

MUM 2021

Manajemen MikroTik untuk RT/RW Net Menggunakan
WhatsApp Bot

Who Am I?



Heru Setiawan

Vocational High School Teacher, SMKN 1 Seyegan Sleman

Creator Winbot (WhatsApp & Telegram Bot for Mikrotik)

- <https://mikrotik-winbot.com>
- <https://herusetiawan.id>
- <https://www.youtube.com/herusetiawanid>

Mikrotik Academy - SMKN 1 Seyegan



Outline

What we are going to discuss

1. Why using WhatsApp Bot?
 2. WhatsApp vs Telegram Bot
 3. How to Create WhatsApp Bot?
 4. How WhatsApp Bot Work?
 5. How to Setup WhatsApp Bot
 6. How to Integrate Mikrotik API
 7. Demo
-



Search “telegram” in MUM Website

<https://mum.mikrotik.com>

Cari presentasi

[Cari](#)

7 presentasi ditemukan - "telegram"

Implementing Internet of Things (IoT) on MikroTik Product for Network Monitoring System (NMS) purpose. - Indonesia 2020

I want to tell MikroTik users how to implement IoT on MikroTik products for Network Management System (NMS). In here if there is an issue in the network products will send notification to the netw...

Tags:

Monitoring system with The Dude and Telegram - Colombia 2019

The Dude: Monitoring system. Powerful, practical and free. Advantage: * Easy integration with RB MikroTik * Installation in Windows or Linux * Web interface * Real-time monitoring * Alerts sys...

Tags:

Monitoreo de infraestructura y notificaciones usando Telegrama - Peru 2019

Resumen: Qué significa monitorear; Que se debe monitorear en nuestras instalaciones y clientes. Mostraremos diversas herramientas para monitoreo. Finalmente enseñaremos como implementar notifiacaci...

Tags:

Mikrotik + Telegram - Uzbekistan 2018

Mikrotik + Telegram

Tags:

Telegram bot + MikroTik - Kazakhstan 2018

Tags:

Telegram bot + MikroTik - Belarus 2018

Tags:

Telegram and RouterOS (chat with your RouterBoard) - Russia 2017

Использование мессенджера Telegram для получения оперативных уведомлений и выполнения удаленных типовых задач ...

Tags:



VS.



Telegram vs WhatsApp Bot

Telegram	WhatsApp
Telegram Bot	WhatsApp Business Account
@BotFather	Business Solution Partner (BSP)
Free	Paid
-	Company Legality (such as PT or CV)
-	Non Official WhatsApp Bot Account

Bekerja dengan Penyedia Solusi

Jika Anda berminat untuk terhubung dengan pelanggan Anda di WhatsApp dengan menggunakan API Business kami, kami anjurkan Anda untuk bekerja dengan [Penyedia Solusi Bisnis WhatsApp \(Business Solution Providers - BSP\)](#) dalam daftar kami. Mereka adalah komunitas global penyedia solusi pihak ketiga dengan keahlian dalam API WhatsApp Business. BSP ini dapat membantu Anda berkomunikasi dengan pelanggan Anda di WhatsApp untuk kasus penggunaan yang disetujui untuk dukungan pelanggan serta notifikasi yang sensitif waktu dan dipersonalisasi.

Sebagaimana dinyatakan dalam [buku putih](#) kami, pihak ketiga yang menawarkan layanan tidak terotorisasi di platform kami, seperti perpesanan otomatis atau massal, melakukan hal yang melanggar Ketentuan Layanan kami. Jika Anda telah bekerja dengan pihak ketiga yang menawarkan layanan tidak terotorisasi, Anda mungkin tidak dapat mengirim pesan dengan layanan mereka di WhatsApp. Kami sarankan Anda bekerja dengan BSP kami yang terotorisasi.

Daftar BSP yang dibagikan hanya untuk referensi. Kami sarankan Anda melakukan uji kelayakan sendiri untuk perusahaan yang Anda pilih untuk bekerja sama.

Buat pelanggan mencintai bisnis Anda.

Dapatkan berbagai solusi untuk kebutuhan bisnis. Ciptakan *customer experience* dan tingkatkan pertumbuhan bisnis secara signifikan.

[Hubungi Kami](#)


Qiscus is one of the WhatsApp Solution Provider in Indonesia

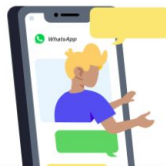
How to Setup WhatsApp Bot

<https://www.qiscus.com/qiscus-whatsapp-integration?lang=id>

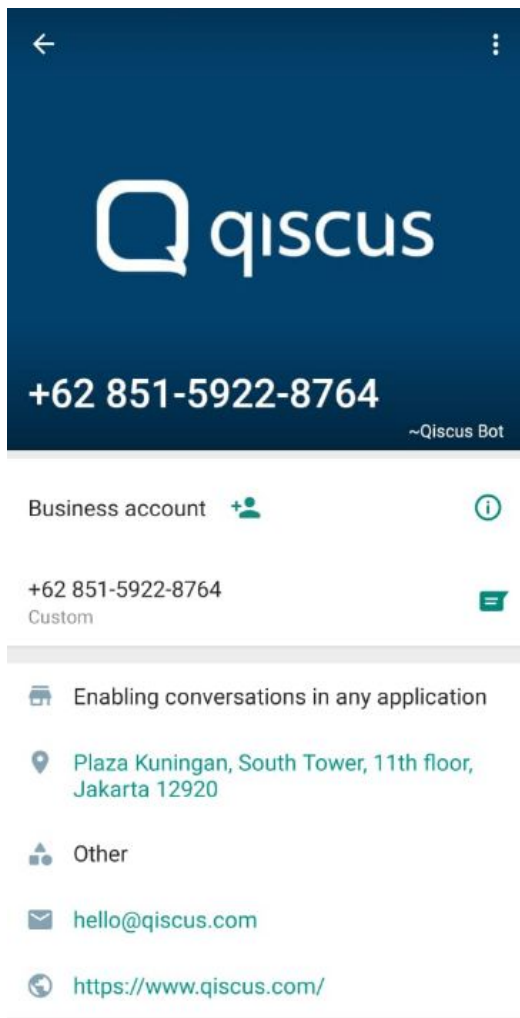
WhatsApp API Resmi dan Official

Terhubung Dengan Semua Pelanggan Anda. Dimanapun.

Ciptakan kenyamanan pengguna yang luar biasa kepada lebih dari 2 Milyar orang di berbagai negara dengan melalui aplikasi pesan instan yang paling populer di dunia.



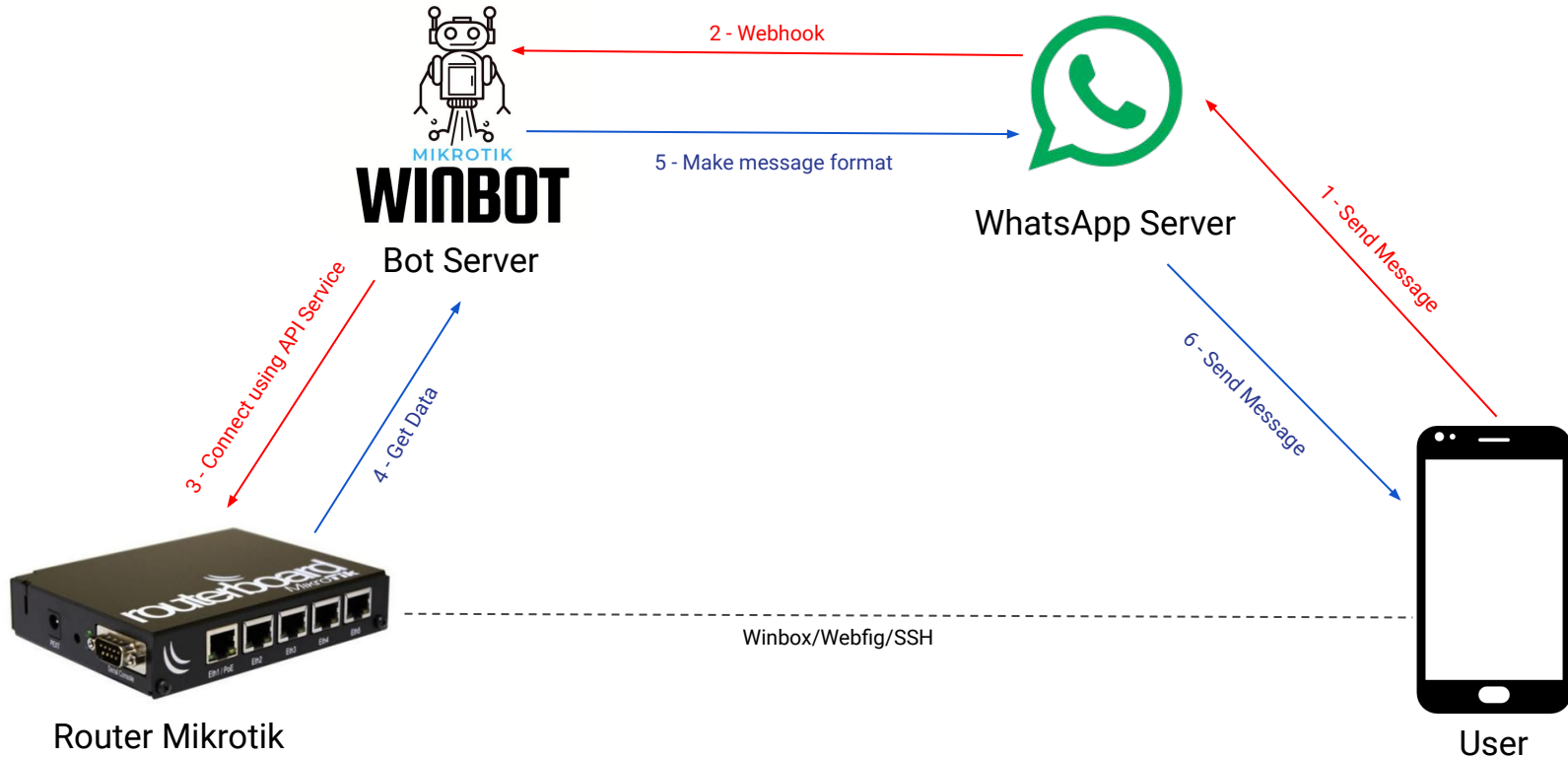
[Hubungi Kami](#)

The Example of WhatsApp Bot

1. Dashboard
2. Token
3. Access WhatsApp API (Connect to Server)

How WhatsApp Bot Work

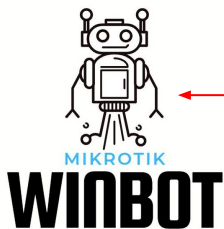


Bot Server (Winbot) Components

1. **Golang** as Main Programming Language
2. **Go-routeros** as Library to Connect with Mikrotik
3. **Heroku** to Host the Code



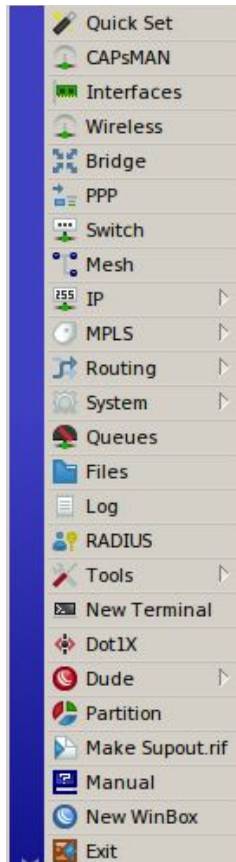
Winbot Webhook

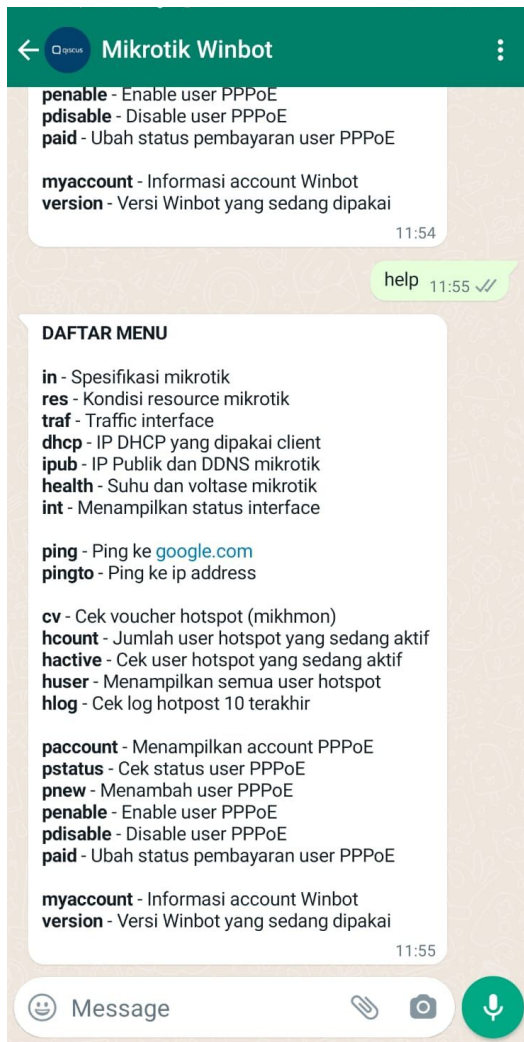


2 - Webhook



```
1 {
2   "app_code": "asdf",
3   "channel_id": 1234,
4   "contacts": [
5     {
6       "profile": {
7         "name": "Heru Setiawan"
8       },
9       "wa_id": "6285123456789"
10    }
11  ],
12  "messages": [
13    {
14      "from": "6285123456789",
15      "id": "ABCDEFGHIJKLMNOPGRSTU",
16      "text": {
17        "body": "test"
18      },
19      "timestamp": "1637506840",
20      "type": "text"
21    }
22  ]
23 }
```





List of Bot Menu

Here is the predefined message to
get mikrotik data

Mikrotik API Service



3 - Connect using API Service



Notes :

1. Mikrotik must be public accessible
2. API service is enable

IP > Services

IP Service List					
Name	Port	Available From	Certificate	TLS Vers.	
api	8728				
api-ssl	8729		none	any	
ftp	21				
ssh	26				
X telnet	23				
winbox	8291				
www	81				
www-ssl	443		Hotspot	any	

8 items

IP Public

VPN Remote

Show User Hotspot (online)

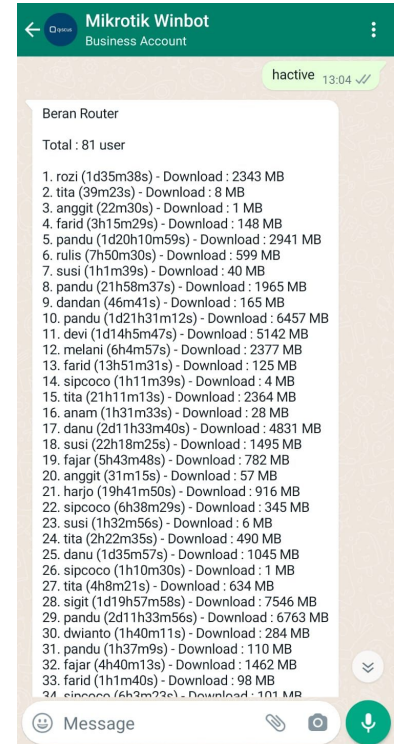
/ip hotspot active print

```
Terminal
[heru@Beran-Router] > /ip hotspot active print
Flags: R - radius, B - blocked
```

#	USER	ADDRESS	UPTIME	SESSION-TIME-LEFT	IDLE-TIMEOUT
0	farid	192.168.10.6	9h14m20s		1h
1	rozi	192.168.10.8	8h42m8s		1h
2	tita	192.168.10.11	5h40m5s		1h
3	farid	192.168.10.13	2h12m28s		1h
4	pandu	192.168.10.14	1d4h17m29s		1h
5	rulis	192.168.10.15	5h21s		1h
6	susi	192.168.10.16	5h47m35s		1h
7	pandu	192.168.10.17	6h5m7s		1h
8	dandan	192.168.10.18	1d13h8m14s		1h
9	farid	192.168.10.19	8h58m44s		1h
10	pandu	192.168.10.20	1d5h37m42s		1h
11	devi	192.168.10.21	22h12m17s		1h
12	melani	192.168.10.22	1h29m57s		1h
13	tita	192.168.10.25	5h17m43s		1h
14	nanta	192.168.10.27	2h33m7s		1h
15	dandan	192.168.10.28	3h20m54s		1h
16	anam	192.168.10.29	37m26s		1h
17	danu	192.168.10.30	1d19h40m10s		1h
18	susi	192.168.10.31	6h24m55s		1h
19	heru	192.168.10.32	6h56m18s		1h
20	anggit	192.168.10.34	4h51m22s		1h

```

-- [Q quit|D dump|down]
```



Get Data from Mikrotik using API

```
func GetHotspotSessionShow() (result string, err error) {  
    reply, err := GetConnection().Run("/ip/hotspot/active/print")  
  
    if err != nil {  
        sentry.CaptureException(err)  
        return  
    }  
}
```

Send Message to WhatsApp Server

```
func SendTextMessage(number string, message string) {  
    client := resty.New()  
  
    _, err := client.R().  
        SetHeader("Content-Type", "application/json").  
        SetHeader("Authorization", qiscusToken).  
        SetHeader("app_id", qiscusAppId).  
        SetBody(`{  
            "recipient_type": "individual",  
            "to": "` + number + `",  
            "type": "text",  
            "text": {  
                "body": "` + message + `"  
            }  
        }`).  
        Post(fmt.Sprintf("https://multichannel.qiscus.com/whatsapp/v1/%v/%v/messages", qiscusAppId, qiscusChannelId))  
  
    if err != nil {  
        sentry.CaptureException(err)  
        return  
    }  
}
```

Check Mikrotik Resource

```
/system resource print
```

```
Terminal
[heru@Beran-Router] > /system resource print
    uptime: 2d11h41m21s
    version: 6.47.2 (stable)
    build-time: Aug/13/2020 06:39:44
    factory-software: 6.45.3
    free-memory: 921.3MiB
    total-memory: 1024.0MiB
    cpu: ARmv7
    cpu-count: 4
    cpu-frequency: 716MHz
    cpu-load: 8%
    free-hdd-space: 401.2MiB
    total-hdd-space: 512.3MiB
    architecture-name: arm
    board-name: RB450Gx4
    platform: MikroTik
[heru@Beran-Router] >
```



Get Data from Mikrotik using API

```
func GetResource() (result string, err error) {  
    reply, err := GetConnection().Run("/system/resource/print")  
  
    if err != nil {  
        sentry.CaptureException(err)  
        return  
    }  
}
```

Send Message to WhatsApp Server

```
func SendTextMessage(number string, message string) {  
    client := resty.New()  
  
    _, err := client.R().  
        SetHeader("Content-Type", "application/json").  
        SetHeader("Authorization", qiscusToken).  
        SetHeader("app_id", qiscusAppId).  
        SetBody(`{  
            "recipient_type": "individual",  
            "to": "` + number + `",  
            "type": "text",  
            "text": {  
                "body": "` + message + `"  
            }  
        }`).  
        Post(fmt.Sprintf("https://multichannel.qiscus.com/whatsapp/v1/%v/%v/messages", qiscusAppId, qiscusChannelId))  
  
    if err != nil {  
        sentry.CaptureException(err)  
        return  
    }  
}
```

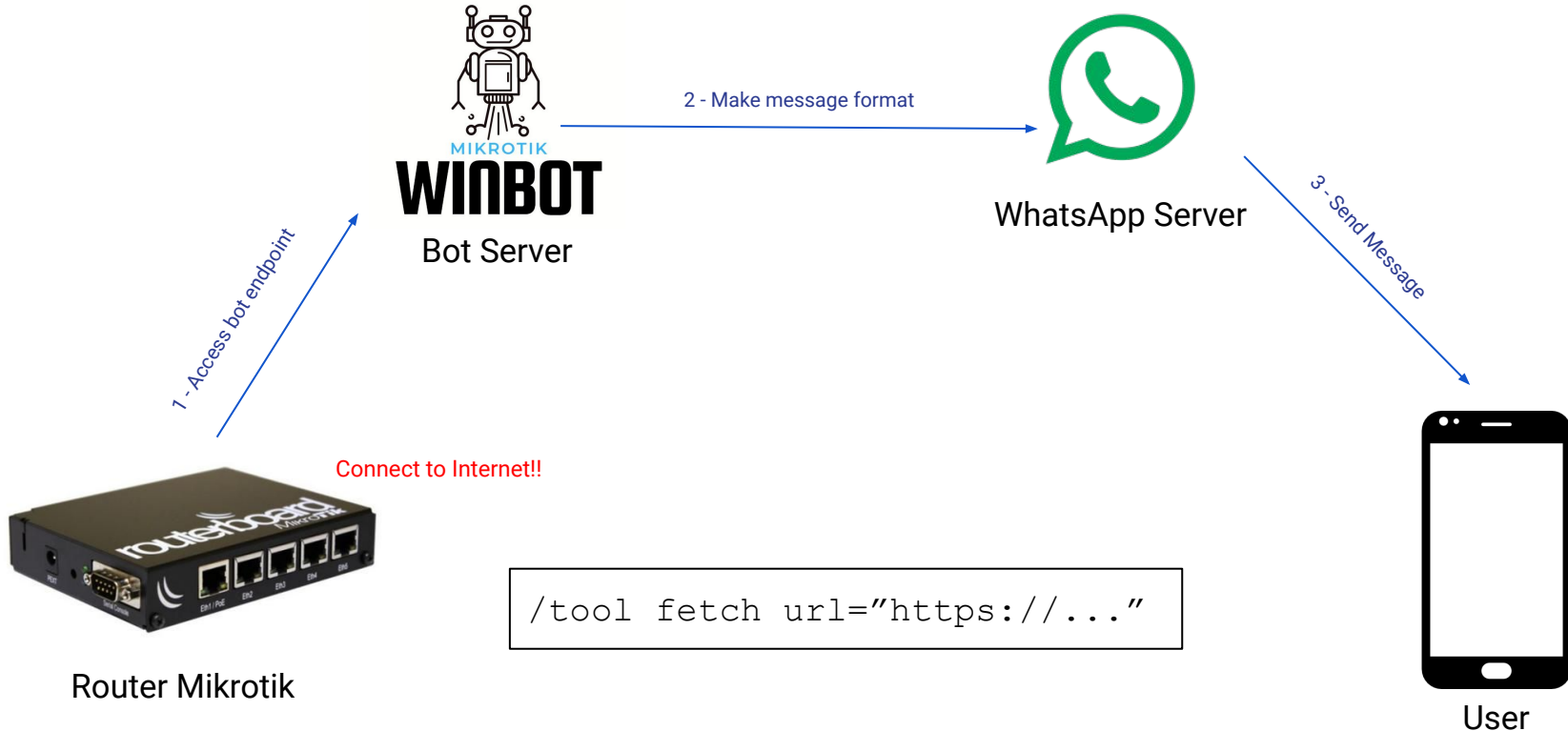
Log			
Freeze		all	
Nov/21/2021 21:17:27	memory	system, info, account	user heru logged in from 192.168.72.1 via api
Nov/21/2021 21:17:35	memory	system, info, account	user heru logged out from 192.168.72.1 via api
Nov/21/2021 21:20:26	memory	hotspot, info, debug	iin (192.168.10.146): logged out: idle timeout
Nov/21/2021 21:20:26	disk	hotspot, info, debug	-> iin (192.168.10.146): logged out: idle timeout
Nov/21/2021 21:20:52	memory	system, error, critical	login failure for user admin from 192.168.10.180 via ssh
Nov/21/2021 21:20:54	memory	system, error, critical	login failure for user admin from 192.168.10.180 via ssh
Nov/21/2021 21:20:59	memory	system, error, critical	login failure for user admin from 192.168.10.180 via ssh
Nov/21/2021 21:21:23	memory	hotspot, info, debug	heru (192.168.10.170): logged out: user request
Nov/21/2021 21:21:23	disk	hotspot, info, debug	-> heru (192.168.10.170): logged out: user request
Nov/21/2021 21:21:29	memory	hotspot, info, debug	test (192.168.10.170): trying to log in by http-chap
Nov/21/2021 21:21:29	disk	hotspot, info, debug	-> test (192.168.10.170): trying to log in by http-chap
Nov/21/2021 21:21:31	memory	hotspot, info, debug	test (192.168.10.170): login failed: invalid username or password
Nov/21/2021 21:21:31	disk	hotspot, info, debug	-> test (192.168.10.170): login failed: invalid username or password
Nov/21/2021 21:21:37	memory	system, info	changed scheduled script settings by heru
Nov/21/2021 21:21:44	memory	system, error, critical	login failure for user admin from 192.168.10.180 via ssh
Nov/21/2021 21:21:45	memory	system, error, critical	login failure for user admin from 192.168.10.180 via ssh
Nov/21/2021 21:21:47	memory	system, error, critical	login failure for user admin from 192.168.10.180 via ssh

Mikrotik Log Monitor

Today	
->: anam (192.168.10.139): login failed: invalid username or password	00:00
->: zvr pfnc (192.168.10.29): login failed: invalid username or password	07:12
->: anam (192.168.10.29): login failed: invalid username or password	07:13
->: pandu (192.168.10.173): login failed: no more sessions are allowed for user	20:37
->: pandu (192.168.10.173): login failed: no more sessions are allowed for user	20:57
->: zvr-nnur (192.168.10.166): login failed: invalid username or password	21:02
->: test (192.168.10.170): login failed: invalid username or password	21:21
login failure for user admin from 192.168.10.180 via ssh	21:22

Send certain mikrotik log to WhatsApp, so we can aware when critical thing happen

How WhatsApp Bot Work



How to Config

System > Script

LogMonitor

Raw

```
1  # This script is inspired by https://forum.mikrotik.com/viewtopic.php?t=151953
2
3  # BEGIN SETUP Edit Here
4  :local myserver ([/system identity get name])
5  :local scheduleName "LogMonitor"
6  :local key "{{KEY FROM WINBOT}}"
7  :local NoHP "628123xxxxxxx"
8  :local startBuf [:toarray [/log find message~"keyword 1" || message~"keyword 2" ]]
9  :local removeThese {"remove keyword"}
10
11 # END SETUP

88 if ([:len $output] > 0) do={
89     /system scheduler set [find name="$scheduleName"] comment=$currentTime
90     /tool fetch url="https://api.mikrotik-winbot.com/api/v1/whatsapp/send_message/$NoHP/$key/$message" keep-result=no;
91 }
```

<https://gist.github.com/therusetiawan/3b80f190a020b912bf99110627cc849d>

Example

System > Script

```
# This script is inspired by https://forum.mikrotik.com/viewtopic.php?t=151953

# BEGIN SETUP Edit Here
:local myserver ([/system identity get name])
:local scheduleName "LogMonitor"
:local key "secret-key!"
:local NoHP "62812345678"
:local startBuf [:toarray [/log find message~"login failed" || message~"via ssh" ]]
:local removeThese {"remove keyword"}

# END SETUP

# warn if schedule does not exist
if ([[:len [/system scheduler find name="$scheduleName"]] = 0]) do={
  /log warning "[LogMonitor] Alert : Schedule does not exist. Creating schedule ...."

  /system scheduler add name=$scheduleName interval=60s start-date=Jul/05/2019 start-time=startup on-event=LogMonitor

  /log warning "[LogMonitor] Alert : Schedule created ."
}

# get last time
:local lastTime [/system scheduler get [find name="$scheduleName"] comment]
```



Today

- >: anam (192.168.10.139): login failed: invalid username or password 00:00
- >: zvr pfnk (192.168.10.29): login failed: invalid username or password 07:12
- >: anam (192.168.10.29): login failed: invalid username or password 07:13
- >: pandu (192.168.10.173): login failed: no more sessions are allowed for user 20:37
- >: pandu (192.168.10.173): login failed: no more sessions are allowed for user 20:57
- >: zvr-nnur (192.168.10.166): login failed: invalid username or password 21:02
- >: test (192.168.10.170): login failed: invalid username or password 21:21
- login failure for user admin from 192.168.10.180 via ssh 21:22



Demo