PROCESS CONTROL SYSTEMS FORUM

SCADA Vulnerability Discovery &Disclosure

a case study

Matthew Franz
franz@digitalbond.com

Collaborating to Advance Control System Security

Vulnerability Session Agenda

- **30 Minute Case Study (this talk)**
- **Panel Discussion (30 - 45 min)**
- **Panel/Audience Discussion (15 - 30 min)**
- **Workshops Tomorrow**
  - Vendor & Researcher Expectations
  - Guidelines for Coordination Centers

Presentation Topics

- **Introduction & Background**
  - *Objectives*
  - *Vulnerability Landscape 2006*
  - *Biases & Caveats*
  - *The Cast*
  - *The Lifecycle of Security Flaw*
  - *Vulnerability Disclosure 101*
- **Six Months and 3+ SCADA Bugs Later**
- **US-CERT Control System Vulnerability Handling Process**
- **Lessons Learned**

"Case Study" Objectives

- **Provide a sanitized view of the SCADA vulnerabilities Digital Bond reported to US-CERT in order to:**
  - Introduce the control system community to key concepts/issues/questions in vulnerability disclosure
  - Illustrate the process developed by US-CERT to handle control system vulnerabilities
  - Provoke discussion and debate within the community
  - Provide a concrete frame of reference for the panel discussion and tomorrow's workshops

Up Front Caveats and Biases

- **Experience determines one's view of the problem**
  - I was a product security engineer in large network vendor
  - Currently a control system security researcher and consultant
- **It is very early in the game**
  - "Too few" bugs and almost zero knowledge of operational vulnerability exploitation
  - Limited number of impacted vendors
- **Digital Bond was not aggressive in dealing with vendors or US-CERT -- *no threats to go public!***

Vulnerability Landscape: 2006

- **On the software security front we've seen progress over the last 5 years**
  - Microsoft and even some SCADA vendors have stepped up
  - Increasing number of security testing products on the market
- **Vuln research is a commercial activity; disclosure is intimately tied to security products & services**
  - Vulnerability Sharing Clubs (3COM, iDefense)
  - The year of 0-days (WMF, Word, SCADA?)
- **SCADA implemention vulnerabilities remain a "dirty little secret" in spite of all the FUD**
  - Focus (perhaps rightly so) on OS vulns
  - Product "not designed to be..." mentality

Disclosure: The Cast

- Researchers (aka “finders” or “reporters”) identify hardware or software flaws in a product or application
- Vendors respond to reported vulnerabilities in a variety of different ways
- Coordination Centers work with vendors and researchers to bring the vuln to resolution
- End Users are not directly involved until fix unless the vulnerability was discovered on their systems

Milestones in the Vulnerability Lifecycle

- Introduction by a vendor when deployed by a user
- Discovery by a researcher or vendor
- Reported to a vendor or coordination center
- Confirmed by the vendor to researcher or coordination center
- Fix released by vendor or a third party
- Disclosure by vendor or coordination center or researcher
- Deployment by user after testing

WHAT IS LEFT OUT?

Presentation Topics

- **Introduction & Background**
- **Six Months and 3+ SCADA Bugs Later**
  - *The Bugs*
  - *Disclosure Timeline Summary*
  - *Vendor and Code Paths*
  - *Responsiveness & Communication Issues*
- **US-CERT Control System Vulnerability Handling Process**
- **Conclusions**

The Security Bugs

For the purpose of this talk, the details of the vulnerabilities/exploits/impact are irrelevant, but...

- We submitted three vulnerabilities to US-CERT (one found during a client engagement, the others discovered during other research projects)
- Most (if not all the bugs) are failures to handle invalid formatted messages and are “low skill” discoveries:
  - Accidently found with scanners
  - Simple “ISIC-style” fuzzing
- Many more not submitted over the last months/years

Bug Timelines Summarized

	Discover (Day 0)	Bug 1	Bug 2	Bug 3	Bug 4*	Median
V	Report	17,57,90	55,96,105	12	2	59
C		69,75,92,104	122	31		72
	Confirm	112 (C)	156 (C)	23, 30 (U)	9	90
				85 (F)		
	Fix	108	176	N/A	9	
	Disclose	187	176	N/A	?	

U - User
V - Vendor
C - CERT
F - Finder

* Not submitted to US-CERT

All Our Bugs were Multi-Vendor Issues

- **We saw two different types of vendors**

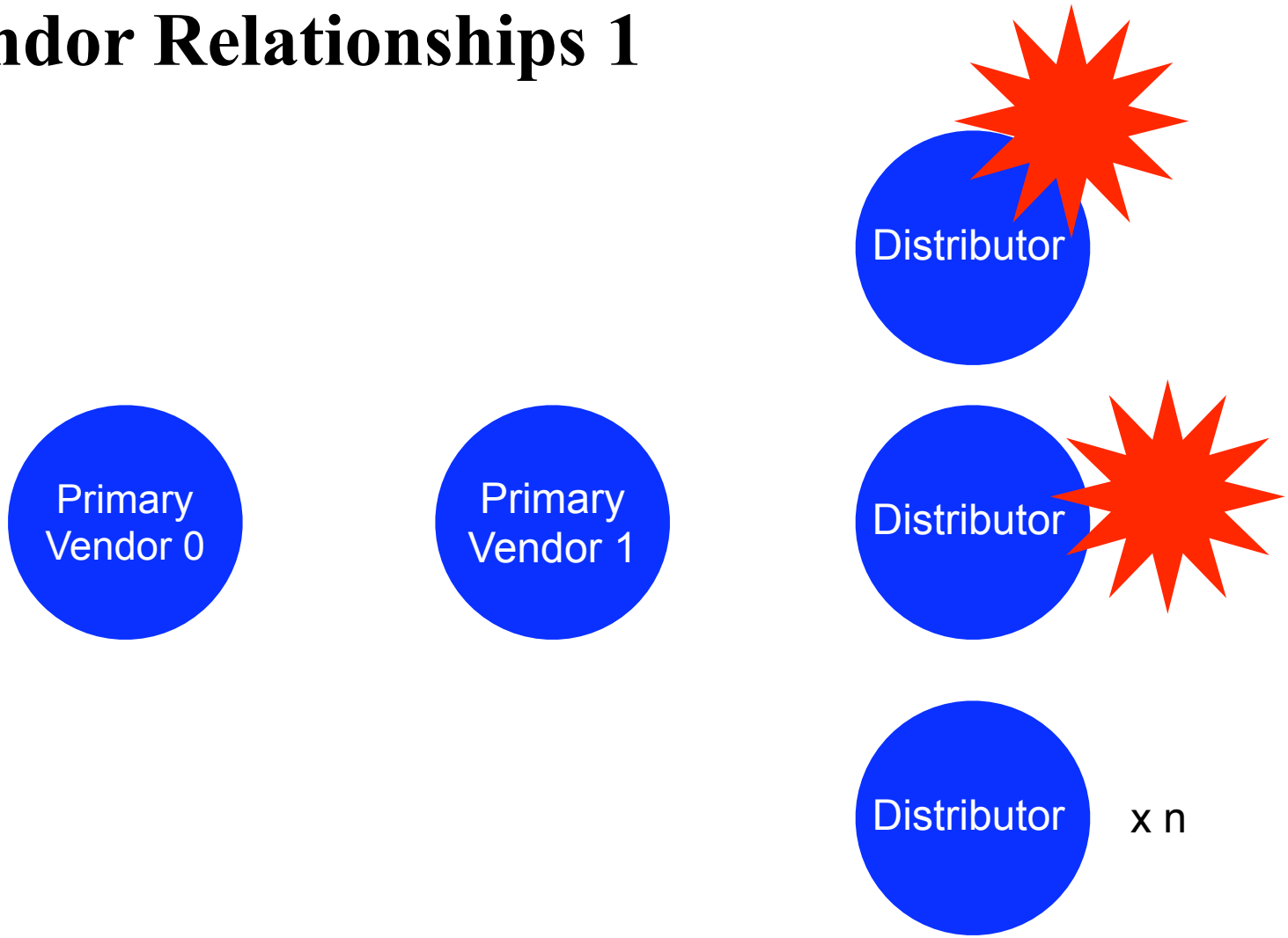
  Distributors - not “responsible” for impacted code and probably couldn’t make changes if they wanted to

  Primary - “own” the code module in question and probably can fix it (but they might have code dependencies, too)

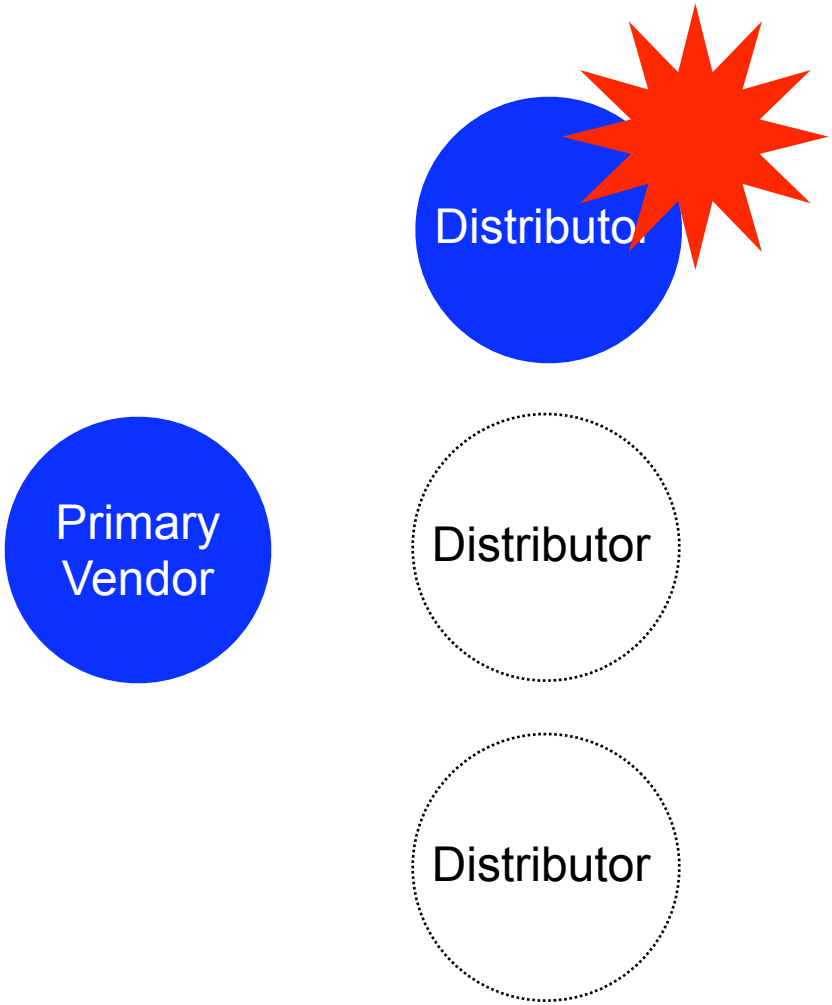
- **Both may sell (and support?) the vulnerable component, but more likely to be “visible” on the primary**
- **One primary vendor who did not appear to sell a product directly**

  We still don’t know which products use the vulnerable component

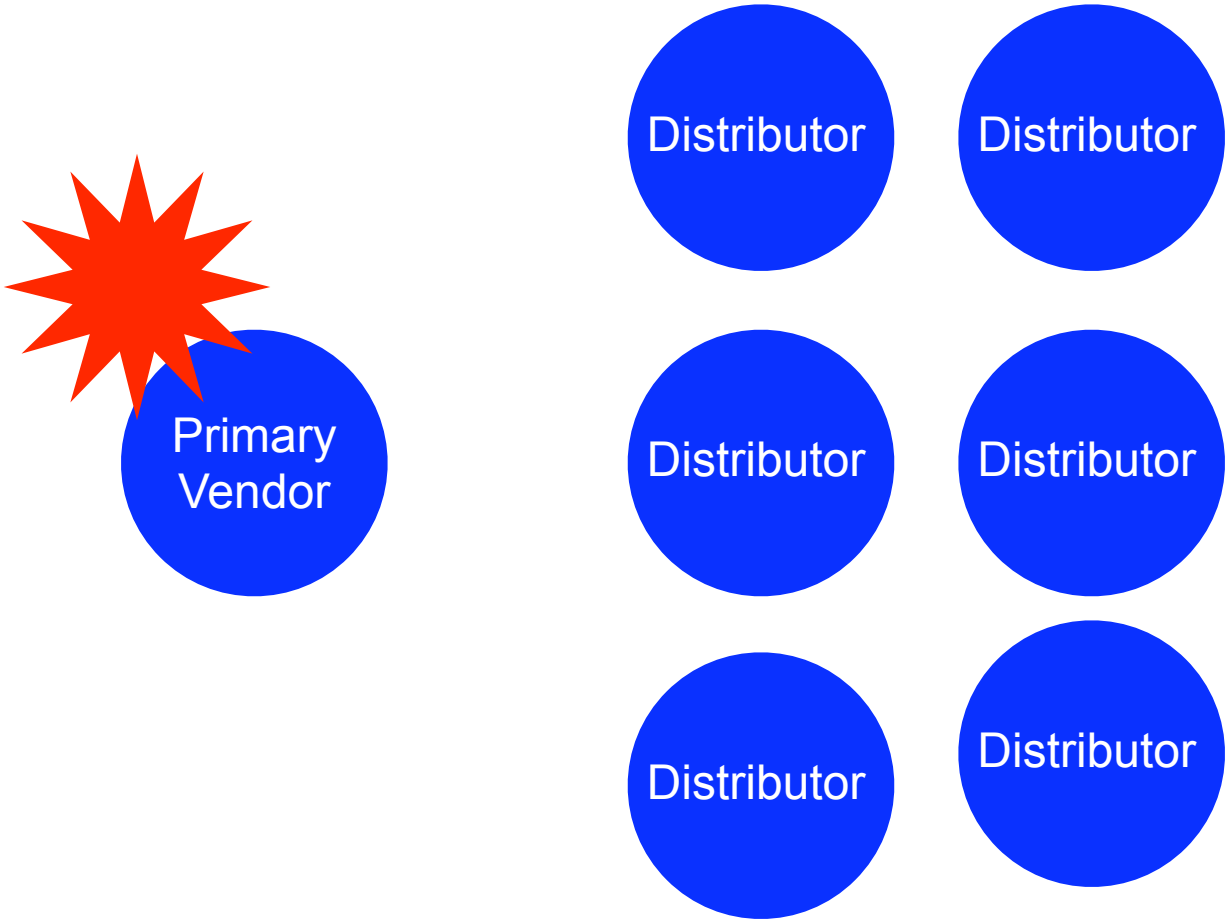
Vendor Relationships 1



Vendor Relationships 2



Vendor Relationships 3



The "Vendor Acknowledgement Differential"

Depending on who reports the problem results in different levels of acknowledgement from vendors

- Researcher - lucky to get acknowledgement of email communication, never received any written (meaning email) confirmation of flaws
- Customer - vendors provided formal written reponses (email or memo) but they tended to be "marketing answers"
- Coordination Center - vendors acknowledged vulnerabilities to CC's and provided details (type of vuln and code module) and *on average* were "interested in working together"

The Challenge of Secure Communication

- **What are the Requirements?**
  - Sensitive vulnerability/exploit information needs to be encrypted when sent over insecure networks or residing on servers
  - All parties need to be able to authenticate each other, but some may have “more to lose”
- **PGP/GPG is the *de facto* standard but is difficult to use**
  - It took several days for Digital Bond to get secure comms with US-CERT (to be able to read message) but no issues with CERT/CC
  - CERT/CC had a difficult time getting secure comms with vendors, in particular getting vendors to sign messages
  - Digital Bond used our “secure email” service which only met the confidentiality requirement

Communication and the "Perception of Responsiveness"

- **Although email is the most common means for communicating vulnerability information it is inadequate:**
  - Just because a vendor doesn't respond to emails doesn't mean then don't care about security, *right?!*
  - Extremely difficult to get correct contact information for vendors
- **Differing opinions on who bears the burden of reaching out**

Miscellaneous Observations

- **Previous dealings with vendor sometimes resulted in quicker communication, but did not increase the likelihood of confirmation or action**
- **Several vendors made detailed technical statements about the vulnerability (risk, mitigation, resolution) with few to no technical details about the vulnerability and without even reproducing the flaw**

  We deliberately did not provide all the information we had, because we wanted to see if they would ask for it.

Presentation Topics

- Introduction & Background
- Vulnerability Disclosure 101
- Six Months and 3+ SCADA Bugs Later
- US-CERT Control System Vulnerability Handling Process
  - *Reporting Vulnerablities*
  - *Organizational and Message Flow*
- Conclusions

Reporting Mechanisms

The US Department of Homeland Security (DHS), through the Control Systems Security Program (CSSP) of the National Cyber Security Division (NCSD) is *helping to secure our nation's critical infrastructure by identifying, analyzing, and reducing cyber risks associated with the control systems that govern these infrastructures.*

Reporting

The US-CERT Control System Security Program is a partnership between the Department of Homeland Security and public and private sectors. Established to work in partnership with the industries whose control systems comprise part of our nation's critical infrastructure, this system is used to report cyber-related incidents to US-CERT.

Report an Incident Report a Vulnerability

```
-----BEGIN PGP SIGNED MESSAGE-----

Version 1.0
October 1996
                        CERT(R) Coordination Center
                  Product Vulnerability Reporting Form

    If you know of a vulnerability in a product, please complete
    this form and return it to cert@cert.org.  We aren't able to
    acknowledge each report we receive; however, if we have additional
    questions, we will contact you for further information.

          We prefer that any vulnerability information you
          send to us be encrypted. We can support a shared DES
          key or PGP. Contact the CERT staff for more information.
          The CERT PGP public key is available in

                     http://www.cert.org/pgp/cert_pgp_key.asc

    Thanks, we appreciate your taking the time to report this
    vulnerability.



CONTACT INFORMATION
================================================================================
Let us know who you are:

 Name                    :
 E-mail                  :
 Phone / fax             :
 Affiliation and address:
```

TO: cert@cert.org
CC: soc@us-cert.gov

US-CERT
UNITED STATES COMPUTER EMERGENCY READINESS TEAM

Publications
Events
Other Resources
About Us

Sign up for email alerts.

DHS Threat Advisory
ELEVATED

Reporting

Report an Incident
Report Phishing
Report a Vulnerability

Welcome

Established in 2003 to protect the nation's Ir
defense against and responses to cyber atta

Learn more about us

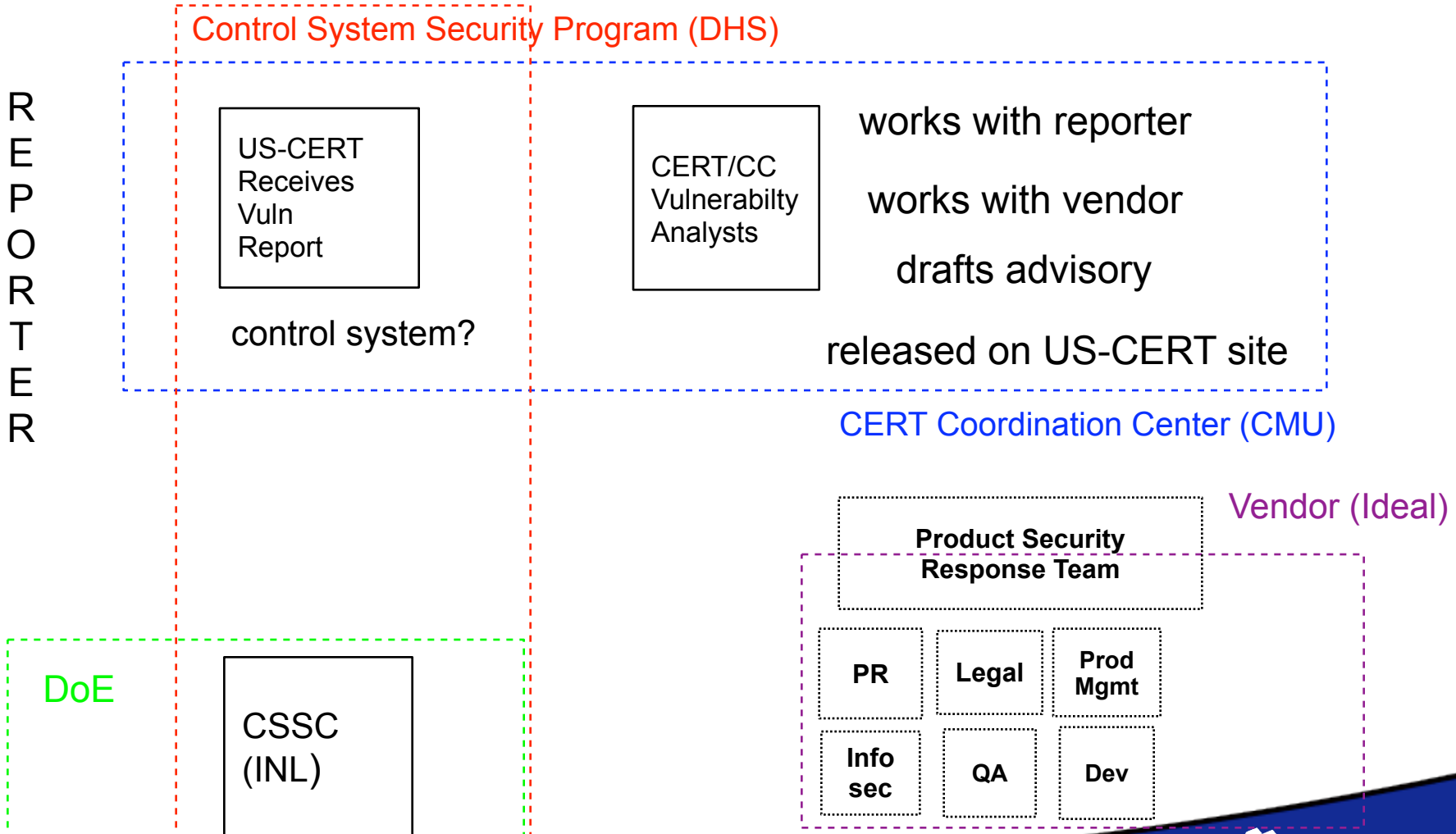
Technical Users

System administrators and computer professionals can review our technical security documents and services.

Government Users

State, local, and federal government users can access information tailored to their needs.

Opening up the Box



Presentation Topics

- Introduction & Background
- Vulnerability Disclosure 101
- Six Months and 3+ SCADA Bugs Later
- US-CERT Control System Vulnerability Handling Process
- Conclusions

My Conclusions from the Last 6 Months

- **Vendors are ill-equipped to handle vuln reports from independent security researchers**
  - Little to no security contact information on public sites
  - A *responsible* "black hat" would at least have to threaten to "go public" to get even implicit acknowledgement of a vuln
- **US-CERT & CERT/CC have created a process for handling SCADA vulns**
  - Communication with vendors was challenging, sometimes because vendors weren't aware who CERT/CC was
  - So far, the "Internet/IT" vulnerability handling process appears to work for SCADA, too

More Conclusions

- **Once you can get their attention, SCADA/EMS Vendors can and did release security fixes in a relatively short time period**
  - Would not have moved as “quickly” if a trusted third-party had not been notified
  - Researchers should consider simultaneous notification to vendors and coordination centers
- **Reproduction & confirmation of findings (whether the flaw or fix) remains problematic**
  - Access to flawed/fixed software
  - IP issues involving vulnerability testing tools
  - Who can provide independent verification to the end user?

Questions or Comments?

Write them down and turn them or:

Track me down here today/tomorrow

Email me at franz@digitalbond.com

SCADA Vulnerability Disclosure Panel Session

Do the case study findings suprise you?
Agree or Disagree?

Should disclosure practices of large IT vendors be held up as a model?

Are security bugs fundamentally different from other bugs?

Should end users have to pay for security fixes?

When (or why) is there a need for vulnerability coordination centers?

How can enough vulnerability information be communicated to make informed risk decisions without arming potential attackers?

What is the best way to notify all users of a vulnerable product, including those with lapsed service contracts?

Under what circumstances should anyone disclose a vulnerability if a fix is not available?

Was it appropriate for US-CERT to release an advisory for the recent ICCP Server Vulnerability?

If not, how else should the vulnerability have been handled?

Is there a role for standards organizations in vulnerability discovery, response, and disclosure?

If detailed vulnerability/exploit information is not readily available, how can patches be independently verified? How can security vendors develop IPS signatures to protect systems that cannot be patched?