Ubiquiti several Hard-coded credential Vulnerability-2

Affected firmware and version

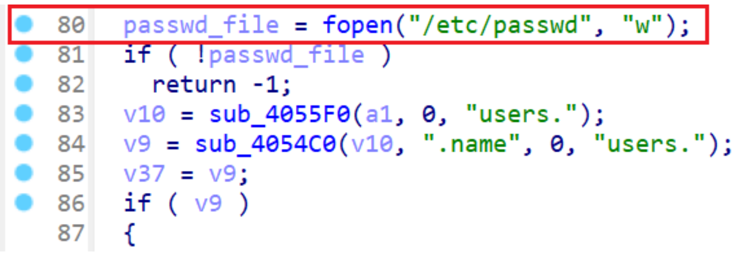
- GigaBeam,v1.4.2
  - GBE.v1.4.2.a96cd2e9.230330.1133.bin
  - URL: https://fw-download.ubnt.com/data/unifi-firmware/4c9a-UBB-1.0.7-9ff2dcefaa25471298e709680726544a.bin
- TI board,v6.3.11
  - Version: TI.v6.3.11.33396.230425.1547.bin
  - URL: https://dl.ubnt.com/firmwares/XN-fw/v6.3.11/TI.v6.3.11.33396.230425.1547.bin
- XM board,v3.6.11
  - XM.v6.3.11.33396.230425.1742.bin
  - URL: https://dl.ubnt.com/firmwares/XN-fw/v6.3.11/XM.v6.3.11.33396.230425.1742.bin
- EdgePower,v1.9.0
  - EP.v1.9.0.a67ced.210524.1407.bin
  - https://dl.ubnt.com/firmwares/edgemax/EdgePower/v1.9.0/EP.v1.9.0.a67ced.210524.1407.bin
- XC board,v8.7.0
  - XC.v8.7.11.42152.200203.1256.bin
  - https://dl.ubnt.com/firmwares/XC-fw/v8.7.11/XC.v8.7.11.46972.220614.0419.bin
- TI board,v6.3.6
  - TI.v6.3.6.33330.210818.1900.bin
  - https://dl.ubnt.com/firmwares/XN-fw/v6.3.6/TI.v6.3.6.33330.210818.1900.bin
- 2WA board,v8.7.4
  - 2WA.v8.7.4.45112.210415.1103.bin
  - https://dl.ubnt.com/firmwares/XC-fw/v8.7.4/2WA.v8.7.4.45112.210415.1103.bin
- 2XC board,v8.7.8
  - 2XC board,v8.7.8
  - https://dl.ubnt.com/firmwares/XC-fw/v8.7.8/2XC.v8.7.8.46705.220201.1820.bin

Description

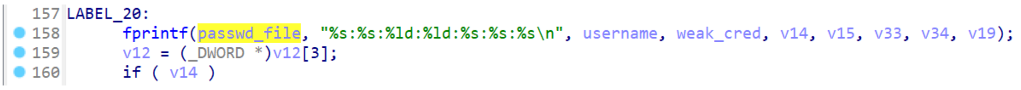
Several Ubiquiti firmware contains Use of Weak Credential vulnerability. The root credential is embedded in binary `ubntbox`. During firmware startup, the following hard-coded credential will be written into `etc/passwd`.

In the function at address `0x40DD60`, weak credential has been written into the `etc/passwd` file of the device.

In the following code, The below line 80 opens `/etc/passwd`, then line 157 writes content into `/etc/passwd`



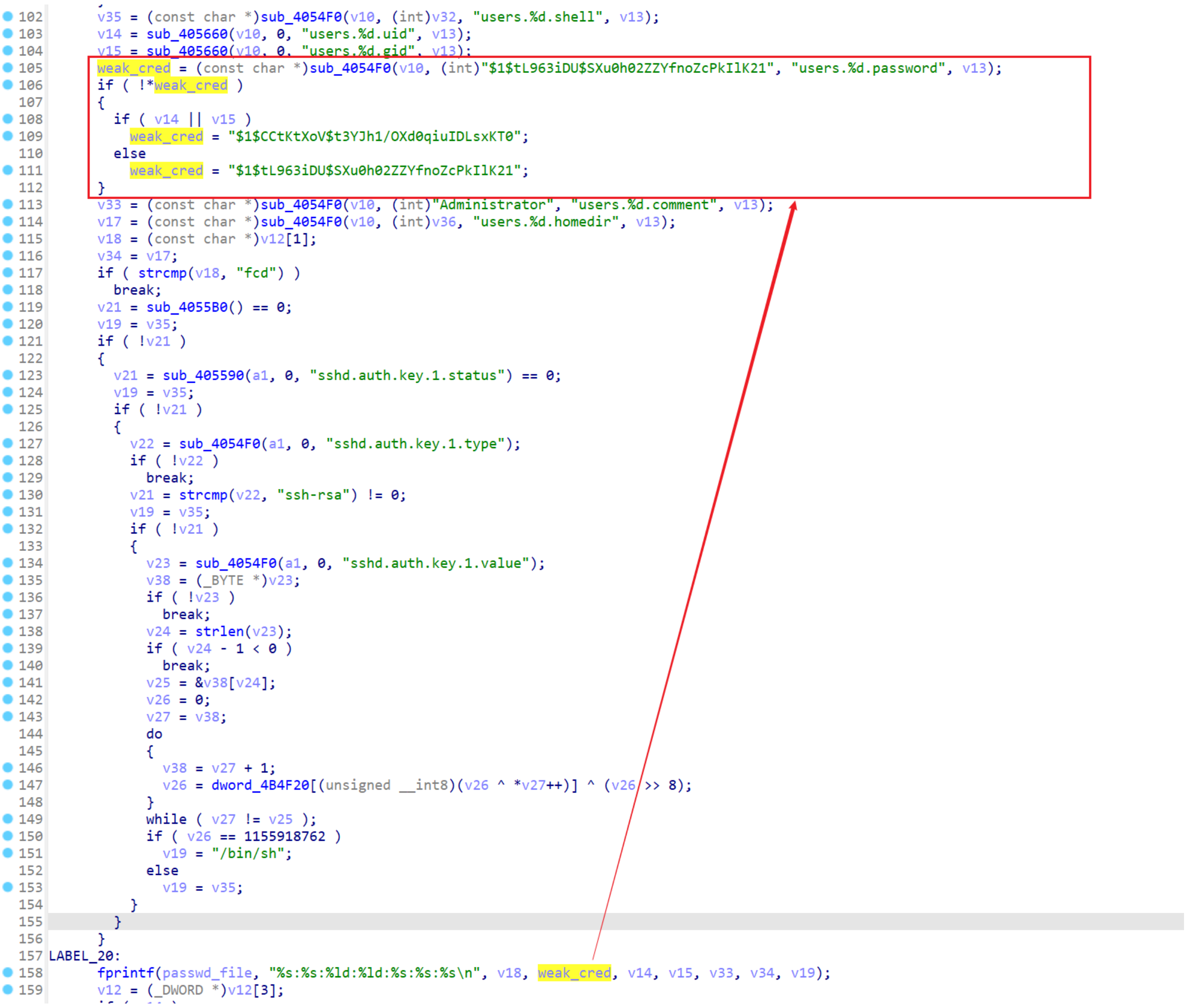
```
80   passwd_file = fopen("/etc/passwd", "w");
81   if ( !passwd_file )
82     return -1;
83   v10 = sub_4055F0(a1, 0, "users.");
84   v9 = sub_4054C0(v10, ".name", 0, "users.");
85   v37 = v9;
86   if ( v9 )
87   {
```



```
157 LABEL_20:
158       fprintf(passwd_file, "%s:%s:%ld:%ld:%s:%s:%s\n", username, weak_cred, v14, v15, v33, v34, v19);
159       v12 = (_DWORD *)v12[3];
160       if ( v14 )
```

Upon inspecting the content of the file, we can see that the default is '$1$tL963iDU$SXu0h02ZZYfnoZcPkIlK21' with '$1$' indicating the hash algorithm, the salt 'tL963iDU' and hash 'SXu0h02ZZYfnoZcPkIlK21'. There are totally two weak credential that will be written into the `/etc/passwd`, depends on the configuration of the device.

> $1$CCtKtXoV$t3YJh1/OXd0qiuIDLsxKT0
>
> $1$tL963iDU$SXu0h02ZZYfnoZcPkIlK21



```
102       v35 = (const char *)sub_4054F0(v10, (int)v32, "users.%d.shell", v13);
103       v14 = sub_405660(v10, 0, "users.%d.uid", v13);
104       v15 = sub_405660(v10, 0, "users.%d.gid", v13);
105       weak_cred = (const char *)sub_4054F0(v10, (int)"$1$tL963iDU$SXu0h02ZZYfnoZcPkIlK21", "users.%d.password", v13);
106       if ( !*weak_cred )
107       {
108         if ( v14 || v15 )
109           weak_cred = "$1$CCtKtXoV$t3YJh1/OXd0qiuIDLsxKT0";
110         else
111           weak_cred = "$1$tL963iDU$SXu0h02ZZYfnoZcPkIlK21";
112       }
113       v33 = (const char *)sub_4054F0(v10, (int)"Administrator", "users.%d.comment", v13);
114       v17 = (const char *)sub_4054F0(v10, (int)v36, "users.%d.homedir", v13);
115       v18 = (const char *)v12[1];
116       v34 = v17;
117       if ( strcmp(v18, "fcd") )
118         break;
119       v21 = sub_4055B0() == 0;
120       v19 = v35;
121       if ( !v21 )
122       {
123         v21 = sub_405590(a1, 0, "sshd.auth.key.1.status") == 0;
124         v19 = v35;
125         if ( !v21 )
126         {
127           v22 = sub_4054F0(a1, 0, "sshd.auth.key.1.type");
128           if ( !v22 )
129             break;
130           v21 = strcmp(v22, "ssh-rsa") != 0;
131           v19 = v35;
132           if ( !v21 )
133           {
134             v23 = sub_4054F0(a1, 0, "sshd.auth.key.1.value");
135             v38 = (_BYTE *)v23;
136             if ( !v23 )
137               break;
138             v24 = strlen(v23);
139             if ( v24 - 1 < 0 )
140               break;
141             v25 = &v38[v24];
142             v26 = 0;
143             v27 = v38;
144             do
145             {
146               v38 = v27 + 1;
147               v26 = dword_4B4F20[(unsigned __int8)(v26 ^ *v27++)] ^ (v26 >> 8);
148             }
149             while ( v27 != v25 );
150             if ( v26 == 1155918762 )
151               v19 = "/bin/sh";
152             else
153               v19 = v35;
154           }
155         }
156       }
157 LABEL_20:
158       fprintf(passwd_file, "%s:%s:%ld:%ld:%s:%s:%s\n", v18, weak_cred, v14, v15, v33, v34, v19);
159       v12 = (_DWORD *)v12[3];
```

Malicious attacker can reverse engineer the firmware and decrypt and gain the credential to log into the firmware.

Security Compliance

According to the **NIST SP 800-63B** Digital Identity Guidelines, predictable or static passwords (even if hashed) are not allowed for initial user authentication.

https://pages.nist.gov/800-63-3/sp800-63b.html

> Memorized secrets that are randomly chosen by the CSP (e.g., at enrollment) or by the verifier (e.g., when a user requests a new PIN) SHALL be at least 6 characters in length and SHALL be generated using an approved random bit generator [SP 800-90Ar1].