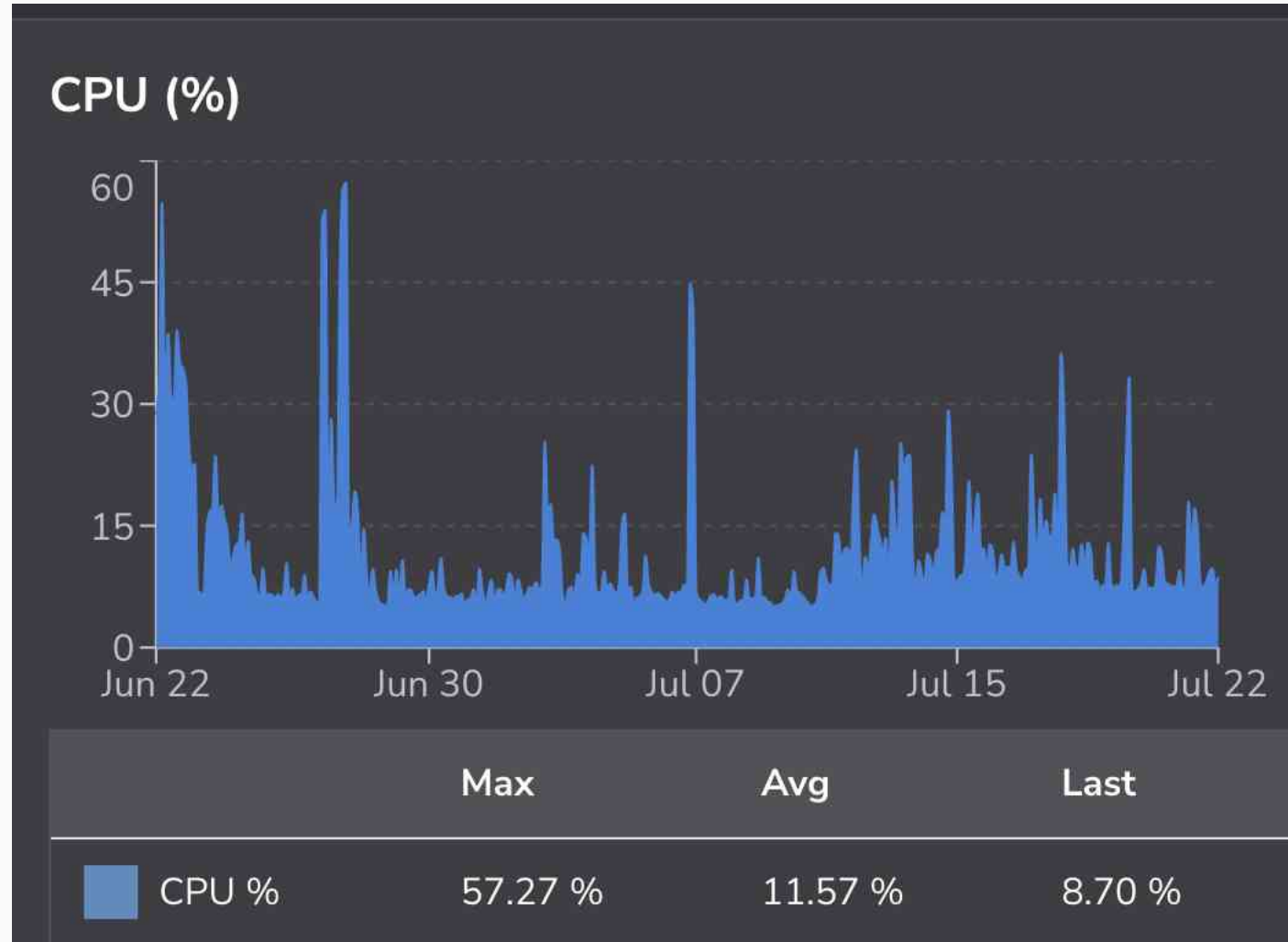
To This



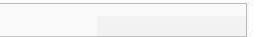
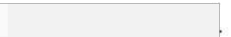
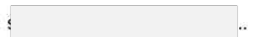
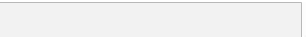
From this



From this



The protected sites

9 Applications		Application	ADVANCED		ADD APPLICATION
	 nar.sch.id	DETAIL ...			DETAIL ...
DEFENSE	Domain: 		DEFENSE	Domain: 	
RQS TD 0	BLK TD 0		RQS TD 413	BLK TD 6	
HTTP FLOOD	BOT PROTECT	AUTH	HTTP FLOOD	BOT PROTECT	AUTH
	 olo.ac.id	DETAIL ...		 lo.ac.id	DETAIL ...
DEFENSE	Domain: 		DEFENSE	Domain: 	
RQS TD 714.6k	BLK TD 694.2k		RQS TD 8.7k	BLK TD 1.4k	
HTTP FLOOD	BOT PROTECT	AUTH	HTTP FLOOD	BOT PROTECT	AUTH
	 diri.sch.id	DETAIL ...		 rb.sch.id	DETAIL ...
DEFENSE	Domain: 		DEFENSE	Domain: 	
RQS TD 4.1k	BLK TD 2.5k		RQS TD 5.7k	BLK TD 4.0k	
HTTP FLOOD	BOT PROTECT	AUTH	HTTP FLOOD	BOT PROTECT	AUTH
	 ro.sch.id	DETAIL ...		 r.id	DETAIL ...
DEFENSE	Domain: 		DEFENSE	Domain: 	
RQS TD 5.1k	BLK TD 925		RQS TD 1.7k	BLK TD 850	
HTTP FLOOD	BOT PROTECT	AUTH	HTTP FLOOD	BOT PROTECT	AUTH

Is it as always smooth?

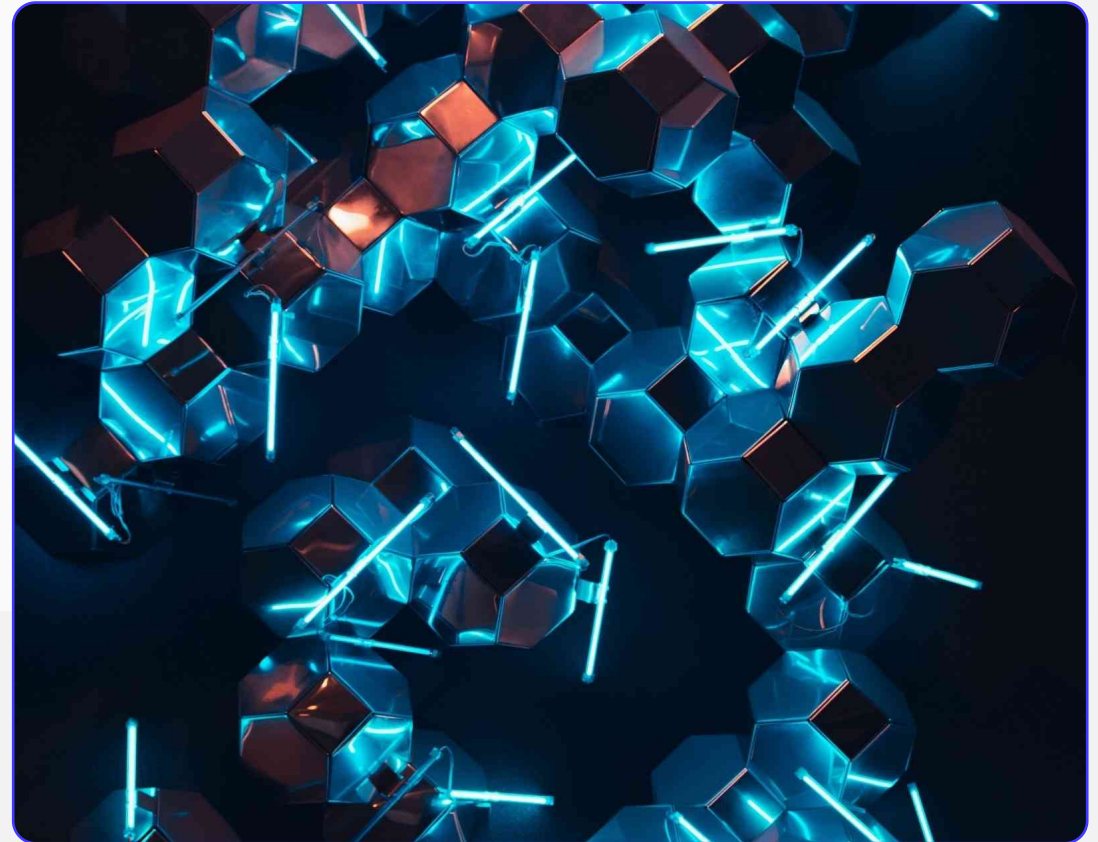
Exactly no..

Use heuristic based on attack IP for some period

Moved blocking to kernel network level to reduce heavyload of application

Better pattern matching

Antibot could restricted to specific pattern of path, need volume based analysis (still on the work), to reduce false positive, add restrict block



02

```
with open('/path/to/your/flashdrive/  
myfile.txt', 'w') as file:  
    file.write('Hello, Flash Drive!')
```

by ■ positive technologies

Addition

- **Some addition**

```
with open('/path/to/your/flashdrive/  
myfile.txt', 'r') as file:  
    content = file.read()  
    print(content)
```



#PHTalks

Same IPs Attacking Different Domain

```
ia.com | CHANGED | rc=0 >>
128 pay.com 182.253.10.21
128 pay.com 182.253.10.21
| CHANGED | rc=0 >>
400 ti.ac.id 182.253.10.21
400 ti.ac.id 182.253.10.21
| CHANGED | rc=0 >>
100 eng.ac.id 182.253.10.21
100 eng.ac.id 182.253.10.21
100 eng.ac.id 182.253.10.21
100 eng.ac.id 182.253.10.21
100 eng.ac.id 182.253.10.21
100 eng.ac.id 182.253.10.21
256 eng.ac.id 182.253.10.21
256 eng.ac.id 182.253.10.21
2025-06-14 00:03:06 [INFO] block_ips_while_ddos : Block 103.161.30.6 104.28.159.66 141
6.50.113 200.82.153.104 217.15.175.162 23.27.168.231 23.28.246.152 38.156.72.242 38.19
hile there is DDoS or DoS for
```

Same IPs Attacking Different Domain

```
| CHANGED | rc=0 >>  
100 [redacted]pro.sch.id 182.253.10.21  
100 [redacted]pro.sch.id 182.253.10.21  
[redacted]  
[redacted]a.com | CHANGED | rc=0 >>  
170 [redacted]ib.ac.id 182.253.10.21  
[redacted]sia.com | FAILED | rc=1 >>  
non-zero return code  
[redacted]a.com | CHANGED | rc=0 >>  
256 [redacted]usna.com 182.253.10.21  
256 [redacted]usna.com 182.253.10.21  
[redacted]a.com | CHANGED | rc=0 >>  
215 [redacted]eth.com 182.253.10.21  
189 [redacted]eth.com 182.253.10.21
```

Is it the same actor or connected to something?

As blue team, a question will be arise?

Is it same actor or connected to some other campaign

Next idea

Collect all uncollected data to SIEM to create pattern to draw conclusions of incident

Use EDR



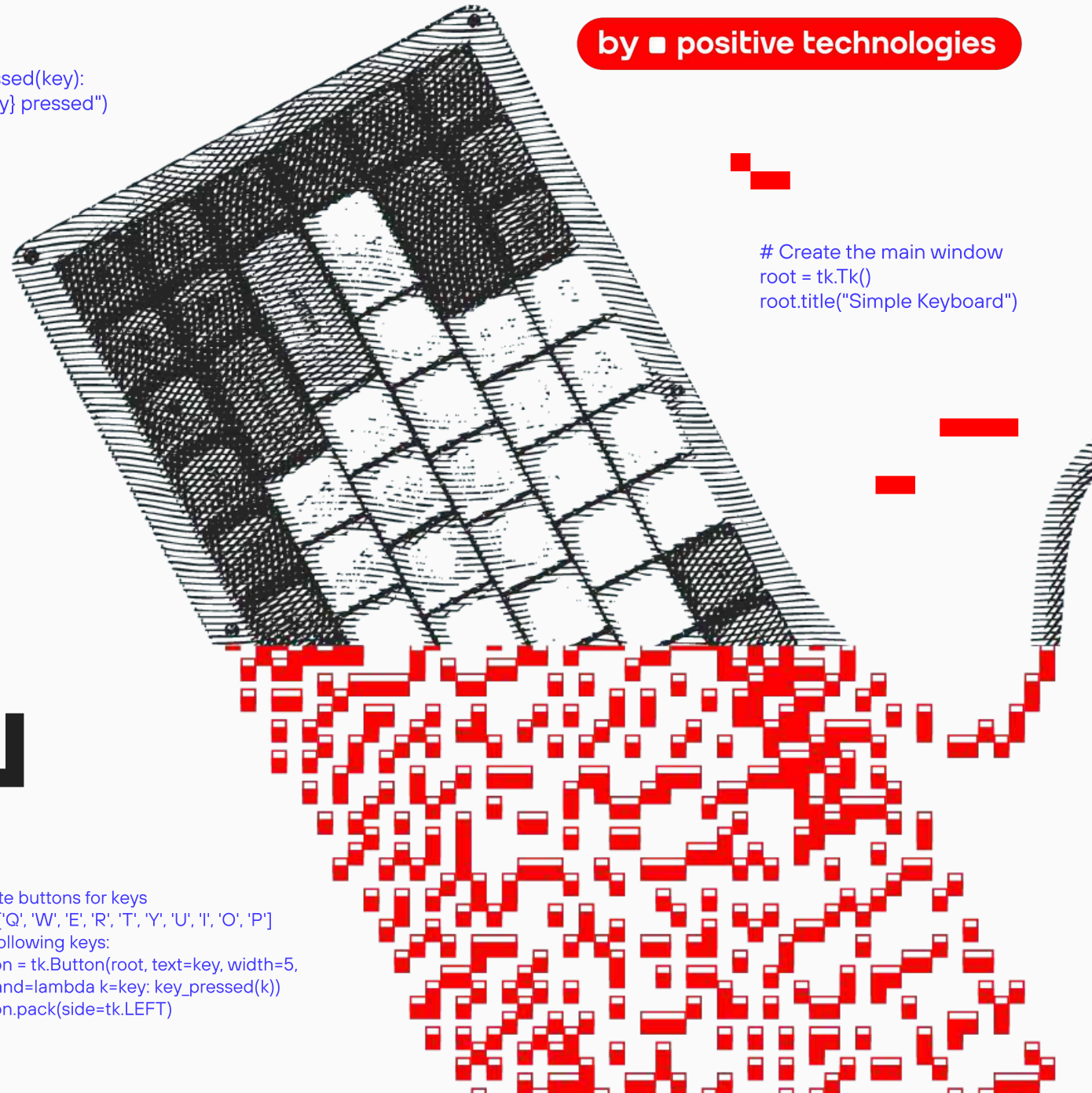
positive hack talks

Jakarta

by ■ positive technologies

```
def key_pressed(key):  
    print(f"{key} pressed")
```

```
# Create the main window  
root = tk.Tk()  
root.title("Simple Keyboard")
```



Thank you

#PHTalks

```
# Create buttons for keys  
keys = ['Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P']  
in the following keys:  
    button = tk.Button(root, text=key, width=5,  
        command=lambda k=key: key_pressed(k))  
    button.pack(side=tk.LEFT)
```