OWASP

Java / J2EE Security

Curriculum

Authored by

Dr. A, L. Gottlieb

The OWASP Java/J2EE software security curriculum is offered as prescriptive guidance for those wishing to educate themselves or others on how to secure Java/J2EE software development.

Curriculum goals:

1. Provide prescriptive guidance to those who wish to educate themselves or others in secure software development.

2. To provide software developers with extensive educational opportunities for professional growth into secure software development throughout their career.

3. Classes that provide students with content reflecting a sound appreciation for the subject and is evidenced by the student skills proficiency.

Curriculum Structure:

The curriculum is segmented by track where each track represents a software engineering job function. Job functions include software architect, requirements analyst, software designer, programmer and tester. For each job function there are mandatory core courses. Students are required to complete the core courses contained in their job function track before choosing a specialization track that is based on security function.

A note on adopting the OWASP Java/J2EE curriculum

The OWASP Java/J2EE curriculum is comprehensive with respect to the application security space and its content is robust. And you may rely on OWASP to maintain the curriculum's informative and helpful guidance. One should not think however that a certain special need or set of goals can't indicate a different curriculum. When such a curriculum is indicated, please feel free to use the OWASP Java/J2EE curriculum as a reference point.

Tracks Based on Job Function with their core courses

1. **Software Architect (core courses)**
   Software Risk Analysis
   Security requirements Methodology
   Architectural artifact creation for secure systems
   Vulnerabilities discovery during the Security Requirements Phase
   Threat Modeling
   Topics in High Assurance Software Design
   Web Application security testing
   Full lifecycle test documentation compilation and validation

2. **Requirements Analyst (core courses)**
   Software Risk Analysis
   Architectural artifact creation for secure systems
   Vulnerabilities discovery during the Security Requirements Phase
   Threat Modeling
   Security Requirements methodology
   Web Application security testing
   Full lifecycle test documentation compilation and validation

   Software Designer (core courses)
   Software Risk Analysis
   Architectural artifact creation for secure systems
   Vulnerabilities discovery during the High Assurance Design Phase
   Threat Modeling
   Topics in High Assurance Software Design
   Web Application security testing
   Full lifecycle test documentation compilation and validation

3. **Programmer (core courses)**
   Security Manager with Class Loading/Verification and the Permission Class Hierarchy
   Programmatic *user centric* authentication and authorization using JAAS
   Declarative *code centric* security of J2EE components
   Manual Code Review of Java source
   Automated Code Review of Java source
   Safe Programming Practices in Java

4. **Security Tester (core courses)**
   Full lifecycle test documentation compilation and validation
   Architectural phase Attacks
   Security Requirements Testing
   Web Application Security Testing
   Crafting input validation

Specialization Tracks

Risk Analysis

The OWASP Risk Rating Methodology
ARA
FAIR
Mosaic
Comparative Study in Risk Assessment and Analysis

Security requirements Methodologies

SQUARE
ARM
Tropos
Reusable Security Requirements
Safety Related Requirements for Software Intensive Systems
Comparative study of security requirements methodologies

Architectural artifact creation for secure systems

Use cases and Misuse Cases for Secure Software Development
Data Flow Diagrams
UML Security Extensions Workshop
Attack Trees
Modeling SOA Architectures using BPMN
Modeling SOA Architectures using UML

Attacks, Vulnerabilities and Threat Modeling

Analysis of the OWASP 10
Architectural Phase Attacks
Threat Modeling with STRIDE
Microsoft Threat Modeling
Taxonomy of miscellaneous Attacks

Topics in High Assurance Software Design

Single-Signon using Kerberos Authentication in Java
J2EE Login Authentication for web containers
Role Engineering for new and legacy software system access control using the RBAC Model
Access control using the ABAC Model XACML based authorization
Access control using the PBAC Model
Access Control using the RadaC Model
PKI

Software security testing

Risk Based Security Testing
Fuzz Testing
OWASP Security Testing of Web Services
Selected Topics from the OWASP Testing Guide 3.0

Safe Programming Practices and Code Review in Java

Secure Programming in Java Part1
Secure Programming in Java Part2
Manual Code Review of Java Part2
Manual Code Review of Java Part3

Guidance to Software Development Management on Education

Apply enough security controls to the software your developing and get your people trained to do it

As the curriculum clearly shows, there are security analyses and controls at each phase of the software development life cycle. Decisions regarding which security controls at those phases shouldn't be made arbitrarily. Instead, OWASP recommends risk analysis, software architectural review (threat modeling), data classification and other system specific influencing factors in making such decisions. Some level of security is likely to be required for almost every type of software. Find that level and apply those security controls with personnel who are well trained. Additionally, the decision to employ a particular software development methodology should also be based on the security controls required by the system under consideration. If by selecting a development methodology, required security analyses or controls are precluded, the developer may be assuming an inordinate or irresponsible amount of risk.

Guidance to education management on courseware development and delivery

Core courses and specialization tracks

It is understood that in addition to developing a full curriculum, some Java/J2EE training providers will emphasize the core courses for one or more job functions whereas others may concentrate their efforts on one or more specialization tracks such as *Safe Programming Practices and Code review in Java*. Indeed, OWASP would enthusiastically recognize the building of a single course such as *Manual Code Review of Java part1*, if it were deemed superior by virtue of its content, courseware development techniques and in-class delivery. Here, content is provided by subject matter experts. Courseware is provided by courseware developers and in-class delivery is by instructors.

Course Content and Subject Matter Experts

A subject matter expert (SME) is just that, an expert with superior knowledge and vast experience with the subject matter to be fashioned into a customer course. Content errors while teaching a customer class are difficult to recover from because it undermines the instructor's credibility and the integrity of the training organization. Therefore the flow of information between the SME and the courseware developer must be timely and unobstructed. In short, the SME's goal is to maintain their superior knowledge and to impart their knowledge to the satisfaction of the courseware developer. And in some cases such as when a lab exercise during a class is not behaving as described, a SME may be pressed into servicing the instructor's request for support. OWASP does not endorse vertical ownership of content (*i.e. course content, courseware development and in-class delivery*) by SMEs.

Courseware development

Student needs may change as the software security control requirements change, based on changing conditions in the wild. This situation demands an upgrade to the content in existing courses or perhaps new courses. In either case, it is the position of OWASP that education providers give consideration to how the new content is to be taught in addition to what content will be included. Here, the advice of creative courseware developers working together with subject matter experts will yield a good course. Courseware developers can have master's degrees in education or instructional technology but as it is the way to teach the subject matter that ends up in the student materials, understand this should be an extremely creative process.

Things Instructors do:

1. Teach students during lecture
2. Provide assistance and helpful hints during lab
3. Take inventory of any students special needs or pre-disposition about the class
4. Return to students with answers to questions
5. Setup the classroom environment
6. Make sure their students are happy with the class because that sells more education
7. Prepare to teach new courses
8. Work as courseware developers when custom training requirements come in
9. Teach a full schedule of classes, some requiring travel to the customer site

Clearly, these are the duties of a fulltime instructor. Fulltime instructors get a new bunch every week causing them to advance through experience their group interaction, interpersonal skills and the way they teach specific topics in their courses. The OWASP global education committee recommends to software security training providers to employ instructors to teach our curriculum.

Course Descriptions

🎗 Indicates courses possibly requiring express written permission, teaching certification and licensing fee to develop and sell training

Software Risk Analysis

This course explores the gathering of the requisite data to make business-level decisions based on vulnerabilities, threats, impact and probability. Elements of risk analysis as it pertains to software are defined including asset, risk and impact. Students will determine risk using mathematical and qualitative techniques.

Security requirements Methodology

This course introduces security requirements capture in software development. Concepts covered include the types of security requirements, the chronological steps in requirements engineering, elements of good requirements and compliance factors driving security needs. Students will attain the requisite knowledge of this vast beginning phase of the SDLC as well as its security implications on later phases.

Architectural artifact creation for secure systems

Architectural modeling of software renders an analyzable view of the software called an artifact. This course teaches techniques in artifact creation for software including UML, UMLsec, use cases, misuse cases, attack trees and abuse cases. The skills gained by the student can then be incorporated during the early phases of development such as threat modeling and security requirements capture.

Vulnerabilities discovery during the Security Requirements Phase

Attacks against application software can be discovered throughout security requirements capture and high assurance design. This course is the starting point for mastery of vulnerability discovery and knowledge. The successful student will learn through analysis to identify vulnerabilities throughout the security requirements phase of development.

Vulnerabilities discovery during the High Assurance Design Phase

Attacks against application software can be discovered throughout, high assurance design. This course is the starting point for mastery of vulnerability discovery and knowledge. The successful student will learn through analysis to identify vulnerabilities throughout the high assurance design phase of development.

Topics in High Assurance Software Design

This course familiarizes students with security considerations inexorably linked to the design phase of the SDLC. Topics include cryptography, transaction integrity, access control and data transport. Lab exercises challenge the student to identify functionally sound but non-secured aspects of existing design documentation.

Web Application Security Testing

This course presents a vast arsenal of testing techniques for web application software. Free tools such as OWASP's WebScarab are used to observe live post data. Various data encoding techniques are explored. Included too is designing security test cases against session state, cookie manipulation and vulnerabilities including cross site scripting and directory traversal.

Security Manager with Class Loading/Verification and the Permission Class Hierarchy

The individual responsibilities and codependency's between the J2SE Class loader, Class verifier and Security Manager combine to form the security fundamentals of the J2SE security architecture. The student will practice class loader methods. The permission class hierarchy is a tree structure containing permission classes/objects and methods. Students will learn to create new permissions.

Programmatic user centric authentication and authorization using JAAS

JAAS is a Java (JRE) security framework providing user-centric authentication and authorization. JAAS augments Java code-based authorization. Topics include principal based authentication, deep analysis of the ProtectionDomain-based authorization algorithm with doAs () and doAsPrivileged () examples and adding a subject to a thread. Students will learn to build the policy file. This course has very demanding programming requirements.

Declarative security of J2EE components

Declarative security policies are bundled with J2EE application web-components, such as servlets and EJB components. Topics in this course include declarative authorization, login-configuration policy, delegation and connection policies, servlet and EJB security. Lab exercises will involve preparation of deployment descriptors containing XML security elements.

Manual Code Review of Java source

At OWASP, we believe every programmer should possess basic security focused code inspection skills. This course teaches programmers detection and remediation procedures relative to members of the OWASP 10 and other vulnerabilities such as Log Forgery and Resource Exhaustion. Students will learn the basic skills needed to perform a security focused Java code review.

Safe Programming Practices in Java

There are with Java, as with all programming languages, coding practices which yield more secure applications. This course covers ESAPI; an OWASP web application security controls library enabling programmers to build lower risk applications. Students will implement ESAPI classes and packages covering cryptography, input validation, authentication and access control.

Full lifecycle test documentation compilation and validation

Security requirements, high assurance design and threat modeling documentation are by themselves, of limited use to testers who must transpose them into meaning application test plans with test cases. Typically, this is done by testers with perhaps a few meetings with the authors. This course teaches testers and documentation authors to write agreed upon testing documentation earlier in the SDLC.

Threat Modeling

Threat modeling allows architects to view an entire software application so that threats are addressed before they become real attacks against the completed software. This course explores attack centric, software centric and asset centric approaches. Students will be assigned roles, then as part of a group, will threat model a web application using a threat modeling technique to be determined by the instructor.

Automated Code Review of Java

In this course, students are familiarized with the operational aspects of static analysis tools such as Fortify and Ounce. Interpretation and prioritization of tool output is presented along with techniques for minimizing the frequency of false positives. Lab exercises will be conducted in both an individual and group setting emphasizing expeditious retirement of the security issues detected in Java source code.

Role Engineering for new and legacy software system access control using the RBAC Model

Role engineering is the definition of roles with their names and permissions. Constraints on roles are included in the definition as is the eventual structured hierarchy of roles. Topics include bottom-up and top-down approaches, defining good roles and role engineering as requirements engineering. Lab exercises will emphasize planning a role engineering project and execution of the same project.

SQUARE

With an emphasis on heavy stakeholder involvement and its focus on building security into the SDLC as early as possible, SQUARE is a nine step process providing means for eliciting, categorizing and prioritizing security requirements. The students will be assigned to groups which will complete the SQUARE process, yielding a set of security requirements.

ARM

ARM (Accelerated Requirements Method) is a facilitated requirements elicitation and description activity. This course explores the Preparation, Session and Closure phases of the method and culminates with lab exercises simulating full enactment of each phase. Successful completion of the course will enable security requirements analysts to choose ARM instead of a conventional interviewing technique.

Secure Tropos -

Tropos is a methodology for agent-oriented software development. This course focuses on the integration of security analysis in the early requirement stage of Tropos. Explored are the concepts of security constraints, security dependencies and security entities. Students will construct actor diagrams leading to formal requirements.

Reusable Security Requirements

Security requirements are limited in their scope compared to functional requirements. Threats and attacks lead toward uniformity with respect to architectural security controls thus making their reuse by other application software possible. This course provides an in-depth study of parameterized reuse of software security requirements. Topics include parameterized templates and asset-based risk driven analysis.

Safety Related Requirements for Software Intensive Systems

This course explores the integration of requirements and safety engineering, two disciplines having their own concepts, terminologies, techniques and tools. A combined method including tasks, roles and output is suggested. Asset, Hazard and Safety-Risk Analyses are presented in support of a collaborative effort between safety and requirements engineering teams. Permission from SEI may be required

Comparative study of security requirements methodologies

There are various approaches to security requirements engineering including Secure Tropos which focuses on agent-based software and holistic security requirement engineering for E-commerce applications. This course examines a group of security requirements engineering techniques for their strengths, weaknesses and appropriateness. Students will provide consultation on technique selection.

Access control using the ABAC Model XACML based authorization

Attribute-Based Access Control (ABAC) uses attributes to define access control rules. Attributes themselves are associated with access requestors. Key concepts presented are subject, action, resource and environment. Students will employ attributes in a structured language to build access control rules. Upon receiving an access request, students will decide to either allow or deny access based on comparing a stated policy to the requestor's attributes.

Access control using the PBAC Model

In industries where government has exercised its authority to demand austere security controls, the Policy-Based Access Control model aids enterprises in implementing access controls based on abstract policy and governance requirements. Topics include various compliance regulations, enterprise wide data repositories and authoritative attribute source. Students will acquire the knowledge needed to consult on this complex model.

Access Control using the RadaC Model

The Risk-Adaptive Access Control model augments earlier access control models by considering environmental conditions and risk levels in the decision to permit or deny access. Exogenous variables such as requestor trustworthiness, organizational IT structure and environmental risk factors will be defined and described in terms of their place as inputs to the decision making process. Demanding by design, RadaC challenges to implementation will be discussed.

PKI

This course presents both concepts and supporting code for the JCE (integrated with Java 2 SDK V1.4) and JCA, which together provide a platform-independent cryptography API. Signature, KeyPairGenerator, X509Certificate and MAC classes are covered. Lab exercises produce programs to do signature generation and verification as well as instantiation of the MessageDigest and KeyStore classes.

Analysis of the OWASP 10

This course undertakes a deep study of the OWASP Top 10 Web Application Security Risks. Each risk will be explored in terms of its vulnerability to be found in software, ways to prevent the risk and code scenarios depicting a successful attack. In addition, manual code review techniques pertinent to risks such as the Cross Site Scripting (XSS) attack will be tested in lab exercises.

J2EE Login Authentication for web containers

HTTP basic, Form-based, Client Certificate and Mutual are available authentication mechanisms used to secure web resources. This course explores the client interaction, setup procedures and/or programming tasks required to implement each. Salient matters such as building, packaging and deploying web applications using an authentication mechanism is demonstrated. Students will program and/or configure authentication mechanisms for J2EE web containers.

Security Requirements Testing

A successful, comprehensive software testing program requires a set of security requirements which illuminate the testing objectives throughout the SDLC. Security test derivation techniques used for functional security requirements, security requirements through use and misuse cases, and developer and tester security testing workflow and requirements validation are covered. Lab exercises involve deriving and correctly documenting security requirements.

Single-Signon using Kerberos Authentication in Java

A single sign-on solution can be implemented based on the Kerberos V5 network authentication protocol. Key tasks to be completed by students include using JAAS authentication and authorization to authenticate a principal to Kerberos and fetch their credentials, using the Java Generic Security Service API (Java GSS-API) to authenticate to a remote peer application using the previously obtained Kerberos credentials.

Threat Modeling with STRIDE -

STRIDE endeavors to decompose a system into relevant components which are then analyzed for vulnerability to threats. Once discovered, those threats are then mitigated. The decomposition process is then repeated until you are comfortable with any remaining threats. Topics include Data Flow diagramming, analysis of processes, data flows and data stores. Students will be become proficient in analysis techniques as they threat model a sample application in lab.

Crafting input validation

A class of applications share similar functionality hence; they share a similar testing strategy. Just as certain are applications that are different from each other. These applications demand custom test programs, the subject of this course. Topics include client & server side validation, centralized encoding modules, black & white listing, potential sources of input and regular expressions. Students will acquire testing skills against OWASP 10 vulnerabilities such as SQL injection.

Architectural Risk Analysis (ARA) -

Architectural Risk Analysis (ARA) yields prioritized risks and for each calculates a "*risk-factor*" reflecting the impact to information assets from flaws found in software architecture. Supporting tasks including vulnerability analysis, ambiguity analysis, and underlying platform vulnerability analysis are covered. Students will be mentored through an ARA based on instructor provided architectural artifacts, risk assessment reports, software test results and other information useful to the analysis.

The OWASP Risk Rating Methodology

Estimating the degree of business risk throughout the SDLC is made possible by following the OWASP approach to risk analysis. Here, risk is calculated using the standard risk model:

Risk = likelihood * impact

Approach steps including risk identification, factors for estimating likelihood and impact, severity of the risk and customization of the risk rating model are covered. Labs will task students at each step.

MOSAIC - 

SEI Mission-Oriented Success Analysis and Improvement Criteria (SEI MOSAIC)—a suite of advanced analysis methods serving the software assurance needs of software intensive systems. SEI MOSAIC establishes and carries' forward a well-founded degree of confidence in success throughout the lifecycle. Key concepts include outcome-based risk management, mission risk and mission risk management.

FAIR - 

Factor Analysis of Information Risk (FAIR) by RMI, provides a framework for analyzing, and measuring information risk. Topics include information risk terminology, analyzing risk scenarios, data discovery and selection criteria, computing risk and the ten steps of a basic FAIR analysis. Students will work with *FAIRlite* to complete their lab exercises.

Taxonomy of miscellaneous Attacks

Various attacks against software are studied in terms of their attack vectors, informational impact, operational impact, defense and their targets. Students will learn to associate attacks with their vulnerabilities and find them in an architectural artifact such as a UML model. The student will have the ability to evaluate unsecured legacy software for a security upgrade.

Data Flow Diagrams

DFDs illustrate the flow of data from external entities into a system, the flow of data from one process to another, including its logical storage. Applicable to security requirements capture, threat modeling, risk assessment and bringing design flaws to light, DFD's can reach highly complex architectural levels. Students will be tasked to create DFD's for software architectures of varying size and type.

UML Security Extensions Workshop

This course culminates in the creation of UML models employing the UMLsec and SecureUML extensions. In the case of UMLsec, a set of security requirements will be built. In the case of SecureUML, access control policies based on Role-Based Access Control (RBAC) for protected resources will be modeled. Prerequisite: Basic UML modeling

Use Cases and Misuse Cases for Secure Software Development

Misuse Cases describe a negative interaction between a bad actor and the proposed system. Authoring misuse cases based on brainstorming approaches including attempting to match known vulnerability from a repository of known vulnerabilities to the same vulnerability in the proposed system are covered. Students will use a misuse case template to describe the misuse case and capture security requirements.

Risk-Based software Testing

Risk-Based software testing is the process by which each risk identified through risk analysis is tested. Tests should include risks belonging to the architectural and build phases of software development. In this course students will learn to design tests based on those risks. Test templates, attack scenarios, incident reports and threat modeling are examined for their value in crafting risk-based test cases.

Comparative Study in Risk Assessment and Analysis

Risk analysis is replete with approaches and techniques, some very interesting. And where avoiding a disastrous loss is concerned, knowledge of alternative, perhaps more suitable risk analyses is important. Topics include human factors, the relationship between regulatory requirements and risk, risk forecasting, text mining and threat-based risk management for federal systems. Students will select a technique on day 1 and present on the same on day 5.

Modeling SOA Architecture using UML

Securing the architectural phases of software development requires an accurate illustration of the architecture and SOA can scale to complex conglomerations. Topics include The SOA manifesto, the SOA design patterns, SOA components, and entry point selection. More detailed components include connections, ESB, the mediation component and services including interaction, information, partner, business application and process services. Lab exercises model components of SOA architecture.

Modeling SOA Architecture using BPMN

This course explores creating artifacts that model SOA architectures. Students will create artifacts and or their components using BPMN. This course covers various levels of process modeling, tasks, events, gateways, orchestration and choreography. Lab exercises model components of SOA architecture.

Selected Topics from the OWASP Testing Guide 3.0

Among an abundance of important information about software security testing, the OWASP testing guide 3.0 contains 9 categories of software testing with 66 controls. Topics and labs come from the guide and are at the instructor's discretion. Possibilities include Incubated vulnerability testing, authorization testing, data validation, session management and various injection vulnerabilities.

Fuzz Testing

Fuzzing, software security testing based on invalid or random inputs expected to produce bad behavior helpful in identifying vulnerabilities. Covered are the fuzzing process, taint analysis, black, gray and white box approaches to fuzzed data and test generation. Students are expected derive creative tests to perform vulnerability analysis.

Attack Trees

Of immense help in determining the set of possible attacks and their steps to success to be mitigated against are attack trees. Students will construct attack trees; assign values to the leaf nodes indicating the attack with the highest probability, the attack with the cheapest low-risk attack and other attack qualifiers as indicated.

OWASP Security Testing of Web Services

This course explores software security testing specific to web services. These XML/parser related vulnerabilities include WS Information Gathering, WSDL testing, XML Structural testing, XML Content-level testing, Naughty SOAP attachments, HTTP GET parameters/REST testing and Replay testing. Lab exercises include developing and running test cases against an instructor supplied web services application.

Secure Programming in Java Part1

Starting with the essentials such as restricting privileges and establishing trust boundaries this course goes on to cover issues of accessibility and extensibility, input and output parameters, objects and classes and serialization and deserialization. In lab, students will engage in code review exercises with and without code modification and testing.

Secure Programming in Java Part2

Object Orientation, Methods, Locking, Threads, exposing standard API's to untrusted code, cleansing or discarding parameters passed by untrusted code, not leaking return values from standard API's to untrusted code, not allowing untrusted code from calling an API that might call reflection APIs, numeric operations and expressions. In lab, students will engage in code review exercises with and without code modification and testing.

Manual Code Review of Java Part2

Students will learn to locate vulnerabilities relative to input validation, authentication, authorization, session management, inheritance, classes, objects and methods through security code inspection. Labs will challenge students to demonstrate their ability to find offending code and make recommendations as to their remediation.

Manual Code Review of Java Part3

Students will learn to locate vulnerabilities relative to EJB, inner classes/reflection, and native methods, file level, memory reference and web container based through security code inspection. Labs will challenge students to demonstrate their ability to find offending pieces of code and make recommendations as to their remediation.

OWASP Global Education Committee members are available for private consultation

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT,

THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. **"Adaptation"** means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. **"Collection"** means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. **"Distribute"** means to make available to the public the original and copies of the Work through sale or other transfer of ownership.

d. **"Licensor"** means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. **"Original Author"** means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. **"Work"** means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a

process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. **"Publicly Perform"** means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. **"Reproduce"** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections; and,

b. to Distribute and Publicly Perform the Work including as incorporated in Collections.

c. For the avoidance of doubt:

    i. **Non-waivable Compulsory License Schemes**. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. **Waivable Compulsory License Schemes**. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. **Voluntary License Schemes**. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats, but otherwise you have no rights to make Adaptations. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. The credit required by this Section 4(b) may be implemented in any reasonable manner; provided,

however, that in the case of a Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

c. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

d. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

e. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.