Who Controls the Controllers?

Hacking Crestron IoT Automation Systems
Who am I?

• Offensive Security Research 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

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

• “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
Deployment

• Massachusetts Bay Transit Authority

• Chicago Police Department

• American Water Corporate Headquarters
Deployment

Building a Reliable, Secure Voting System with Crestron Technology

Richmond, Virginia

Problem: Because voting is the bedrock of American democracy, any electronic voting system must be reliable, always available, easy to use, and extremely secure. For the Senate of Virginia, these were the key criteria for a new voting system to register and tabulate votes, control the order of business, call votes, recognize speakers, and summon members and pages. Decision Process: The Senate wanted a system that would give it better control over its technology. According to Jonathan Palmore, Senior Assistant Clerk, Technology, for the Senate of Virginia, “We really wanted complete control over the legislative mechanism, and we felt comfortable developing the application ourselves,” recalls Palmore. “The one thing we needed help with was the physical layer of voting—the part where our members would press a button, ‘yes’ or ‘no.’”

Related Models:

XPANEL  CP3  TS-1542-B-S  TSW-760-B-S

Deployment

http://hughsaudiovideo.com/hospitality_showcase.pdf
Products

• 3-Series controllers
  – CP3, MC3, PRO3
  – DIN rail
• Touch screens
  – TSx
  – TPCS, TPMC
  – “One in every room” type deployments
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
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
  - Product
  - Firmware version
  - Build date
Discovery

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

results from 2018/06/11
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...
Anonymous Admin on CTP Console
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> whoami
whoami   User
         Anonymous User
         Access Level
         Administrator
```

```
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 0x07900002  6 1368364/1296876
CRESLOG.exe         0x079B0006  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 Certificate Store
REM Certificate_Store Certificate_Name Certificate_UID - Remove Specified Certificate From Specified Certificate Store
LIST Certificate_Store - List All Certificates In Specified Certificate Store
VIEW Certificate_Store Certificate_Name Certificate_UID - View Details Of Specified Certificate In Specified Certificate Store
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?)

<table>
<thead>
<tr>
<th>Device</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>EEPROM-AT24C128N</td>
</tr>
<tr>
<td>01</td>
<td>VIDEO_DECODER-CH7026</td>
</tr>
<tr>
<td>02</td>
<td>RTC-M41T60</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Device</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>EEPROM-AT24C128N</td>
</tr>
<tr>
<td>01</td>
<td>VIDEO_DECODER-CH7026</td>
</tr>
<tr>
<td>02</td>
<td>RTC-M41T60</td>
</tr>
</tbody>
</table>

MC3>writei2c?
writei2c WRITEI2C [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

<table>
<thead>
<tr>
<th>Device</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>EEPROM-AT24C128N</td>
</tr>
<tr>
<td>01</td>
<td>VIDEO_DECODER-CH7026</td>
</tr>
<tr>
<td>02</td>
<td>RTC-M41T60</td>
</tr>
</tbody>
</table>
Hidden CTP Functionality

- Browser remote control: browseropen/browserclose (Android)
Hidden CTP Functionality

• UI interaction: fakekey/faketouch (Android)
Hidden CTP Functionality

• Record audio via microphone: recwave (Android)

```plaintext
TSW-760>recwave ?
RECWAVE [name] [length]
name - Name of WAV file.
length - length of recording in seconds.
```
A Few 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
  - But also adduser, cd, copyfile, delete, dir, fgetfile, fputfile, isdir, makedir, movefile, removedir, routeadd, routed delete, udir, updatepassword, wifipskpassword, wifissid, wifiwephexpassword, wifiweppassword, and more...
Cmd Inj Vulns on Android Platform

• Commands implemented programatically on WinCE platform
• Just punted to shell on Android
• Most were simple to exploit
  – EX: isdir `cmd`
Cmd Inj Vulns on Android Platform

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

    v2 = a1;
    v3 = a2;

    __printf_chk(&v5, 0, 1024, "cd %s & pwd | grep %s", a1, a2);
    printf("cd %s & pwd | grep %s\n", v2, v3);

    return system(&v5);
}
```
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
Round 2?

- Kept finding more vulns while root causing others
- Had to cut myself off due to time constraints
- Pretty positive there is more to find
I Want More!

• Significant amount of control by default
• Can escape CTP sandbox on Android using vulns
• But what about WinCE?...What about a more “legit” escape on Android?
SUPER SECRET BONUS 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?
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

• Github
  – https://github.com/headlesszeke
Thank You