CyBOK Mapping Framework for NCSC Certified Degrees

Guidance Document for UK Higher Education

Lata Nautiyal | University of Bristol

Awais Rashid | University of Bristol

1 STEP BY STEP IMPLEMENTATION OF MAPPING PROCESS BY TAKING EXAMPLE OF ONE MODULE DESCRIPTION - FROM HARVARD UNIVERSITY, USA

1.1 Formation Phase:

(Introduction to Cybersecurity: Harvard University, USA)

Cybersecurity Introduction

- Introduction: Cybersecurity
- Learning from the past: Multics
- Examples of what can go wrong
- Capability architectures
- Tagged architectures, including Memory safety, Type safety, Information flow, and “Zero Kernel” Security Overview
- Why security is a hard goal to achieve
- Broad strategies that one can employ to create secure systems

Systems Security

- Hardware Architectures for Security
- How novel hardware architectures can help to enforce the security properties that Operating Systems and Programming Languages expect, including memory safety, type safety, information flow, and access control
- How to enforce properties in hardware can be much more systematic and dramatically more efficient than enforcement by software alone Operating Systems Security
- Taking a global, systems-wide view of security. Viewing security as a ”negative goal,” considering all possible paths to security breaches- permissions, access, trojans, bugs, and many others
- Discussion of various design approaches to securing systems, including complete mediation, separation privilege, and minimizing the trusted computer base Verifying Systems
- How to formulate requirements on secure behavior of C-like programs as rigorous logical formulas
- How to argue that programs meet such requirements

(Introduction to Cybersecurity: Harvard University, USA)

Cybersecurity Introduction

- Introduction: Cybersecurity
- Learning from the past: Multics
- Examples of what can go wrong

- Capability architectures
- Tagged architectures, including Memory safety, Type safety, Information flow, and Zero Kernel, Security Overview
- Why security is a hard goal to achieve
- Broad strategies that one can employ to create secure systems

Systems Security

- Hardware Architectures for Security
- How novel hardware architectures can help to enforce the security properties that Operating Systems and Programming Languages expect, including memory safety, type safety, information flow, and access control
- How to enforce properties in hardware can be much more systematic and dramatically more efficient than enforcement by software alone Operating Systems Security
- Taking a global, systems-wide view of security. Viewing security as a ”negative goal,” considering all possible paths to security breaches- permissions, access, trojans, bugs, and many others
- Discussion of various design approaches to securing systems, including complete mediation, separation privilege, and minimizing the trusted computer base Verifying Systems
- How to formulate requirements on secure behavior of C-like programs as rigorous logical formulas
- How to argue that programs meet such requirements

1.2 Connecting Phase:

Searching for those highlighted ***keywords*** or a ***set of keywords*** using the resources in the ***“CyBOK Mapping Structure Guide”***. This phase is comprised of 5 steps (**Steps A** to **E)**.

Step A: – Mapping with an alphabetical version of the CyBOK knowledge areas indicative material from NCSC’s certification document: –

Start your search with this document. If your Highlighted/Underlined ***keywords*** or a ***set of keywords*** are found in this part, then record these in the table and move on to the next ***keywords*** or a ***set of keywords***. Repeat the process until the last ***keywords*** or a ***set of keywords***. **(Move to step B)**

S.No.	Broad Category	KA	Topic	Indicative Material	Keyword or a Set of Keywords	Mapping with an alphabetical version of the CyBOK knowledge areas indicative material
1					Introduction: Cybersecurity	Not Found
2	Systems Security	OSV	Primitives for isolation and mediation	Multics	Multics	Found and Recorded
3					Examples of what can go wrong (Vulnerabilities and security issue)	Not Found
4	Systems Security	OSV	Primitives for isolation and mediation	Capabilities	Capability architectures	Found and Recorded
5					Tagged architectures	Not Found
6	Systems Security	OSV	Primitives for isolation and mediation	Memory protection and address spaces	Memory safety (Memory protection and address spaces)	Found and Recorded
7					Type safety	Not Found
8	Software and Platform Security	SS	Prevention of vulnerabilities	Information flow	Information flow	Found and Recorded
9					Zero Kernel	Not Found
10					Security Overview	Not Found
11					security is a hard goal to achieve, Broad strategies that one can employ to create secure systems	Not Found
12	Systems Security	OSV	OS security principles	Security models	Hardware architectures can help to enforce the security properties that Operating Systems and Programming Languages expect, including memory safety, type safety, information flow, and access control (Security models)	Found and Recorded
13					How to enforce properties in hardware can be much more systematic and dramatically more efficient than enforcement by software alone Operating Systems Security	Not Found
14					wide view of security	Not Found
15					security breaches	Not Found
16					permissions	Not Found
17					access	Not Found
18					trojans	Not Found
19					bugs	Not Found
20	Systems Security	OSV	Role of operating system	Design choices	design approaches to securing systems	Found and Recorded
21	Systems Security	OSV	Role of operating system	Mediation	complete mediation (mediation)	Found and Recorded

22	Systems Security	OSV	OS security principles	Saltzer and Shroeder Principles	separation privilege (Saltzer and Shroeder Principles)	Found and Recorded
23					Computer base Verifying Systems	Not Found
24					formulate requirements on secure behavior of C-like programs as rigorous logical formulas	Not Found
25					How to argue that programs meet such requirements	Not Found

Step B: – Mapping with CyBOK Mapping Reference 1.1: –

Continue your search with this document. If your remaining **(Not Found)** ***keywords*** or a ***set of keywords*** are found in this part, then record these in the table and move on to the next ***keywords*** or a ***set of keywords***. Repeat the process until the last ***keywords*** or a ***set of keywords***. **(Move to step C)**

S.No.	Broad Category	KA	Keyword or a Set of Keywords	Mapping with CyBOK Mapping Reference 1.1
1			Introduction : Cybersecurity	Not Found
3	Software and Platform Security	SS CPS	Examples of what can go wrong (Vulnerabilities and security issue)	Found and Recorded (Selected SS as relevant)
5	Systems Security	OSV	Tagged architectures (Hybrid Microkernel architectures)	Found and Recorded
7	Software and Platform Security	SS	Type safety (Safe languages)	Found and Recorded
9	Systems Security	OSV	Zero Kernel (Kernel)	Found and Recorded
10			Security Overview	Not Found
11	Attacks and Defences	SOIM	security is a hard goal to achieve, Broad strategies that one can employ to create secure systems (security-system-level)	Found and Recorded
13			How to enforce properties in hardware can be much more systematic and dramatically more efficient than enforcement by software alone Operating Systems Security	Not Found
14			Wide view of security	Not Found
15	Attacks and Defences	SOIM	security breaches (breaches)	Found and Recorded
16	Systems Security	AAA	permissions	Found and Recorded
17	Systems Security	AAA	access	Found and Recorded
18	Attacks and Defences	MAT	trojans	Found and Recorded
19	Software and Platform Security	SS	bugs	Found and Recorded
23			Computer base Verifying Systems	Not Found
24	Software and Platform Security	SS	formulate requirements on secure behavior of C-like programs as rigorous logical formulas (Program logic)	Found and Recorded
25			How to argue that programs meet such requirements	Not Found

Step C: – Complete the missing topics and indicative material from CyBOK Knowledge Trees for all the recorded keyword or a set of keywords found through CyBOK Mapping reference 1.1: –

Searching topics from CyBOK Knowledge Trees for all the recorded ***keywords*** or a ***set of keywords*** found through CyBOK Mapping reference 1.1 as it provides relevant CyBOK knowledge areas but not the topic and indicative material, therefore CyBOK Knowledge Trees are used. **(Move to step D)**

S.No.	Broad Category	KA	Topic	Indicative Material	Keyword or a set of Keywords	Mapping missing Topics and Indicative Material from CyBOK Knowledge Trees
3	Software and Platform Security	SS CPS	Categories of vulnerabilities	CVEs and CWEs or memory management vulnerabilities or structured output generation vulnerabilities or race condition vulnerabilities or API vulnerabilities or side channel vulnerabilities	Examples of what can go wrong Can be assumed as (Vulnerabilities and security issue)	Found and Recorded (Selected SS as relevant)
5	Systems Security	OSV	Primitives for isolation and mediation	Modern hardware extensions for memory protection	Tagged architectures (Hybrid Microkernel architectures)	Found and Recorded
7	Software and Platform Security	SS	Prevention of Vulnerabilities	Coding practices	Type safety (Safe languages)	Found and Recorded
9	Systems Security	OSV	Role of Operating Systems	Design choices	Zero Kernel (Kernel)	Found and Recorded
11	Attacks and Defences	SOIM	Fundamental Concepts	***	security is a hard goal to achieve, Broad strategies that one can employ to create secure systems (security-system-level)	Found and Recorded
15	Attacks and Defences	SOIM	Monitor: data sources	***	security breaches (breaches)	Found and Recorded
16	Systems Security	AAA	Authorisation	Access control	permissions	Found and Recorded
17	Systems Security	AAA	Authorisation	Access control	access	Found and Recorded
18	Attacks and Defences	MAT	Malware Taxonomy	Kinds	trojans	Found and Recorded
19	Software and Platform Security	SS	Categories of vulnerabilities	***	bugs	Found and Recorded
24	Software and Platform Security	SS	Prevention of Vulnerabilities	Coding practices	formulate requirements on secure behavior of C-like programs as rigorous logical formulas (Program logic)	Found and Recorded

Step D:– Mapping with CyBOK Knowledge Trees: –

Continue your search with this document. If your Highlighted/Underlined ***keywords*** or a ***set of keywords*** are found in this part, then record these in the table and move on to the next ***keywords*** or a ***set of keywords***. Repeat the process until the last ***keywords*** or a ***set of keywords***. **(Move to step E)**

S.No.	Broad Category	KA	Topic	Indicative Material	Keyword or a set of Keywords	Mapping with CyBOK Knowledge Trees
1	CyBOK Introduction	CI	Foundational Concepts	Definition of cyber security	Introduction: Cybersecurity	Found and Recorded
10	CyBOK Introduction	CI	Foundational Concepts	Definition of cyber security	Security overview	Found and Recorded
13	Systems Security	OSV	Primitives for Isolation and Mediation	Modern hardware extensions for memory protection	How to enforce properties in hardware can be much more systematic and dramatically more efficient than enforcement by software alone Operating Systems Security	Found and Recorded
14	CyBOK Introduction	CI	Foundational Concepts	Definition of cyber security	Wide view of security	Found and Recorded
23	Formal Method for Security	FMS	Analysis and Verification	***	Computer base Verifying Systems	Found and Recorded
25	Formal Method for Security	FMS	Analysis and Verification	***	How to argue that programs meet such requirements	Found and Recorded

Step E:– Complete final missing keywords using the Tabular representation of CyBOK broad categories, knowledge areas and their description: –

If the ***keywords*** or a ***set of keywords*** are not found in any of the materials provided to support the mapping process then identify the most relevant knowledge area using this document and then record the relevant Knowledge Areas.

Not Applicable - All the keywords have been mapped by using Step A to D

1.3 Finalising Phase:

This phase is a result of the mapping process; the results are transferred from the various tables to the **Final table**. It will be helpful to fill **Table (3.3)** in the application for NCSC certification. **Table (3.3)** is required as a part of the application for NCSC certification.

Broad Category	KA	Topic	Indicative Material	Keyword/ Set of Keywords/Course keywords
CyBOK Introduction	CI	Foundational Concepts	Definition of cyber security	Introduction: Cybersecurity
Systems Security	OSV	Primitives for isolation and mediation	Multics	Multics

Software and Platform Security	SS	Categories of vulnerabilities	CVEs and CWEs or memory management vulnerabilities or structured output generation vulnerabilities or race condition vulnerabilities or API vulnerabilities or side channel vulnerabilities	Examples of what can go wrong
Systems Security	OSV	Primitives for isolation and mediation	Capabilities	Capability architectures
Systems Security	OSV	Primitives for isolation and mediation	Modern hardware extensions for memory protection	Tagged architectures
Systems Security	OSV	Primitives for isolation and mediation	Memory protection and address spaces	Memory safety
Software and Platform Security	SS	Prevention of Vulnerabilities	Coding Practices	Type safety
Software and Platform Security	SS	Prevention of vulnerabilities	Information flow	Information flow
Systems Security	OSV	Role of Operating Systems	Design choices	Zero Kernel
CyBOK Introduction	CI	Foundational Concepts	Definition of cyber security	Security Overview
Attacks and Defences	SOIM	Fundamental Concepts	***	security is a hard goal to achieve, Broad strategies that one can employ to create secure systems
Systems Security	OSV	OS security principles	Security models	Hardware architectures can help to enforce the security properties that Operating Systems and Programming Languages expect, including memory safety, type safety, information flow, and access control
Systems Security	OSV	Primitives for Isolation and Mediation	Modern hardware extensions for memory protection	How to enforce properties in hardware can be much more systematic and dramatically more efficient than enforcement by software alone Operating Systems Security
CyBOK Introduction	CI	Foundational Concepts	Definition of cyber security	wide view of security
Attacks and Defences	SOIM	Monitor: data sources	***	security breaches
Systems Security	AAA	Authorisation	Access control	permissions
Systems Security	AAA	Authorisation	Access control	access
Attacks and Defences	MAT	Malware Taxonomy	Kinds	trojans
Software and Platform Security	SS	Categories of vulnerabilities	***	bugs
Systems Security	OSV	Role of operating system	Design choices	design approaches to securing systems
Systems Security	OSV	Role of operating system	Mediation	complete mediation
Systems Security	OSV	OS security principles	Saltzer and Schroeder's principles	separation privilege
Formal Methods for Security	FMS	Analysis and Verification	***	Computer base Verifying Systems

Software and Platform Security	SS	Prevention of Vulnerabilities	Coding Practices	formulate requirements on secure behavior of C-like programs as rigorous logical formulas
Formal Method for Security	FMS	Analysis and Verification	***	How to argue that programs meet such requirements

Note :- Some topics are too broad to be covered in a single KA, therefore if terms are so broad, they can't be mapped without more context. It is better to consider the context and then record the appropriate Indicate Material, Topic, Knowledge Areas and Broad Category.

*** Indicated that there is no direct mapping of keyword with Indicative material but with Topic coverage.

2 SOURCE OF THE MODULE CONTENTS

https://canvas.harvard.edu/courses/63118/assignments/syllabus