27th annual INCOSE international symposium

Adelaide, Australia

July 15 - 20, 2017

A case for using High Throughput Testing (HTT) Techniques in Cybersecurity

Test Strategy to detect Industrial Control Systems' common Cyber Weaknesses and Vulnerabilities

Author

Dr. Obaid Ur Rehman

Dr. Keith F. Joiner

www.incose.org/symp2017

Capability Systems Centre

UNSW CANBERRA

Introduction

- ICSs such as SCADA systems, DCS & PLCs play a vital role in the operation and monitoring of a nation's critical infrastructure.
  - Nuclear reactor, Water treatment facility, and transport network etc.
- Attack on any of these activities may cause catastrophic & cascading effects which has a potential to reduce the defence capability of a nation.

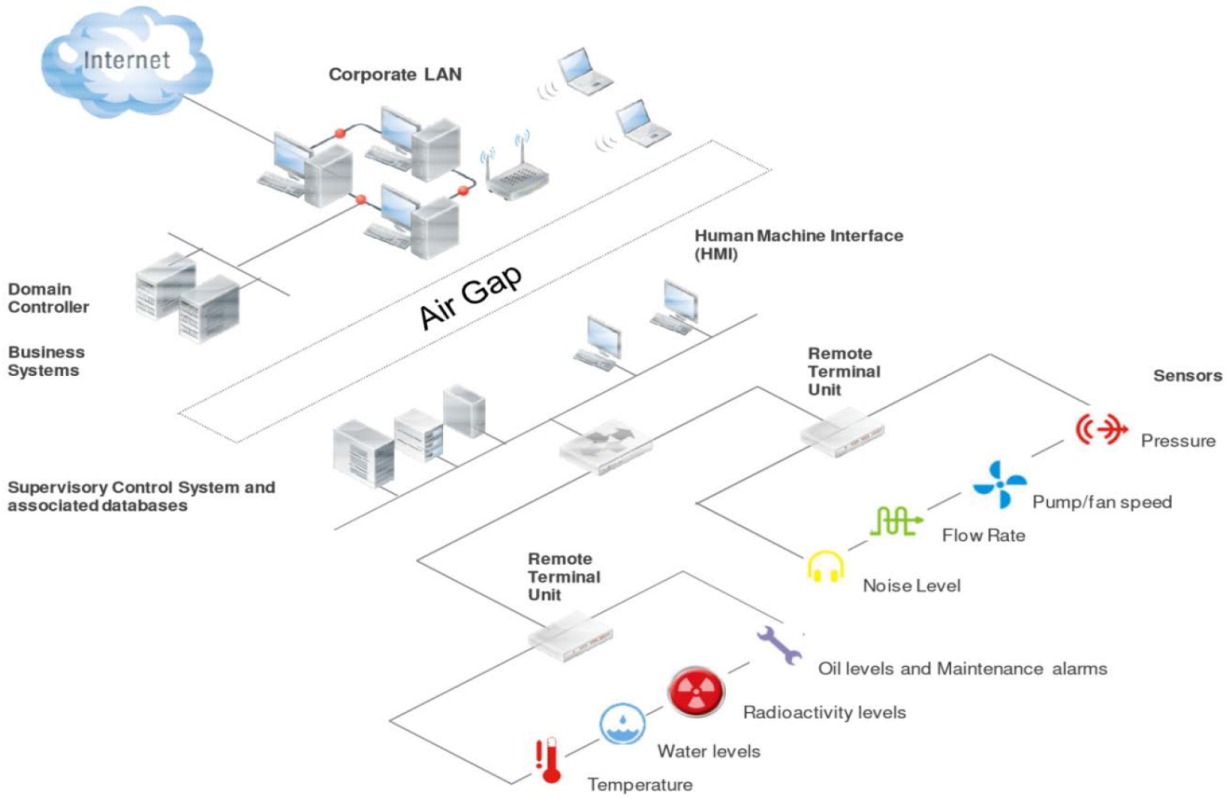
Image from www.pixabay.com

SCADA System

- Industrial control systems use a combination of hardware and software components to accomplish the control and monitoring of a system and they are usually connected to a network.

- The ICSs which are controlled through SCADA are located on several different geographical locations and connected to the SCADA server through a network.

- The communication protocols used in the system is not designed for security.

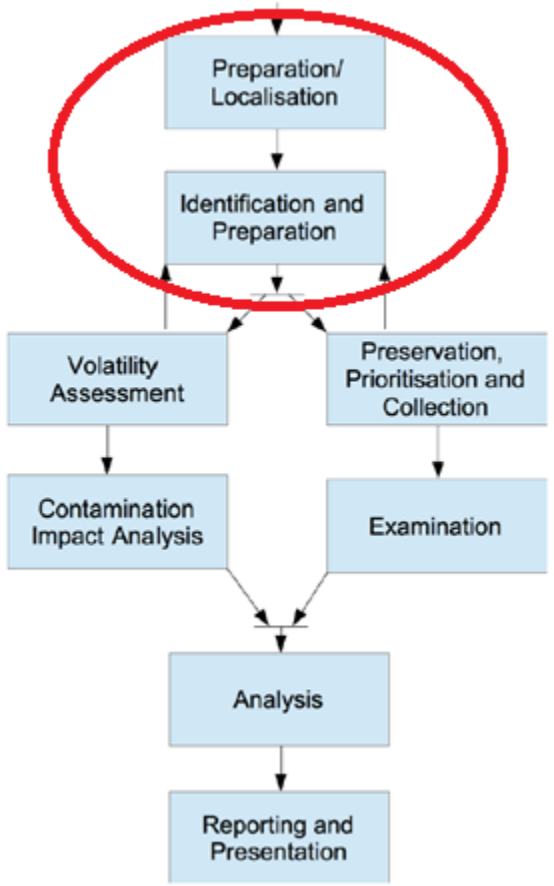


SCADA System

- SCADA systems cannot be updated/patched regularly due to operational reasons.
- SCADA components such as RTUs & PLCs
  - designed for their functionality &
  - do not have any authentication process available in they system.
- Since, SCADA is now a days connected to the corporate network which connects to the internet, this made SCADA system an easier target for cyber attack.
- Over 1 million SCADA systems are connected to the internet and this number is growing day by day.
- Challenge of defending more complex legacy platform systems like aircraft, vehicles and ships that until recently were relatively standalone from ICT systems

Aim and Scope

- The focus of this research is to
  - explore the ways to identify and prepare for any malicious attack on ICS/SCADA system.
  - find a structured technique for dealing with very high numbers of test permutations that arise when considering complex system architectures,
  - explore the efficacy of statistically rigorous methods such as high throughput testing (HTT)/ DOE methods considering high number of test cases in testing the cyber vulnerabilities of ICSs.



Phases of SCADA incident reporting and forensic process[1].

[1]J. Stirland et. al. Developing Cyber Forensic for SCADA Industrial Control Systems. Proc. of the Int. Conf. on Info. Security and Cyber Forensic, 2014

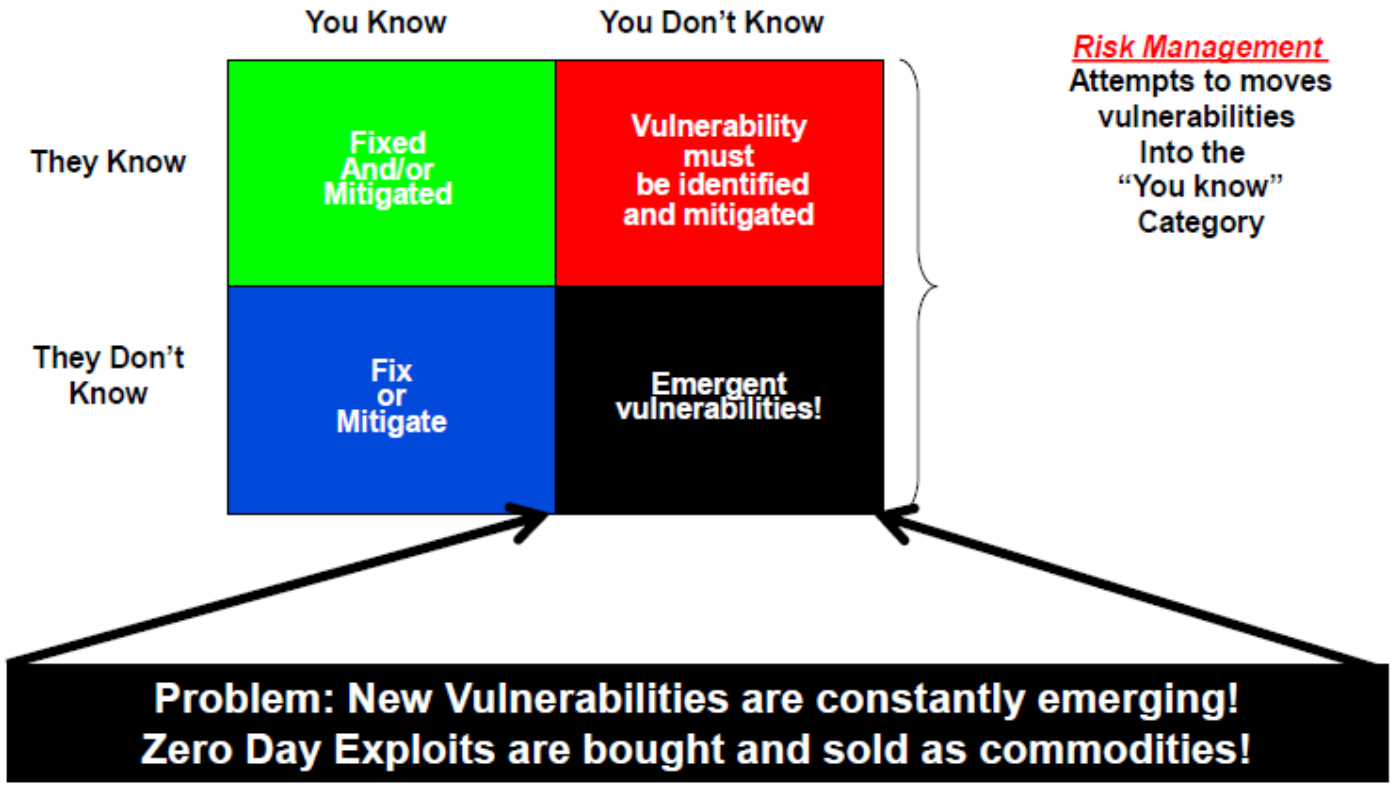
Outline

- Common threats on ICS/SCADA systems.
- Preparation to identify threats.
- Discuss Common Weakness Enumeration (CWE) and scoring system.
- Testing Strategy
- High Throughput Testing (HTT).

Common - Threat

- **On communication network layer**, i.e. gaining access to the communication protocol
- **On hardware**, i.e. changing configuration file on RTUs/PLCs.
- **On application**, i.e. using malicious software on the SCADA system to make system invisible from operator view etc.

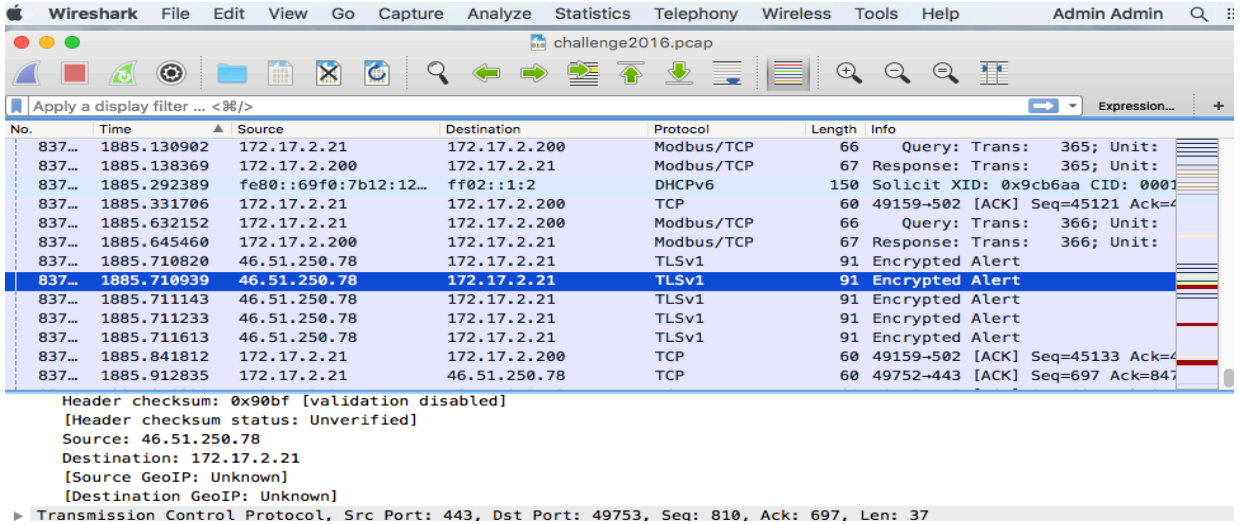
- If a SCADA system connected to the internet these attack can be carried out remotely.
- The attack on SCADA system are generally originated from IT systems and reach to the SCADA system through communication network.
- In order to avoid these attacks the first step is to prepare for any possible threat so that any malicious attack quickly identified and system is restored in quick time.

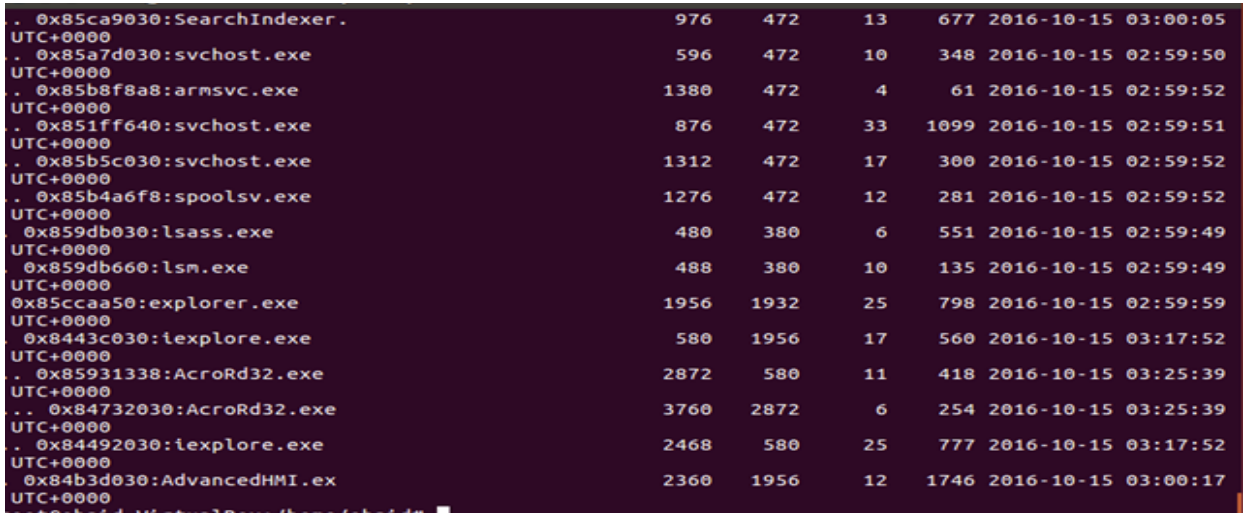


Picture from Christensen, P. (2015). *Introduction to Cybersecurity T&E.* Tutorial at 32nd International T&E Association Symposium, Washington, August.

System White-list

- In order to identify the system component, a white-list of all the software, protocols, ports and hardware which are used in the system should be prepared.
- The white-list provides a way to identify application and system components that are authorized to be present or active during any operation.





Common Weakness Enumeration (CWE)

- CWE is a community-developed dictionary of software weakness type available free for public[1].
- CWE provides a unified & measurable set of software weaknesses which can be used to better understand any threat to the system.
- CWE provides an opportunity to obtain information about software related vulnerabilities, its impact, the attack vector & suitable mitigation strategies.
- Although, CWE deals with the software weaknesses it is also applicable to the ICS because of extensive use of software & communication network.
- The CWE construct can effectively be used to design initial screening test strategies for ICS..

Picture from the 2013 U.S. Defense Science Board Report on *Resilient Military Systems and The Advanced Cyber Threat*.

[1]http://cwe.mitre.org/index.html

For each CWE following information is provided:

Ranking	The ranking of the weakness in the general list.
Score Summary	A summary of the individual ratings and scores that were given to this weakness, including Prevalence, Importance, and Adjusted Score.
CWE ID and name	CWE identifier and short name of the weakness
Supporting Information	Supplementary information about the weakness that may be useful for decision-makers to further prioritize the entries.
Discussion	Short, informal discussion of the nature of the weakness and its consequences. The discussion avoids digging too deeply into technical detail.
Prevention and Mitigations	Steps that developers can take to mitigate or eliminate the weakness. Developers may choose one or more of these mitigations to fit their own needs. Note that the effectiveness of these techniques vary, and multiple techniques may be combined for greater defense-in-depth.
Related CWEs	Other CWE entries that are related to the Top 25 weakness. Note: This list is illustrative, not comprehensive.
General Parent	One or more pointers to more general CWE entries, so you can see the breadth and depth of the problem.
Related Attack Patterns	CAPEC entries for attacks that may be successfully conducted against the weakness. Note: the list is not necessarily complete.
Other pointers	Links to more details including source code examples that demonstrate the weakness, methods for detection, etc.

Table is taken from http://cwe.mitre.org/

- Each CWE provides target area of particular weakness & following information on system vulnerabilities:
  - Weakness prevalence
  - Remediation cost
  - Attack frequency
  - Consequences
  - Ease of detection
  - Attacker awareness

Table is taken from http://cwe.mitre.org/

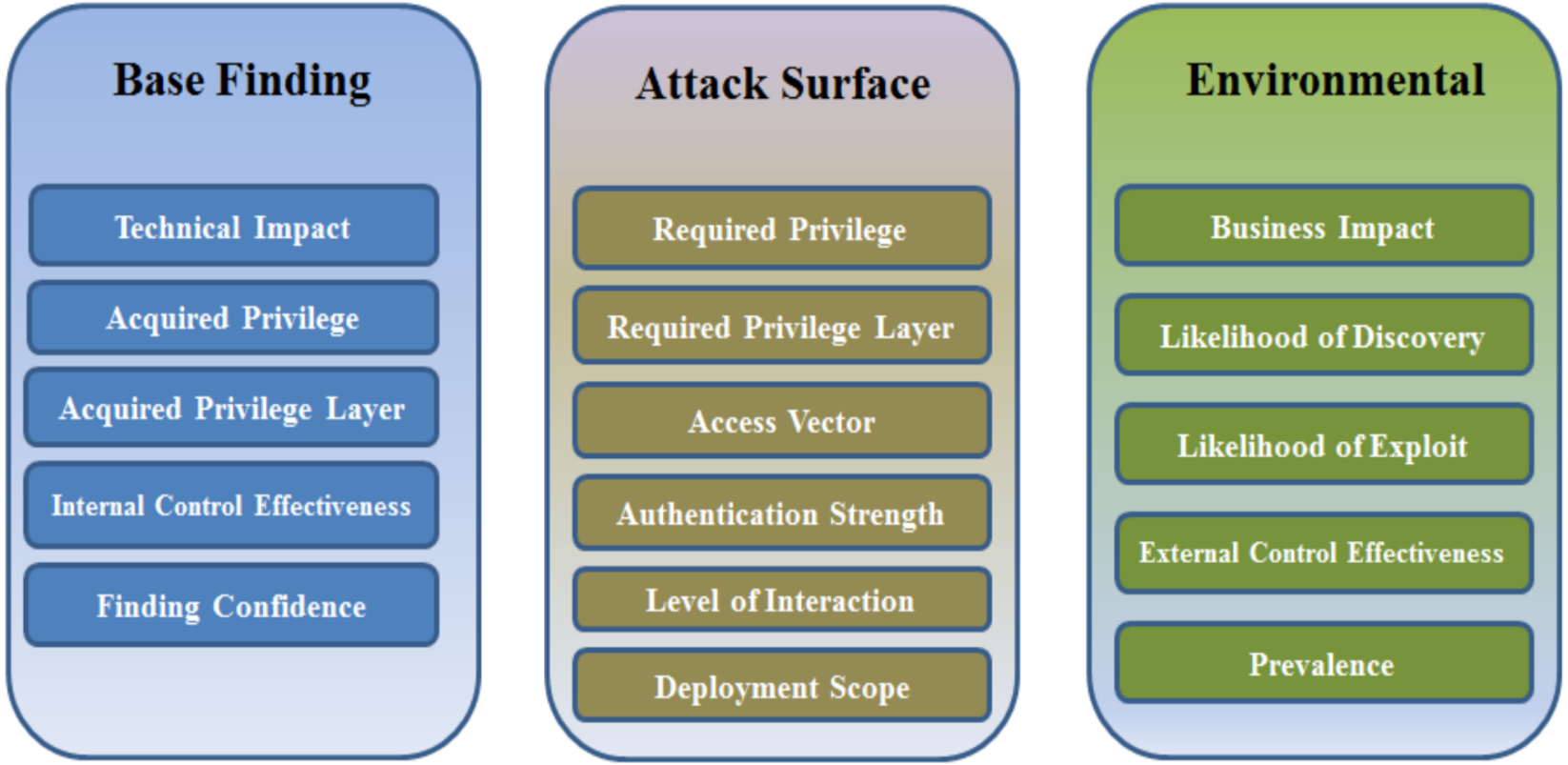
1	CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

Summary			
Weakness Prevalence	High	Consequences	Data loss, Security bypass
Remediation Cost	Low	Ease of Detection	Easy
Attack Frequency	Often	Attacker Awareness	High

- it is necessary as part of the test preparation to devise a mechanism to quantify the threat level based on the impact on the system under test.

Common Weakness Scoring System (CWSS)

- Mechanism to obtain a single score value for the impact.
- Subdivided into three metric groups shown



Picture from http://cwe.mitre.org/

CWSS score calculated considering the impact of each factors affecting the metric group

Value	Code	Weight	Description
Critical	C	1.0	Complete control over the software being analyzed, to the point where operations cannot take place.
High	H	0.9	Significant control over the software being analyzed, or access to critical information can be obtained.
Medium	M	0.6	Moderate control over the software being analyzed, or access to moderately important information can be obtained.
Low	L	0.3	Minimal control over the software being analyzed, or only access to relatively unimportant information can be obtained.
None	N	0.0	There is no technical impact to the software being analyzed at all. In other words, this does not lead to a vulnerability.
Default	D	0.6	The Default weight is the median of the weights for Critical, High, Medium, Low, and None.
Unknown	UK	0.5	There is not enough information to provide a value for this factor. Further analysis may be necessary. In the future, a different value might be chosen, which could affect the score.
Not Applicable	NA	1.0	This factor is being intentionally ignored in the score calculation because it is not relevant to how the scorer prioritizes weaknesses. This factor might not be applicable in an environment with high assurance requirements; the user might want to investigate every weakness finding of interest, regardless of confidence.
Quantified	Q		This factor could be quantified with custom weights.

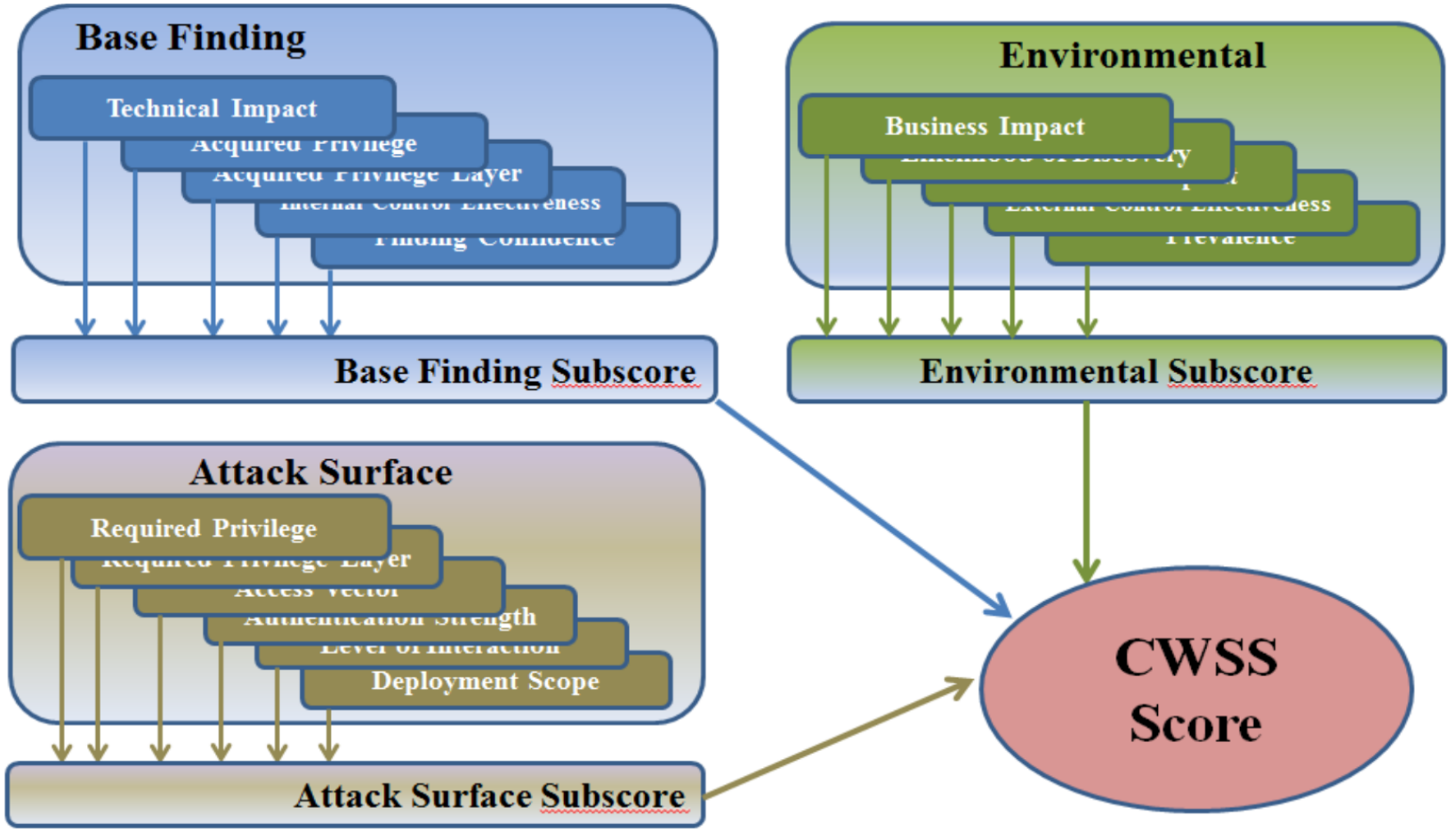
Table is taken from http://cwe.mitre.org/

All other factors have similar scoring system & thus can be quantified.

See http://cwe.mitre.org/cwss/cwss_v1.0.1.html.

Picture taken from http://cwe.mitre.org/

- Once scored, weight assigned to each factor according to entity's preferences
- Produces three sub scores for three group metrics
- Finally a single score is calculated as shown



Thus CWSS scores each weakness, prioritizes them & helps attribute impact of each CWE on any particular application or system.

Test Strategy

- There are a large number of CWE available and their factors some time overlap and produce the same impact on a system.
- If all CWE are taken into account then it may require significant time and resources for testing.
- Considering the information available under each CWE information we develop a screening process to prioritize mitigation strategies and develop deeper test strategies around each of these weaknesses.

CWE Screening

- Categorise CWE according to its impact to the system components
- E.g. take 18 CWE which are sorted under three high-level categories (2009 top 25).

1) Insecure interaction between components.

Rank	CWE ID	Name
[1]	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
[2]	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
[4]	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
[9]	CWE-434	Unrestricted Upload of File with Dangerous Type
[12]	CWE-352	Cross-Site Request Forgery (CSRF)
[22]	CWE-601	URL Redirection to Untrusted Site ('Open Redirect')

Table is taken from http://cwe.mitre.org/

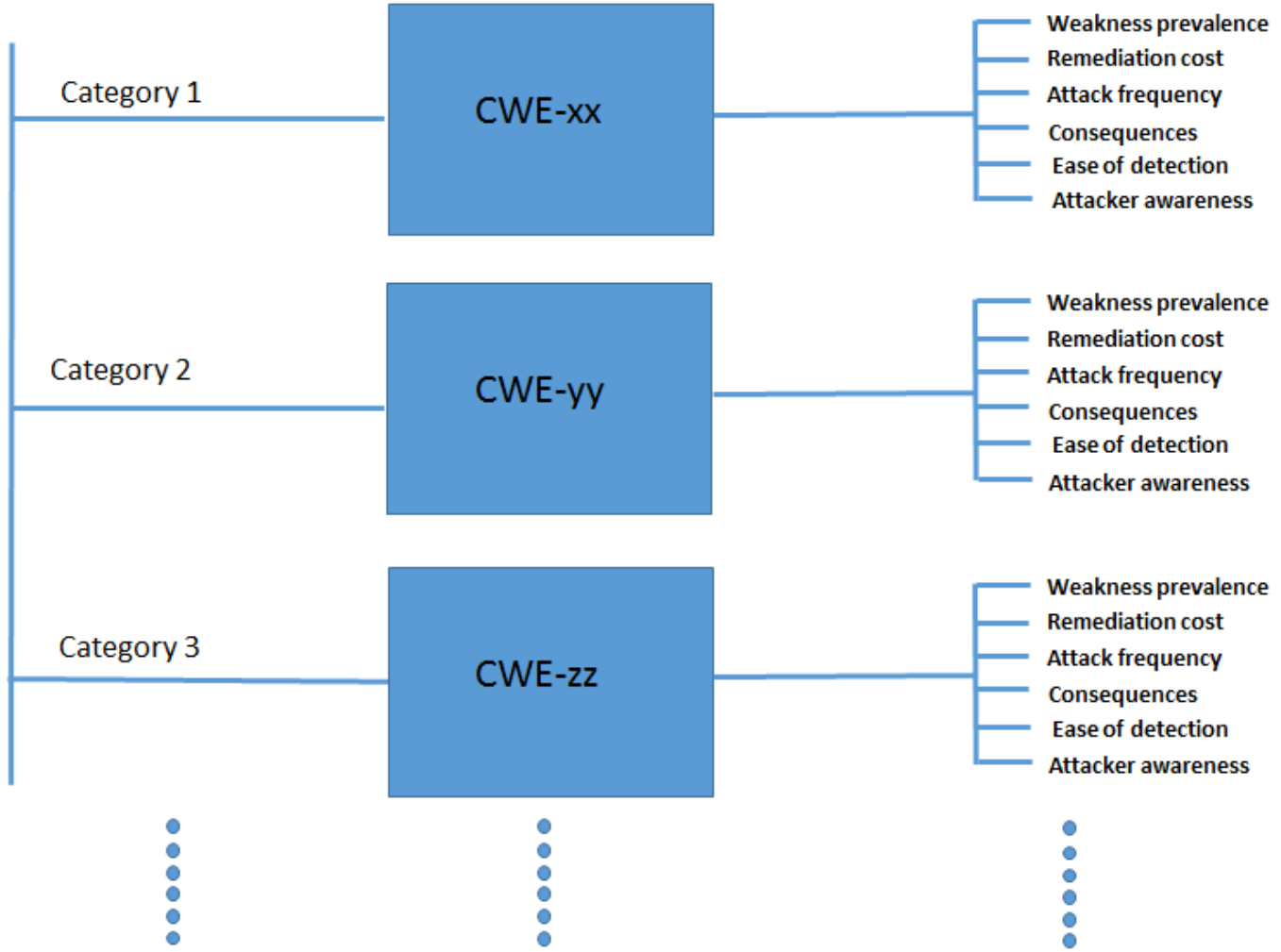
2) Risky Resource Management

Rank	CWE ID	Name
[3]	CWE-120	Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
[13]	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
[14]	CWE-494	Download of Code Without Integrity Check
[16]	CWE-829	Inclusion of Functionality from Untrusted Control Sphere
[18]	CWE-676	Use of Potentially Dangerous Function
[20]	CWE-131	Incorrect Calculation of Buffer Size
[23]	CWE-134	Uncontrolled Format String
[24]	CWE-190	Integer Overflow or Wraparound

3) Porous Defences

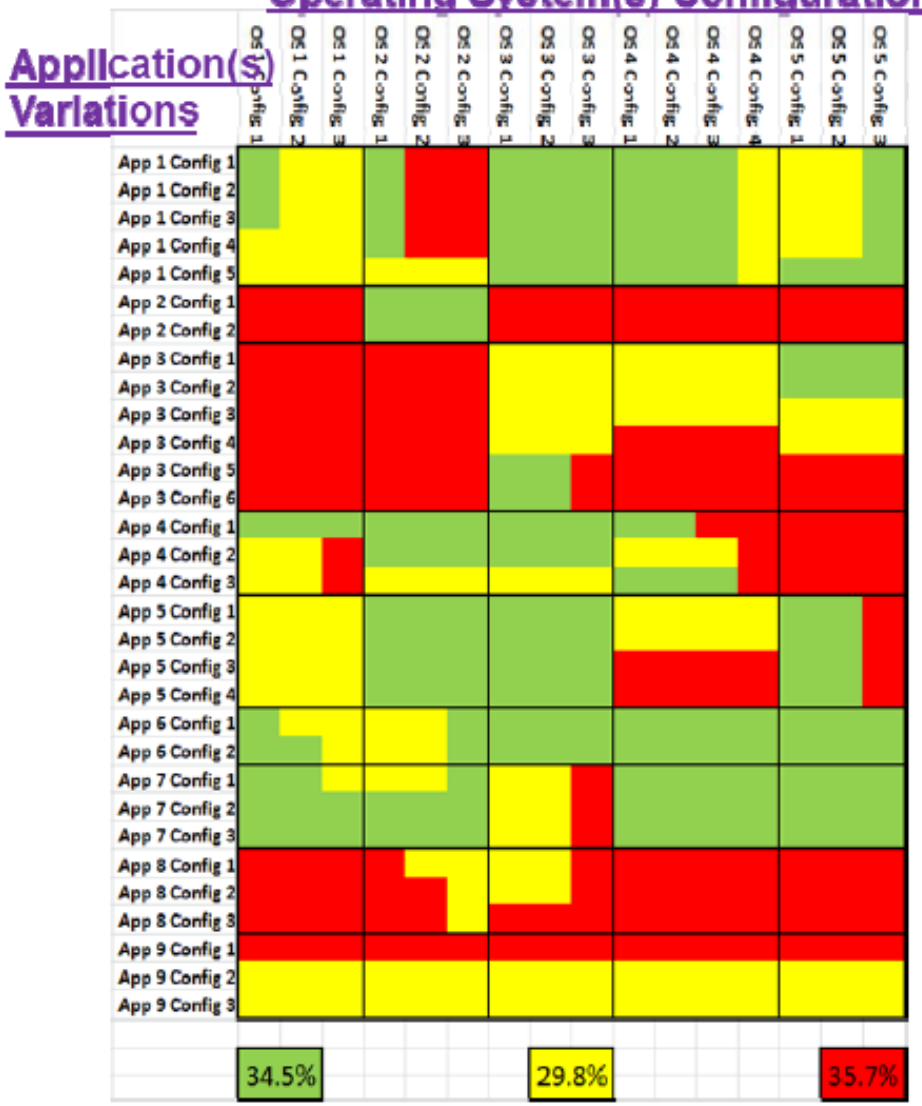
Rank	CWE ID	Name
[5]	CWE-306	Missing Authentication for Critical Function
[6]	CWE-862	Missing Authorization
[7]	CWE-798	Use of Hard-coded Credentials
[8]	CWE-311	Missing Encryption of Sensitive Data
[10]	CWE-807	Reliance on Untrusted Inputs in a Security Decision
[11]	CWE-250	Execution with Unnecessary Privileges
[15]	CWE-863	Incorrect Authorization
[17]	CWE-732	Incorrect Permission Assignment for Critical Resource
[19]	CWE-327	Use of a Broken or Risky Cryptographic Algorithm
[21]	CWE-307	Improper Restriction of Excessive Authentication Attempts
[25]	CWE-759	Use of a One-Way Hash without a Salt

CWE Category



Motivation to use HTT

- To propose developing a rigorous test methodology which helps to identify the weaknesses in the system using the information available in CWE.
- The test strategy obtained through this method will help to prioritize mitigation strategies & thus reduce the incident response time.
- HTT can screen CWE according to their impact on a particular system & use them to develop deeper test strategies around each of these weaknesses i.e. cooperative vulnerability & penetration testing by red & blue teams.



Picture from Troester (2015). National Cyber Range Overview. Presentation to ITEA Cybersecurity Workshop: Test and Evaluation to meet the Persistent Threat, Belcamp MD, February

High Throughput Testing (HTT)

- Using HTT we can determine “all pair” test coverage for combinations of many factors.
- It provides a screening solution to the problem of excessive test cases.
- HTT uses combinatorial mathematics, which is based on the complex optimization algorithms and heuristics, to reduce the total number of test cases to a minimum while ensuring a predetermined coverage level.
- The orthogonal “all pairs” test matrix can be determined using software packages such as “rdExpert”, “Pro-test” or “Praxis”.
- The result will help to focus in areas of identified weakness, which in turn will help choose the most appropriate mitigation strategies for each critical weakness and thus improve the cyber-incident response and overall cyber-resilience.

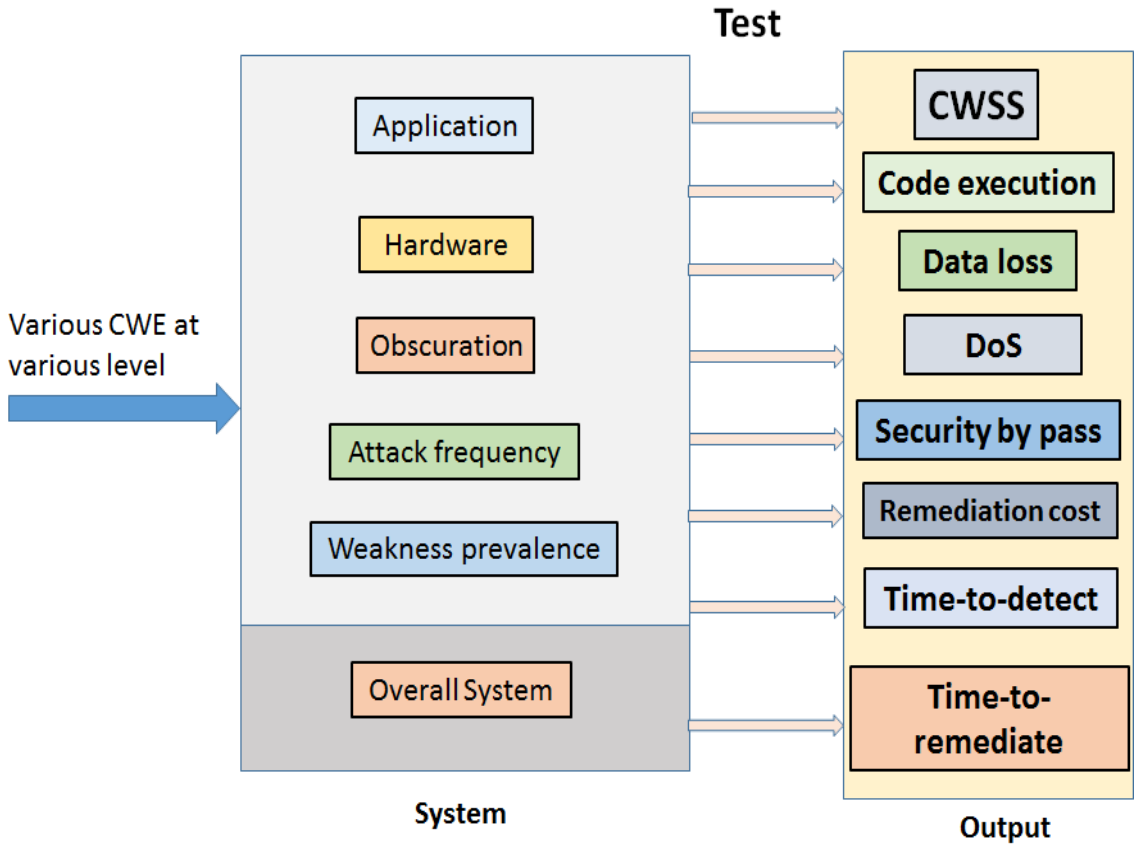
HTT Input factors

- For HTT, information provided in CWE should be extracted in a certain way.
- We need to extract input factors which can be analysed in the test based on the define output responses.
- Following table shows an examples of input factors at their specified levels.

Lbl	Factor Name	No. Lvls	Level 1	Level 2	Level 3	Level 4	Level 5
A	Application	2	Active	Not Active			
B	Hardware	2	Type 1	Type 2			
C	Obscuration	3	Low	Moderate	High		
D	Attack Frequency	3	Rarely	Sometimes	Often		
E	Weakness Prevalence	3	Rarely	Sometimes	Often		
F	CWE Category	5	Type 1	Type 2	Type 3	Type 4	Type 5

Test Setup

- Figure shows a test setup for a representative system.
- The testing needs to analyze the impact of each factors on the system as well as determine if there are any interactions among factors.
- Other tests can be designed at different abstraction levels depending on the requirement.



Test Model

Conventional test approach

- In order to run a full factorial test considering three-level input factors (obscuration, frequency & prevalence) with two-level hardware & software configuration and five-level CWE categories, conventional full factorial test approach would require
  - full factorial = $3^3 \times 2 \times 2 \times 5 = 540$ runs

  If we consider only two-level for obscuration, frequency & prevalence then we require
  - full factorial = $2^3 \times 2 \times 2 \times 5 = 160$ runs

Orthogonal All-Pair Tests

- Using HTT the screening can be done using just 18 runs.
- These 18 tests cover all-pairs test combinations and provide an efficient way to screen test the SCADA system for the most significant areas of weakness.
- If the number of applications or the number of hardware types grows (e.g. five), the HTT OA test design will only grow to 25 runs whereas the conventional approach will grow to over 3000 for 3-level and 1000 for 2-level full factorial testing.

Problem Formulation - Display Log - OA Test Plan

OA : 18 Tests | Deg of Freedom : 13 | All Combinations : 540

Test No.	Dupl. Of Test	A : Application	B : Hardware	C : Obscuration	D : Attack Frequency	E : Weakness Prevalence	F : CWE Category
1		Active	Type 1	Low	Rarely	Rarely	Type 1
2		Not Active	Type 2	Moderate	Sometimes	Sometimes	Type 1
3		Active	Type 1	High	Often	Often	Type 1
4		Not Active	Type 1	Low	Rarely	Sometimes	Type 2
5		Active	Type 1	Moderate	Sometimes	Often	Type 2
6		Active	Type 2	High	Often	Rarely	Type 2
7		Active	Type 2	Low	Sometimes	Rarely	Type 3
8		Active	Type 1	Moderate	Often	Sometimes	Type 3
9		Not Active	Type 1	High	Rarely	Often	Type 3
10		Not Active	Type 2	Low	Often	Often	Type 4
11		Active	Type 1	Moderate	Rarely	Rarely	Type 4
12		Active	Type 1	High	Sometimes	Sometimes	Type 4
13		Active	Type 1	Low	Sometimes	Often	Type 5
14		Not Active	Type 1	Moderate	Often	Rarely	Type 5
15		Active	Type 2	High	Rarely	Sometimes	Type 5
16		Active	Type 1	Low	Often	Sometimes	Type 1
17		Active	Type 2	Moderate	Rarely	Often	Type 1
18		Not Active	Type 1	High	Sometimes	Rarely	Type 1

Conclusion

- In this work we presented few ways of identification and preparation for the threat.
- We investigated that publically available information on software weaknesses such as CWE & CWSS can be used to structure defensive & offensive test of ICS/SCADA systems.
- Due to the multiplicative effect of possible architecture permutations, defensive postures, offensive threats & weakness types, such testing needs to use HTT for an efficient test strategy.
- We are seeking opportunities to:
  - apply these research techniques &
  - teach cybersecurity practitioners how to use HTT in the UNSW subject ZEIT 8034
- https://www.unsw.adfa.edu.au/capability-systems-centre/advanced-test-and-evaluation-techniques#overlay-context=

27th annual INCOSE international symposium

Adelaide, Australia

July 15 - 20, 2017

ENGINEERS AUSTRALIA

SESA

www.incose.org/symp2017