# SEMESTER II

|  |  |
|---|---|
| **Course Code:** 511<br>**Total Credits:** 04 (60 Lecture Hrs)<br>**University assessment:** 50 marks | **Course Name:** Big Data Analytics<br>**Total Marks:** 100 marks<br>**College/Department assessment:** 50 marks |

Prerequistes:

Fundamental knowledge of Databases

**Course Objectives:**

- To provide an overview of an exciting growing field of big data analytics.
- To introduce the tools required to manage and analyze big data like Hadoop, NoSql MapReduce.
- To teach the fundamental techniques and principles in achieving big data analytics with scalability and streaming capability.
- To enable students to have skills that will help them to solve complex real-world problems in for decision support.

| Units | Details | Lecture Hrs 4 credits |
|---|---|---|
|  | **Module I** |  |
| **I** | Introduction to Big Data, Characteristics of Data, and Big Data Evolution of Big Data, Definition of Big Data, Challenges with big data, Why Big data? Data Warehouse environment, Traditional Business Intelligence versus Big Data. State of Practice in Analytics, Key roles for New Big Data Ecosystems, Examples of big Data Analytics.<br>Big Data Analytics, Introduction to big data analytics, Classification of Analytics, Challenges of Big Data, Importance of Big Data, Big Data Technologies, Data Science, Responsibilities, Soft state eventual consistency. Data Analytics Life Cycle | **12 [OC1-OC2]** |
| **II** | Analytical Theory and Methods: Clustering and Associated Algorithms, Association Rules, Apriori Algorithm, Candidate Rules, Applications of Association Rules, Validation and Testing, Diagnostics, Regression, Linear Regression, Logistic Regression, Additional Regression Models.<br>Analytical Theory and Methods: Classification, Decision Trees, Naïve Bayes, Diagnostics of Classifiers, Additional Classification Methods, Time Series Analysis, Box Jenkins methodology, ARIMA Model, Additional methods. Text Analysis, Steps, Text Analysis Example, Collecting Raw Text, Representing Text, Term Frequency-Inverse Document Frequency (TFIDF), Categorizing Documents by Topics, Determining Sentiments | **12 [OC3-OC4]** |
|  | **Module II** |  |
| **III** | Data Product, Building Data Products at Scale with Hadoop, Data Science Pipeline and Hadoop Ecosystem, Operating System for Big Data, Concepts, Hadoop Architecture, Working | **12[OC5-OC6]** |

| | | |
|---|---|---|
| | with Distributed file system, Working with Distributed Computation, Framework for Python and Hadoop Streaming, Hadoop Streaming, MapReduce with Python, Advanced MapReduce. In-Memory Computing with Spark, Spark Basics, Interactive Spark with PySpark, Writing Spark Applications, | |
| **IV** | **Unit 4**<br>Distributed Analysis and Patterns, Computing with Keys, Design Patterns, Last-Mile Analytics, Data Mining and Warehousing, Structured Data Queries with Hive, HBase, Data Ingestion, Importing Relational data with Sqoop, Injesting stream data with flume. Analytics with higher level APIs, Pig, Spark's higher level APIs. | **12**<br>**OC7** |

,

**Books and References:**

| Sr. No. | Title | Author/s | Publisher | Edition | Year |
|---|---|---|---|---|---|
| 1. | Big Data and Analytics | Subhashini Chellappan<br>Seema Acharya | Wiley | First | |
| 2. | Data Analytics with Hadoop *An Introduction for Data Scientists* | *Benjamin Bengfort and Jenny Kim* | O'Reilly | | 2016 |
| 3. | Big Data and Hadoop | V.K Jain | Khanna Publishing | First | 2018 |

## Course Outcomes(OCs)

Upon completion of this course the Students will be able to:

| | |
|---|---|
| OC1 | Understand Big Data Concepts |
| OC2 | Do Data Collection and Integration |
| OC3 | Develop Data Storage and Management |
| OC4 | Perform Data Preprocessing and Cleaning |
| OC5 | Understand Data Transformation and Feature Engineering |
| OC6 | Perform Exploratory Data Analysis (EDA) |
| OC7 | Use Big Data Analytics Tools |

| | |
|---|---|
| **Course Code:** 512<br>**Total Credits:** 02 (60 Lecture Hrs)<br>**University assessment:** 25 marks | **Course Name:** Big Data Analytics Practical<br>**Total Marks:** 50 marks<br>**College/Department assessment:** 25 marks |

**Prerequisites: Conceptual understanding of Big Data and DBMS**

**Course Objectives:**

**To teach the students the implementation of Big data analytic as per the concepts learnt**

| Units | Sr. No | Details | Lecture Hrs 2 credits |
|---|---|---|---|
| I | 1 | Install, configure and run Hadoop and HDFS ad explore HDFS. | 30 Hrs [OC1-OC2] |
| | 2 | Implement word count / frequency programs using MapReduce | |
| | 3 | Implement an MapReduce program that processes a weather dataset. | |
| | 4 | Implement an application that stores big data in Hbase / MongoDB and manipulate it using R / Python | |
| | 5 | Implement the program in practical 4 using Pig. | |
| | 6 | Configure the Hive and implement the application in Hive. | |
| | 7 | Write a program to illustrate the working of Jaql. | |
| | 8 | Implement the following: | |
| | 9 | Implement Decision tree classification techniques | |
| II | 10 | Implement SVM classification techniques | 30 Hrs [OC3-OC54 |
| | 11 | Solve the following: | |
| | 12 | REGRESSION MODEL Import a data from web storage. Name the dataset and now do Logistic Regression to find out relation between variables that are affecting the admission of a student in an institute based on his or her GRE score, GPA obtained and rank of the student. Also check the model is fit or not. require (foreign), require(MASS). | |
| | 13 | MULTIPLE REGRESSION MODEL Apply multiple regressions, if data have a continuous independent variable. Apply on above dataset. | |
| | 14 | Solve the Following: | |
| | 15 | CLASSIFICATION MODEL a. Install relevant package for classification. b. Choose classifier for classification problem. c. Evaluate the performance of classifier. | |
| | 16 | CLUSTERING MODEL a. Clustering algorithms for unsupervised classification.<br>b. Plot the cluster data using R visualizations. | |

OC 1: Understand the key issues in big data management and its associated applications in intelligent business and scientific computing.

OC 2: Acquire fundamental enabling techniques and scalable algorithms like Hadoop, Map Reduce and NO SQL in big data analytics.

OC 3: Interpret business models and scientific computing paradigms, and apply software tools for big data analytics.

OC 4: Achieve adequate perspectives of big data analytics in various applications like recommender systems, social media applications etc.

| | |
|---|---|
| **Course Code:** 513<br>**Total Credits:** 02 (60 Lecture Hrs)<br>**University assessment:** 50 marks | **Course Name:** Modern Networking<br>**Total Marks:** 100 marks<br>**College/Department assessment:** 50 marks |

| | |
|---|---|
| **Pre requisites** | Fundamentals of Networking |

## Course Objectives(COs)

CO1. To understand the state-of-the-art in network protocols, architectures and applications.

CO2. Analyze existing network protocols and networks.

CO3. Develop new protocols in networking

CO4. To understand how networking research is done

CO5. To investigate novel ideas in the area of Networking via term-long research projects.

| Unit | Details | Lecture Hrs |
|---|---|---|
| | **Module I** | **2 credits** |
| **I** | Modern Networking<br>Elements of Modern Networking<br>The Networking Ecosystem ,Example Network Architectures,Global Network Architecture,A Typical Network Hierarchy Ethernet Applications of Ethernet Standards Ethernet Data Rates Wi-Fi Applications of Wi-Fi,Standards Wi-Fi Data Rates 4G/5G Cellular First Generation Second Generation, Third Generation Fourth Generation Fifth Generation, Cloud Computing Cloud Computing Concepts The Benefits of Cloud Computing Cloud Networking Cloud Storage, Internet of Things Things on the Internet of Things, Evolution Layers of the Internet of Things, Network Convergence Unified Communications, Requirements and Technology Types of Network and Internet Traffic,Elastic Traffic,Inelastic Traffic, Real-Time Traffic Characteristics Demand: Big Data, Cloud Computing, and Mobile TrafficBig Data Cloud Computing,,Mobile Traffic, Requirements: QoS and QoE,,Quality of Service,Quality of Experience, Routing Characteristics, Packet Forwarding, Congestion Control ,Effects of Congestion,Congestion Control Techniques, SDN and NFV Software-Defined Networking,Network Functions Virtualization Modern Networking Elements | **30 Hrs** |
| | Software-Defined Networks<br>SDN: Background and Motivation, Evolving Network Requirements | |

| | | |
|---|---|---|
| | Demand Is Increasing,Supply Is IncreasingTraffic Patterns Are More ComplexTraditional Network Architectures are Inadequate, The SDN Approach Requirements SDN Architecture Characteristics of Software-Defined Networking, SDN- and NFV-Related Standards Standards-Developing Organizations Industry Consortia Open Development Initiatives, SDN Data Plane and OpenFlow SDN Data Plane, Data Plane Functions Data Plane Protocols OpenFlow Logical Network Device Flow Table Structure Flow Table Pipeline, The Use of Multiple Tables Group Table OpenFlow Protocol, SDN Control Plane<br>SDN Control Plane Architecture Control Plane Functions, Southbound Interface Northbound InterfaceRouting, ITU-T Model, OpenDaylight OpenDaylight Architecture OpenDaylight Helium, REST REST Constraints Example REST API, Cooperation and Coordination Among Controllers, Centralized Versus Distributed Controllers, High-Availability Clusters Federated SDN Networks, Border Gateway Protocol Routing and QoS Between Domains, Using BGP for QoS Management IETF SDNi OpenDaylight SNDi SDN Application Plane SDN Application Plane Architecture Northbound Interface Network Services Abstraction Layer Network Applications, User Interface, Network Services Abstraction Layer Abstractions in SDN, Frenetic Traffic Engineering PolicyCop Measurement and Monitoring Security<br>OpenDaylight DDoS Application Data Center Networking, Big Data over SDN Cloud Networking over SDN Mobility and Wireless Information-Centric Networking CCNx, Use of an Abstraction Layer | |
| **II** | Virtualization, Network Functions Virtualization: Concepts and Architecture, Background and Motivation for NFV, Virtual Machines The Virtual Machine Monitor, Architectural Approaches Container Virtualization, NFV Concepts Simple Example of the Use of NFV, NFV Principles High-Level NFV Framework, NFV Benefits and Requirements NFV Benefits, NFV Requirements, NFV Reference Architecture NFV Management and Orchestration, Reference Points Implementation, NFV Functionality, NFV Infrastructure,Container Interface,Deployment of NFVI Containers,Logical Structure of NFVI Domains,Compute Domain, Hypervisor Domain,Infrastructure Network Domain, Virtualized Network Functions, VNF Interfaces,VNFC to VNFC Communication,VNF Scaling, NFV Management and Orchestration, Virtualized Infrastructure Manager,Virtual Network Function | **30 Hrs** |

|  |  |  |
|---|---|---|
|  | Manager,NFV Orchestrator, Repositories, Element Management, OSS/BSS, NFV Use Cases Architectural Use Cases, Service-Oriented Use Cases, SDN and NFV<br>Network Virtualization, Virtual LANs ,The Use of Virtual LANs,Defining VLANs, Communicating VLAN Membership,IEEE 802.1Q VLAN Standard, Nested VLANs, OpenFlow VLAN Support, Virtual Private Networks, IPsec VPNs,MPLS VPNs, Network Virtualization, Simplified Example, Network Virtualization Architecture, Benefits of Network Virtualization, OpenDaylight's Virtual Tenant Network, Software-Defined Infrastructure,Software-Defined Storage, SDI Architecture |  |
|  | Defining and Supporting User Needs, Quality of Service, Background, QoS Architectural Framework, Data Plane, Control Plane, Management Plane, Integrated Services Architecture, ISA Approach<br>ISA Components, ISA Services, Queuing Discipline, Differentiated Services, Services, DiffServ Field, DiffServ Configuration and Operation, Per-Hop Behavior, Default Forwarding PHB, Service Level Agreements, IP Performance Metrics, OpenFlow QoS Support, Queue Structures, Meters, QoE: User Quality of Experience, Why QoE?,Online Video Content Delivery, Service Failures Due to Inadequate QoE Considerations QoE-Related Standardization Projects, Definition of Quality of Experience, Definition of Quality, Definition of Experience Quality Formation Process, Definition of Quality of Experience, QoE Strategies in Practice, The QoE/QoS Layered Model<br>Summarizing and Merging the ,QoE/QoS Layers, Factors Influencing QoE, Measurements of QoE, Subjective Assessment, Objective Assessment, End-User Device Analytics, Summarizing the QoE Measurement Methods, Applications of QoE Network Design Implications of QoS and QoE Classification of QoE/ QoS Mapping Models, Black-Box Media-Based QoS/QoE Mapping Models, Glass-Box Parameter-Based QoS/QoE Mapping Models,Gray-Box QoS/QoE Mapping Models, Tips for QoS/QoE Mapping Model Selection,IP-Oriented Parameter-Based QoS/QoE Mapping Models,Network Layer QoE/QoS Mapping Models for Video Services, Application Layer QoE/QoS Mapping Models for Video Services Actionable QoE over IP-Based Networks, The System-Oriented Actionable QoE Solution, The Service-Oriented Actionable QoE Solution, QoE Versus QoS Service Monitoring, QoS Monitoring Solutions, QoE Monitoring Solutions, QoE-Based Network and Service Management, QoE-Based Management of VoIP Calls, QoE- | **15** |

| | | |
|---|---|---|
| | Based Host-Centric Vertical Handover, QoE-Based Network-Centric Vertical Handover | |

| Books and References: | | | | | |
|---|---|---|---|---|---|
| **Sr. No.** | **Title** | **Author/s** | **Publisher** | **Edition** | **Year** |
| 1. | Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud | William Stallings | Addison-Wesley Professional | | October 2015 |
| 2. | SDN and NFV Simplified A Visual Guide to Understanding Software Defined Networks and Network Function Virtualization | **Jim Doherty** | Pearson Education, Inc | | |
| 3. | Network Functions Virtualization (NFV) with a Touch of SDN | Rajendra Chayapathi Syed Farrukh Hassan | Addison-Wesley | | |
| 4. | CCIE and CCDE Evolving Technologies Study Guide | Brad dgeworth, Jason Gooley, Ramiro Garza Rios | Pearson Education, Inc | | 2019 |

**Course Outcomes(OCs)**

OC1 Understand the modern networking concepts and implement

| | |
|---|---|
| **Course Code:** 514<br>**Total Credits:** 02 (60 Lecture Hrs)<br>**University assessment:** 25 marks | **Course Name:** Modern Networking Practical<br>**Total Marks:** 50 marks<br>**College/Department assessment:** 25 marks |

Prerequisite: Concepts of Modern Networking

**Course Objectives: To gain practical knowledge in Modern networking**
**All practical are expected to be performed on GNS3/EVE-Ng network Emulator/MININET**

| Units | Sr. No | Details | Lecture Hrs 2 credits |
|---|---|---|---|
| I | 1 | Configure IP SLA Tracking and Path Control Topology | 30 hrs<br>[OC1-OC2] |
| | 2 | Using the AS_PATH Attribute | |
| | 3 | Configuring IBGP and EBGP Sessions, Local Preference, and MED | |
| | 4 | Secure the Management Plane | |
| | 5 | Configure and Verify Path Control Using PBR | |
| II | 6 | IP Service Level Agreements and Remote SPAN in a Campus Environment | 30 Hrs<br>[OC2-OC3] |
| | 7 | Inter-VLAN Routing | |
| | 8 | Simulating MPLS environment and Simulating VRF | |
| | 9 | Simulating SDN with<br>• OpenDaylight SDN Controller with the Mininet Network Emulator<br>• OFNet SDN network emulator | |
| | 10 | Simulating OpenFlow Using MININET | |

OC 1: Demonstrate in-depth knowledge in the area of Computer Networking.

OC 2: To demonstrate scholarship of knowledge through performing in a group to identify, formulate and solve a problem related to Computer Networks

OC 3: Prepare a technical document for the identified Networking System Conducting experiments to analyze the identified research work in building Computer Networks

|  |  |
|---|---|
| **Course Code:** 515<br>**Total Credits:** 02 (30 Lecture Hrs)<br>**University assessment:** 25 marks | **Course Name:** Microservices Architecture<br>**Total Marks:** 50 marks<br>**College/Department assessment:** 25 marks |

**Prerequisites: Networking, cloud concepts**

## Course Objectives(COs)

CO1. Gain a thorough understanding of the philosophy and architecture of Web applications using ASP.NET Core MVC;

CO2. Gain a practical understanding of.NET Core;

CO3. Acquire a working knowledge of Web application development using ASP.NET Core MVC 6 and Visual Studio

CO4. Persist data with XML Serialization and ADO.NET with SQL Server

CO5. Create HTTP services using ASP.NET Core Web API;

CO6. Deploy ASP.NET Core MVC applications to the Windows Azure cloud.

| Units | Details | Lectures |
|---|---|---|
| **I** | **Microservices:** Understanding Microservices, Adopting Microservices, The Microservices Way. **Microservices Value Proposition:** Deriving Business Value, defining a Goal-Oriented, Layered Approach, Applying the Goal-Oriented, Layered Approach. **Designing Microservice Systems:** The Systems Approach to Microservices, A Microservices Design Process, Establishing a Foundation**:** Goals and Principles, Platforms, Culture. | **15**<br>**[OC1]** |
| **II** | **Unit 2**<br>**Service Design:** Microservice Boundaries, API design for Microservices, Data and Microservices, Distributed Transactions and Sagas, Asynchronous Message-Passing and Microservices, dealing with Dependencies, **System Design and Operations:** Independent Deployability, More Servers, Docker and Microservices, Role of Service Discovery, Need for an API Gateway, Monitoring and Alerting.<br>**Adopting Microservices in Practice:** Solution Architecture Guidance, Organizational Guidance, Culture Guidance, Tools and Process Guidance, Services Guidance. | **15**<br>**[OC2]** |

| **Books and References:** | | | | | |
|---|---|---|---|---|---|
| **Sr. No.** | **Title** | **Author/s** | **Publisher** | **Edition** | **Year** |
| 1. | Microservice Architecture: *Aligning Principles, Practices, and Culture* | Irakli Nadareishvili, Ronnie Mitra, Matt McLarty, and Mike Amundsen | O'Reilly | First | 2016 |
| 2. | Building Microservices with ASP.NET Core | Kevin Hoffman | O'Reilly | First | 2017 |
| 3. | Building Microservices: Designing Fine-Grained Systems | Sam Newman | O'Reilly | First | |
| 4. | Production-ready Microservices | Susan J. Fowler | O'Reilly | | 2016 |

## Course Outcomes:

OC 1: Develop web applications using Model View Controller.

OC 2: Think and apply the microservices way to software development.

| | |
|---|---|
| **Course Code:** 516a<br>**Total Credits:** 04<br>**University assessment:** 50 marks | **Course Name:** Malware Analysis Practical<br>**Total Marks:** 100 marks<br>**College/Department assessment:** 50 marks |

**Prerequisites:**

**Basic security concepts**

**Course Objectives(COs)**

CO1. Possess the skills necessary to carry out independent analysis of modern malware samples using both static and dynamic analysis techniques.

CO2. Have an intimate understanding of executable formats, Windows internals and API, and analysis techniques.

CO3. Extract investigative leads from host and network-based indicators associated with a malicious program.

CO4. Apply techniques and concepts to unpack, extract, decrypt, or bypass new anti-analysis techniques in future malware samples.

CO5. Achieve proficiency with industry standard tools including IDA Pro, OllyDbg, WinDBG, PE Explorer, ProcMon etc.

**Course Outcomes:**

After completion of the course, a student should be able to:

**OC 1:** Understand various introductory techniques of malware analysis and creating the testing environment

**OC 2:** Perform advanced dynamic analysis and recognize constructs in assembly code.

**OC 3:** Perform Reverse Engineering using OLLYDBG and WINDBG and study the behaviours and functions of malware

**OC 4:** Understand data encoding, various techniques for anti-disassembly and anti-debugging

**OC 5:** Understand various anti virtual machine techniques and perform shellcode analysis of various languages along with x64 architecture.

**List of Practical as per Annexure I for a total duration of 120 hrs with course outcomes of able to completely perform identification, detection and performing removal and protections process of malware analysis**

| | |
|---|---|
| **Course Code:** 516b | **Course Name:** Cloud Management Practical |
| **Total Credits:** 04 (120 Lecture Hrs) | **Total Marks:** 100 marks |
| **University assessment:** 50 marks | **College/Department assessment:** 50 marks |

**Prerequistes: Basic cloud knowledge**
**Course Objectives:**

- Understand System Center 2019 and its different components.
  Each unit of 30 hrs duration

| **List of Practical:** | |
|---|---|
| Unit I | a. Create and Manage Cloud using SCVMM 2019 |
| | b. Deploy a guarded host fabric using Microsoft SCVMM 2019 |
| | a. Deploy and manage SDN Infra structure using SCVMM 2019 |
| | b. Deploy and Manage Storage Space Direct (S2D) using SCVMM 2019 |
| | a. Deploy Service Manager 2019 and install on 4 Computer Scenario |
| | b. Setup SQL Server reporting Service using Service Manager 2019 |
| | a. User Connectors to import data:<br>i. Import data from Active Directory Domain Services<br>ii. Import data and alerts from Operations Manager<br>iii. Import data from Configuration Manager<br>iv. Import runbooks from Orchestrator<br>v. Import data from VMM<br>vi. Use a CSV file to import data |
| II | b. Automate IT processes with workflows<br>vii. Add or remove workflow activities<br>viii. Configure the way activities manage and pass information<br>ix. Deploy a workflow to Service Manager using the Authoring Tool<br>x. Configure the Activities Toolbox in the Authoring Tool |
| III | a. Managing devices with Configuration Manager |
| | b. Design a hierarchy of sites using Microsoft End Point Configuration manager. |
| | a. Data transfers between sites<br>i. Types of data transfer<br>ii. File-based replication<br>iii. Database replication |
| | b. Configure sites and hierarchies<br>i. Add site system roles<br>ii. Install site system roles<br>iii. Install cloud-based distribution points<br>iv. Configuration options for site system roles<br>v. Database replicas for management points |
| | a. Install Orchestrator. |
| | b. Create and test a monitor runbook |
| | a. Manage Orchestrator Servers – 1<br>i. Runbook permissions |

| | |
|---|---|
| | ii. Back up Orchestrator<br>iii. Bench mark<br>iv. Optimize performance of .Net activities<br>v. Configure runbook throttling<br>vi. Recover a database |
| IV | b. Manage Orchestrator Servers – 2<br>i. Recover web components<br>ii. Add an integration pack<br>iii. View Orchestrator data with PowerPivot<br>iv. Change Orchestrator user groups<br>v. Common activity properties<br>vi. Computer groups |
| | Install and Deploy DPM<br>i. Install DPM<br>ii. Deploy the DPM protection agent<br>iii. Deploy protection groups<br>iv. Configure firewall settings |
| | Protect Workloads<br>i. Back up Hyper-V virtual machines<br>ii. Back up SQL Server with DPM<br>iii. Back up file data with DPM<br>iv. Backup system state and bare metal<br>v. Backup and restore VMware servers<br>vi. Backup and restore VMM servers |

**Course Outcomes:**
After completion of the course, a student should be able to:

**OC 1:** Understand the concepts of VMM, SDN, NAS , HyperV etc.
**OC 2:** Understand and use of Service manager with various deployments that can be performed using it.
**OC 3:** Understand and use SCCM and Demonstrate the use of Configuration Manager
**OC 4:** Use automation with runbooks and demonstrate the use of Windows Orchestrator
**OC 5:** Use Data Protection Manager

| | |
|---|---|
| **Course Code:** 516c<br>**Total Credits:** 04 (120 Lecture Hrs)<br>**University assessment:** 50 marks | **Course Name:** Computer Vision Practical<br>**Total Marks:** 100 marks<br>**College/Department assessment:** 50 marks |

Prerequisites: Knowledge of Digital Image Processing

**Course Objectives:**

CO1. To develop the student's understanding of the issues involved in trying to define and simulate perception.

CO2. To familiarize the student with specific, well known computer vision methods, algorithms and results.

Each Unit of 30 hrs duration

| Units | Details |
|---|---|
| I | Perform Geometric transformations |
| | Perform Image Stitching |
| | Perform Camera Calibration |
| II | Perform the following:<br>a. Face detection<br>b. Object detection<br>c. Pedestrian detection<br>d. Face recognition |
| | Construct 3D model from images |
| | Implement object detection and tracking from video |
| III | Perform Feature extraction using RANSAC |
| | Perform Colorization |
| IV | Perform Text detection and recognition |
| | Perform Image matting and Composting |

| **Books and References:** | | | | | |
|---|---|---|---|---|---|
| Sr. No. | Title | Author/s | Publisher | Edition | Year |
| 1. | Computer Vision: Algorithms and Applications | Richard Szeliski | Springer | 1st Edition | 2010 |

**Course Outcomes:**

After completion of the course, a student should be able to:
OC 1: Understand the basics of computer vision
OC 2: Understand and analyse various structure form motion and various estimates of Dense Motion
OC 3: Apply various motion models to images and understand computation photography techniques
OC 4: Apply Epipolar geometry , Rectification and various other 3D correspondence and Stereo reconstruction techniques
OC 5: Understand image-based rendering and reconstruction.

(to be implemented in a cloud environment.)

## Malware Analysis Practical List Annexure I

| List of Practical: | |
|---|---|
| 1. | a. Files: *Lab01-01.exe* and *Lab01-01.dll*. |
| | i. Upload the files to *http://www.VirusTotal.com/* and view the reports. Does either file match any existing antivirus signatures? |
| | ii. When were these files compiled? |
| | iii. Are there any indications that either of these files is packed or obfuscated? If so, what are these indicators? |
| | iv. Do any imports hint at what this malware does? If so, which imports are they? |
| | v. Are there any other files or host-based indicators that you could look for on infected systems? |
| | vi. What network-based indicators could be used to find this malware on infected machines? |
| | vii. What would you guess is the purpose of these files? |
| | b. Analyze the file *Lab01-02.exe*. |
| | i. Upload the *Lab01-02.exe* file to *http://www.VirusTotal.com/*. Does it match any existing antivirus definitions? |
| | ii. Are there any indications that this file is packed or obfuscated? If so, what are these indicators? If the file is packed, unpack it if possible. |
| | iii. Do any imports hint at this program's functionality? If so, which imports are they and what do they tell you? |
| | iv. What host- or network-based indicators could be used to identify this malware on infected machines? |
| | c. Analyze the file Lab01-03.exe. |
| | i. Upload the *Lab01-03.exe* file to *http://www.VirusTotal.com/*. Does it match any existing antivirus definitions? |
| | ii. Are there any indications that this file is packed or obfuscated? If so, what are these indicators? If the file is packed, unpack it if possible. |
| | iii. Do any imports hint at this program's functionality? If so, which imports are they and what do they tell you? |
| | iv. What host- or network-based indicators could be used to identify this malware on infected machines? |
| | d. Analyze the file Lab01-04.exe. |
| | i. Upload the *Lab01-04.exe* file to *http://www.VirusTotal.com/*. Does it match any existing antivirus definitions? |
| | ii. Are there any indications that this file is packed or obfuscated? If so, what are these indicators? If the file is packed, unpack it if possible. |
| | iii. When was this program compiled? |
| | iv. Do any imports hint at this program's functionality? If so, which imports are they and what do they tell you? |
| | v. What host- or network-based indicators could be used to identify this malware on infected machines? |
| | vi. This file has one resource in the resource section. Use Resource Hacker to examine that resource, and then use it to extract the resource. What can you learn from the resource? |

| | |
|---|---|
| | e. Analyze the malware found in the file Lab03-01.exe using basic dynamic analysis tools. |
| | i. What are this malware's imports and strings? |
| | ii. What are the malware's host-based indicators? |
| | iii. Are there any useful network-based signatures for this malware? If so, what are they? |
| | f. Analyze the malware found in the file Lab03-02.dll using basic dynamic analysis tools. |
| | i. How can you get this malware to install itself? |
| | ii. How would you get this malware to run after installation? |
| | iii. How can you find the process under which this malware is running? |
| | iv. Which filters could you set in order to use procmon to glean information? |
| | v. What are the malware's host-based indicators? |
| | vi. Are there any useful network-based signatures for this malware? |
| | g. Execute the malware found in the file Lab03-03.exe while monitoring it using basic dynamic analysis tools in a safe environment |
| | i. What do you notice when monitoring this malware with Process Explorer? |
| | ii. Can you identify any live memory modifications? |
| | iii. What are the malware's host-based indicators? |
| | iv. What is the purpose of this program? |
| | h. Analyze the malware found in the file Lab03-04.exe using basic dynamic analysis tools. |
| | i. What happens when you run this file? |
| | ii. What is causing the roadblock in dynamic analysis? |
| | iii. Are there other ways to run this program? |
| | |
| 2. | a. Analyze the malware found in the file Lab05-01.dll using only IDA Pro. The goal of this lab is to give you hands-on experience with IDA Pro. If you've already worked with IDA Pro, you may choose to ignore these questions and focus on reverse-engineering the malware. |
| | i. What is the address of DllMain? |
| | ii. Use the Imports window to browse to gethostbyname. Where is the import located? |
| | iii. How many functions call gethostbyname? |
| | iv. Focusing on the call to gethostbynamelocated at 0x10001757, can you fig- ure out which DNS request will be made? |
| | v. How many local variables has IDA Pro recognized for the subroutine at 0x10001656? |
| | vi. How many parameters has IDA Pro recognized for the subroutine at 0x10001656? |
| | vii. Use the Strings window to locate the string \cmd.exe /cin the disassembly. Where is it located? |
| | viii. What is happening in the area of code that references \cmd.exe/c? |
| | ix. In the same area, at 0x100101C8, it looks like dword_1008E5C4 is a global variable that helps decide which path to take. How does the malware set dword_1008E5C4? (Hint: Use dword_1008E5C4's cross-references.) |
| | x. A few hundred lines into the subroutine at 0x1000FF58, a series of com- parisons use memcmpto compare strings. What happens if the string compar- ison to robotwork is successful (when memcmpreturns 0)? |
| | xi. What does the export PSLISTdo? |
| | xii. Use the graph mode to graph the cross-references from sub_10004E79. Which API functions could be called by entering this function? Based on the API functions alone, what could you rename this function? |
| | xiii. How many Windows API functions does DllMaincall directly? How many at a depth of 2? |
| | xiv. At 0x10001358, there is a call to Sleep (an API function that takes one parameter containing the number of milliseconds to sleep). Looking backward through the code, how long will the program sleep if this code executes? |
| | xv. At 0x10001701 is a call to socket. What are the three parameters? |
| | xvi. Using the MSDN page for socketand the named symbolic constants func- tionality in IDA Pro, can you make the parameters more meaningful? What are the parameters after you apply changes? |
| | xvii. Search for usage of the in instruction (opcode 0xED). This instruction is used with a magic string VMXh to perform VMware detection. Is that in use in this malware? Using the cross- |

| | |
|---|---|
| | references to the function that executes the in instruction, is there further evidence of VMware detection? |
| | xviii. Jump your cursor to 0x1001D988. What do you find? |
| | xix. If you have the IDA Python plug-in installed (included with the com- mercial version of IDA Pro), run *Lab05-01.py*, an IDA Pro Python script provided with the malware for this book. (Make sure the cursor is at 0x1001D988.) What happens after you run the script? |
| | xx. With the cursor in the same location, how do you turn this data into a single ASCII string? |
| | xxi. Open the script with a text editor. How does it work? |
| | b. analyze the malware found in the file Lab06-01.exe. |
| | i. What is the major code construct found in the only subroutine called by main? |
| | ii. What is the subroutine located at0x40105F? |
| | iii. What is the purpose of this program? |
| | c. Analyze the malware found in the file Lab06-02.exe. |
| | i. What operation does the first subroutine called by mainperform? |
| | ii. What is the subroutine located at0x40117F? |
| | iii. What does the second subroutine called by maindo? |
| | iv. What type of code construct is used in this subroutine? |
| | v. Are there any network-based indicators for thisprogram? |
| | vi. What is the purpose of this malware? |
| | d. analyze the malware found in the file Lab06-03.exe. |
| | i. Compare the calls in mainto Lab 6-2's mainmethod. What is the new function called from main? |
| | ii. What parameters does this new function take? |
| | iii. What major code construct does this function contain? |
| | iv. What can this function do? |
| | v. Are there any host-based indicators for this malware? |
| | vi. What is the purpose of this malware? |
| | e. analyze the malware found in the file Lab06-04.exe. |
| | i. What is the difference between the calls made from the main method in Labs 6-3 and 6-4? |
| | ii. What new code construct has been added to main? |
| | iii. What is the difference between this lab's parse HTML function and those of the previous labs? |
| | iv. How long will this program run? (Assume that it is connected to the Internet.) |
| | v. Are there any new network-based indicators for this malware? |
| | vi. What is the purpose of this malware? |
| 3. | a. Analyze the malware found in the file Lab07-01.exe. |
| | i. How does this program ensure that it continues running (achieves per- sistence) when the computer is restarted? |
| | ii. Why does this program use a mutex? |
| | iii. What is a good host-based signature to use for detecting this program? |
| | iv. What is a good network-based signature for detecting this malware? |
| | v. What is the purpose of this program? |
| | vi. When will this program finish executing? |
| | b. Analyze the malware found in the file Lab07-02.exe. |
| | i. How does this program achieve persistence? |
| | ii. What is the purpose of this program? |
| | iii. When will this program finish executing? |
| | c. For this lab, we obtained the malicious executable, Lab07-03.exe, and DLL, Lab07-03.dll, prior to executing. This is important to note because the mal- ware might change once it runs. Both files were found in the same directory on the victim machine. If you run the program, you should ensure that both files are in the same directory on the analysis machine. A visible IP string beginning with 127 (a loopback address) connects to the local machine. (In the real version of this malware, this address connects to a remote machine, but we've set it to connect to localhost to protect you.) |
| | i. How does this program achieve persistence to ensure that it continues running when the computer is restarted? |

| | |
|---|---|
| | ii. What are two good host-based signatures for this malware? |
| | iii. What is the purpose of this program? |
| | iv. How could you remove this malware once it is installed? |
| | d. Analyze the malware found in the file Lab09-01.exe using OllyDbg and IDA Pro to answer the following questions. This malware was initially analyzed in the Chapter 3 labs using basic static and dynamic analysis techniques. |
| | i. How can you get this malware to install itself? |
| | ii. What are the command-line options for this program? What is the pass- word requirement? |
| | iii. How can you use OllyDbg to permanently patch this malware, so that it doesn't require the special command-line password? |
| | iv. What are the host-based indicators of this malware? |
| | v. What are the different actions this malware can be instructed to take via the network? |
| | vi. Are there any useful network-based signatures for this malware? |
| | e. Analyze the malware found in the file Lab09-02.exe using OllyDbg to answer the following questions. |
| | i. What strings do you see statically in the binary? |
| | ii. What happens when you run this binary? |
| | iii. How can you get this sample to run its malicious payload? |
| | iv. What is happening at 0x00401133? |
| | v. What arguments are being passed to subroutine 0x00401089? |
| | vi. What domain name does this malware use? |
| | vii. What encoding routine is being used to obfuscate the domain name? |
| | viii. What is the significance of the CreateProcessA call at 0x0040106E? |
| | f. Analyze the malware found in the file Lab09-03.exe using OllyDbg and IDA Pro. This malware loads three included DLLs (DLL1.dll, DLL2.dll, and DLL3.dll ) that are all built to request the same memory load location. Therefore, when viewing these DLLs in OllyDbg versus IDA Pro, code may appear at different memory locations. The purpose of this lab is to make you comfortable with finding the correct location of code within IDA Pro when you are looking at code in OllyDbg |
| | i. What DLLs are imported by *Lab09-03.exe*? |
| | ii. What is the base address requested by *DLL1.dll*, *DLL2.dll*, and *DLL3.dll*? |
| | iii. When you use OllyDbg to debug *Lab09-03.exe*, what is the assigned based address for: *DLL1.dll*, *DLL2.dll*, and *DLL3.dll*? |
| | iv. When *Lab09-03.exe* calls an import function from *DLL1.dll*, what does this import function do? |
| | v. When *Lab09-03.exe* calls WriteFile, what is the filename it writes to? |
| | vi. When *Lab09-03.exe* creates a job using NetScheduleJobAdd, where does it get the data for the second parameter? |
| | vii. While running or debugging the program, you will see that it prints out three pieces of mystery data. What are the following: DLL 1 mystery data 1, DLL 2 mystery data 2, and DLL 3 mystery data 3? |
| | viii. How can you load *DLL2.dll* into IDA Pro so that it matches the load address used by OllyDbg? |
| | |
| 4. | a. This lab includes both a driver and an executable. You can run the executable from anywhere, but in order for the program to work properly, the driver must be placed in the C:\Windows\ System32 directory where it was origi- nally found on the victim computer. The executable is Lab10-01.exe, and the driver is Lab10-01.sys. |
| | i. Does this program make any direct changes to the registry? (Use procmon to check.) |
| | ii. The user-space program calls the ControlService function. Can you set a breakpoint with WinDbg to see what is executed in the kernel as a result of the call to ControlService? |
| | iii. What does this program do? |
| | b. The file for this lab is Lab10-02.exe. |
| | i. Does this program create any files? If so, what are they? |
| | ii. Does this program have a kernel component? |
| | iii. What does this program do? |

| | |
|---|---|
| | c. This lab includes a driver and an executable. You can run the executable from anywhere, but in order for the program to work properly, the driver must be placed in the C:\Windows\System32 directory where it was originally found on the victim computer. The executable is Lab10-03.exe, and the driver is Lab10-03.sys. |
| | i. What does this program do? |
| | ii. Once this program is running, how do you stop it? |
| | iii. What does the kernel component do? |
| 5. | a. Analyze the malware found in Lab11-01.exe |
| | i. What does the malware drop to disk? |
| | ii. How does the malware achieve persistence? |
| | iii. How does the malware steal user credentials? |
| | iv. What does the malware do with stolen credentials? |
| | v. How can you use this malware to get user credentials from your test environment? |
| | b. Analyze the malware found in *Lab11-02.dll*. Assume that a suspicious file named *Lab11-02.ini* was also found with this malware. |
| | i. What are the exports for this DLL malware? |
| | ii. What happens after you attempt to install this malware using |
| | iii. *rundll32.exe*? |
| | iv. Where must *Lab11-02.ini* reside in order for the malware to install properly? |
| | v. How is this malware installed for persistence? |
| | vi. What user-space rootkit technique does this malware employ? |
| | vii. What does the hooking code do? |
| | viii. Which process(es) does this malware attack and why? |
| | ix. What is the significance of the *.ini* file? |
| | c. Analyze the malware found in *Lab11-03.exe* and *Lab11-03.dll*. Make sure that both files are in the same directory during analysis |
| | i. What interesting analysis leads can you discover using basic static analysis? |
| | ii. What happens when you run this malware? |
| | iii. How does *Lab11-03.exe* persistently install *Lab11-03.dll*? |
| | iv. Which Windows system file does the malware infect? |
| | v. What does *Lab11-03.dll* do? |
| | vi. Where does the malware store the data it collects? |
| 6. | a. Analyze the malware found in the file *Lab12-01.exe* and *Lab12-01.dll*. Make sure that these files are in the same directory when performing the analysis. |
| | i. What happens when you run the malware executable? |
| | ii. What process is being injected? |
| | iii. How can you make the malware stop the pop-ups? |
| | iv. How does this malware operate? |
| | b. Analyze the malware found in the file *Lab12-02.exe*. |
| | i. What is the purpose of this program? |
| | ii. How does the launcher program hide execution? |
| | iii. Where is the malicious payload stored? |
| | iv. How is the malicious payload protected? |
| | v. How are strings protected? |
| | c. Analyze the malware extracted during the analysis of Lab 12-2, or use the file *Lab12-03.exe*. |
| | i. What is the purpose of this malicious payload? |
| | ii. How does the malicious payload inject itself? |
| | iii. What filesystem residue does this program create? |
| | d. Analyze the malware found in the file *Lab12-04.exe*. |
| | i. What does the code at 0x401000 accomplish? |
| | ii. Which process has code injected? |
| | iii. What DLL is loaded using LoadLibraryA? |
| | iv. What is the fourth argument passed to the CreateRemoteThread call? |
| | v. What malware is dropped by the main executable? |
| 7. | a. Analyze the malware found in the file *Lab13-01.exe*. |

| | |
|---|---|
| | i. Compare the strings in the malware (from the output of the stringscommand) with the information available via dynamic analysis. Based on this comparison, which elements might be encoded? |
| | ii. Use IDA Pro to look for potential encoding by searching for the string xor. What type of encoding do you find? |
| | iii. What is the key used for encoding and what content does it encode? |
| | iv. Use the static tools FindCrypt2, Krypto ANALyzer (KANAL), and the IDA Entropy Plugin to identify any other encoding mechanisms. What do you find? |
| | v. What type of encoding is used for a portion of the network traffic sent by the malware? |
| | vi. Where is the Base64 function in the disassembly? |
| | vii. What is the maximum length of the Base64-encoded data that is sent? What is encoded? |
| | viii. In this malware, would you ever see the padding characters (=or ==) in the Base64-encoded data? |
| | ix. What does this malware do? |
| | b. Analyze the malware found in the file *Lab13-02.exe*. |
| | i. Using dynamic analysis, determine what this malware creates. |
| | ii. Use static techniques such as an xor search, FindCrypt2, KANAL, and the IDA Entropy Plugin to look for potential encoding. What do you find? |
| | iii. Based on your answer to question 1, which imported function would be a good prospect for finding the encoding functions? |
| | iv. Where is the encoding function in thedisassembly? |
| | v. Trace from the encoding function to the source of the encoded content. What is the content? |
| | vi. Can you find the algorithm used for encoding? If not, how can you decode the content? |
| | vii. Using instrumentation, can you recover the original source of one of the encoded files? |
| | c. Analyze the malware found in the file *Lab13-03.exe*. |
| | i. Compare the output of strings with the information available via dynamic analysis. Based on this comparison, which elements might be encoded? |
| | ii. Use static analysis to look for potential encoding by searching for the string xor. What type of encoding do you find? |
| | iii. Use static tools like FindCrypt2, KANAL, and the IDA Entropy Plugin to identify any other encoding mechanisms. How do these findings com- pare with the XOR findings? |
| | iv. Which two encoding techniques are used in thismalware? |
| | v. For each encoding technique, what is thekey? |
| | vi. For the cryptographic encryption algorithm, is the key sufficient? What else must be known? |
| | vii. What does this malware do? |
| | viii. Create code to decrypt some of the content produced during dynamic analysis. What is this content? |
| 8. | a. Analyze the malware found in file *Lab14-01.exe*. This program is not harmful to your system. |
| | i. Which networking libraries does the malware use, and what are their advantages? |
| | ii. What source elements are used to construct the networking beacon, and what conditions would cause the beacon to change? |
| | iii. Why might the information embedded in the networking beacon be of interest to the attacker? |
| | iv. Does the malware use standard Base64 encoding? If not, how is the encoding unusual? |
| | v. What is the overall purpose of this malware? |
| | vi. What elements of the malware's communication may be effectively detected using a network signature? |
| | vii. What mistakes might analysts make in trying to develop a signature for this malware? |
| | viii. What set of signatures would detect this malware (and future variants)? |
| | b. Analyze the malware found in file *Lab14-02.exe*. This malware has been configured to beacon to a hard-coded loopback address in order to prevent it from harming your system, but imagine that it is a hard-coded external address. |
| | i. What are the advantages or disadvantages of coding malware to use direct IP addresses? |

| | |
|---|---|
| | ii. Which networking libraries does this malware use? What are the advantages or disadvantages of using these libraries? |
| | iii. What is the source of the URL that the malware uses for beaconing? What advantages does this source offer? |
| | iv. Which aspect of the HTTP protocol does the malware leverage to achieve its objectives? |
| | v. What kind of information is communicated in the malware's initial beacon? |
| | vi. What are some disadvantages in the design of this malware's communication channels? |
| | vii. Is the malware's encoding scheme standard? |
| | viii. How is communication terminated? |
| | ix. What is the purpose of this malware, and what role might it play in the attacker's arsenal? |
| | c. This lab builds on Practical 8 a. Imagine that this malware is an attempt by the attacker to improve his techniques. Analyze the malware found in file *Lab14-03.exe*. |
| | i. What hard-coded elements are used in the initial beacon? What elements, if any, would make a good signature? |
| | ii. What elements of the initial beacon may not be conducive to a longlasting signature? |
| | iii. How does the malware obtain commands? What example from the chapter used a similar methodology? What are the advantages of this technique? |
| | iv. When the malware receives input, what checks are performed on the input to determine whether it is a valid command? How does the attacker hide the list of commands the malware is searching for? |
| | v. What type of encoding is used for command arguments? How is it different from Base64, and what advantages or disadvantages does it offer? |
| | vi. What commands are available to this malware? |
| | vii. What is the purpose of this malware? |
| | viii. This chapter introduced the idea of targeting different areas of code with independent signatures (where possible) in order to add resiliency to network indicators. What are some distinct areas of code or configuration data that can be targeted by network signatures? |
| | ix. What set of signatures should be used for this malware? |
| | d. Analyze the sample found in the file Lab15-01.exe. This is a command-line program that takes an argument and prints "Good Job!" if the argument matches a secret code. |
| | i. What anti-disassembly technique is used in this binary? |
| | ii. What rogue opcode is the disassembly tricked into disassembling? |
| | iii. How many times is this technique used? |
| | iv. What command-line argument will cause the program to print "Good Job!"? |
| | e. Analyze the malware found in the file Lab15-02.exe. Correct all anti-disassembly countermeasures before analyzing the binary in order to answer the questions. |
| | i. What URL is initially requested by the program? |
| | ii. How is the User-Agent generated? |
| | iii. What does the program look for in the page it initially requests? |
| | iv. What does the program do with the information it extracts from the page? |
| | f. Analyze the malware found in the file *Lab15-03.exe*. At first glance, this binary appears to be a legitimate tool, but it actually contains more functionality than advertised. |
| | i. How is the malicious code initially called? |
| | ii. What does the malicious code do? |
| | iii. What URL does the malware use? |
| | iv. What filename does the malware use? |
| 9. | a. Analyze the malware found in *Lab16-01.exe* using a debugger. This is the same malware as *Lab09-01.exe*, with added anti-debugging techniques. |
| | i. Which anti-debugging techniques does this malware employ? |
| | ii. What happens when each anti-debugging technique succeeds? |
| | iii. How can you get around these anti-debugging techniques? |
| | iv. How do you manually change the structures checked during runtime? |
| | v. Which OllyDbg plug-in will protect you from the anti-debugging tech- niques used by this malware? |

| | |
|---|---|
| | b. Analyze the malware found in *Lab16-02.exe* using a debugger. The goal of this lab is to figure out the correct password. The malware does not drop a mali- cious payload. |
| | i. What happens when you run *Lab16-02.exe* from the command line? |
| | ii. What happens when you run *Lab16-02.exe* and guess the command-line parameter? |
| | iii. What is the command-line password? |
| | iv. Load *Lab16-02.exe* into IDA Pro. Where in the mainfunction is strncmp |
| | v. found? |
| | vi. What happens when you load this malware into OllyDbg using the default settings? |
| | vii. What is unique about the PE structure of *Lab16-02.exe*? |
| | viii. Where is the callback located? (Hint: Use CTRL-E in IDA Pro.) |
| | ix. Which anti-debugging technique is the program using to terminate immediately in the debugger and how can you avoid this check? |
| | x. What is the command-line password you see in the debugger after you disable the anti-debugging technique? |
| | xi. Does the password found in the debugger work on the command line? |
| | c. Analyze the malware in *Lab16-03.exe* using a debugger. This malware is similar to *Lab09-02.exe*, with certain modifications, including the introduction of anti-debugging techniques. |
| | i. Which strings do you see when using static analysis on the binary? |
| | ii. What happens when you run this binary? |
| | iii. How must you rename the sample in order for it to runproperly? |
| | iv. Which anti-debugging techniques does this malwareemploy? |
| | v. For each technique, what does the malware do if it determines it is running in a debugger? |
| | vi. Why are the anti-debugging techniques successful in this malware? |
| | vii. What domain name does this malware use? |
| | d. Analyze the malware found in *Lab17-01.exe* inside VMware. This is the same malware as *Lab07-01.exe*, with added anti-VMware techniques. |
| | i. What anti-VM techniques does this malware use? |
| | ii. If you have the commercial version of IDA Pro, run the IDA Python script from Listing 17-4 in Chapter 17 (provided here as *findAntiVM.py*). What does it find? |
| | iii. What happens when each anti-VM technique succeeds? |
| | iv. Which of these anti-VM techniques work against your virtual machine? |
| | v. Why does each anti-VM technique work or fail? |
| | vi. How could you disable these anti-VM techniques and get the malware to run? |
| | e. Analyze the malware found in the file *Lab17-02.dll* inside VMware. After answering the first question in this lab, try to run the installation exports using *rundll32.exe* and monitor them with a tool like procmon. The following is an example command line for executing the DLL:<br>rundll32.exe Lab17-02.dll,InstallRT (or InstallSA/InstallSB) |
| | i. What are the exports for this DLL? |
| | ii. What happens after the attempted installation using *rundll32.exe*? |
| | iii. Which files are created and what do theycontain? |
| | iv. What method of anti-VM is in use? |
| | v. How could you force the malware to install during runtime? |
| | vi. How could you permanently disable the anti-VMtechnique? |
| | vii. How does each installation export function work? |
| | f. Analyze the malware *Lab17-03.exe* inside VMware. |
| | i. What happens when you run this malware in a virtual machine? |
| | ii. How could you get this malware to run and drop its keylogger? |
| | iii. Which anti-VM techniques does this malware use? |
| | iv. What system changes could you make to permanently avoid the anti-VM techniques used by this malware? |
| | v. How could you patch the binary in OllyDbg to force the anti-VM techniques to permanently fail? |

| | |
|---|---|
| 10. | a. Analyze the file *Lab19-01.bin* using *shellcode_launcher.exe* |
| | i. How is the shellcode encoded? |
| | ii. Which functions does the shellcode manually import? |
| | iii. What network host does the shellcode communicate with? |
| | iv. What filesystem residue does the shellcode leave? |
| | v. What does the shellcode do? |
| | b. The file *Lab19-02.exe* contains a piece of shellcode that will be injected into another process and run. Analyze this file. |
| | i. What process is injected with the shellcode? |
| | ii. Where is the shellcode located? |
| | iii. How is the shellcode encoded? |
| | iv. Which functions does the shellcode manually import? |
| | v. What network hosts does the shellcode communicate with? |
| | vi. What does the shellcode do? |
| | c. Analyze the file *Lab19-03.pdf*. If you get stuck and can't find the shellcode, just skip that part of the lab and analyze file *Lab19-03_sc.bin* using *shellcode_launcher.exe*. |
| | i. What exploit is used in this PDF? |
| | ii. How is the shellcode encoded? |
| | iii. Which functions does the shellcode manually import? |
| | iv. What filesystem residue does the shellcode leave? |
| | v. What does the shellcode do? |
| | d. The purpose of this first lab is to demonstrate the usage of the thispointer. Analyze the malware in *Lab20-01.exe*. |
| | i. Does the function at 0x401040 take any parameters? |
| | ii. Which URL is used in the call to URLDownloadToFile? |
| | iii. What does this program do? |
| | e. Analyze the malware In Lab20-02.exe. |
| | i. What can you learn from the interesting strings in this program? |
| | ii. What do the imports tell you about this program? |
| | iii. What is the purpose of the object created at 0x4011D9? Does it have any virtual functions? |
| | iv. Which functions could possibly be called by the call [edx]instruction at 0x401349? |
| | v. How could you easily set up the server that this malware expects in order to fully analyze the malware without connecting it to the Internet? |
| | vi. What is the purpose of this program? |
| | vii. What is the purpose of implementing a virtual function call in this program? |
| | f. Analyze the malware in Lab20-03.exe. |
| | i. What can you learn from the interesting strings in this program? |
| | ii. What do the imports tell you about this program? |
| | iii. At 0x4036F0, there is a function call that takes the string Config error, followed a few instructions later by a call to CxxThrowException. Does the function take any parameters other than the string? Does the function return anything? What can you tell about this function from the context in which it's used? |
| | iv. What do the six entries in the switch table at 0x4025C8 do? |
| | v. What is the purpose of this program? |
| | g. Analyze the code in *Lab21-01.exe* |
| | i. What happens when you run this program without any parameters? |
| | ii. Depending on your version of IDA Pro, main may not be recognized automatically. How can you identify the call to the main function? |
| | iii. What is being stored on the stack in the instructions from 0x0000000140001150 to 0x0000000140001161? |
| | iv. How can you get this program to run its payload without changing the filename of the executable? |
| | v. Which two strings are being compared by the call to strncmp at 0x0000000140001205? |
| | vi. Does the function at 0x00000001400013C8 take any parameters? |

| | |
|---|---|
| | vii. How many arguments are passed to the call to CreateProcess at 0x0000000140001093? How do you know? |
| | h. Analyze the malware found in *Lab21-02.exe* on both x86 and x64 virtual machines. |
| | i. What is interesting about the malware's resource sections? |
| | ii. Is this malware compiled for x64 or x86? |
| | iii. How does the malware determine the type of environment in which it is running? |
| | iv. What does this malware do differently in an x64 environment versus an x86 environment? |
| | v. Which files does the malware drop when running on an x86 machine? Where would you find the file or files? |
| | vi. Which files does the malware drop when running on an x64 machine? Where would you find the file or files? |
| | vii. What type of process does the malware launch when run on an x64 system? |
| | viii. What does the malware do? |

# Evaluation Scheme

**Theory courses of 4 credits:** Total marks 100. Out of the total, 50 % each for internal and external evaluation.

A. **Internal Evaluation (30m + 10m + 10m = 50 Marks )**

The internal assessment marks shall be awarded as follows:

1. **30 marks (Any one of the following):**
   a. Written Test of 30 Marks
   b. SWAYAM (Advanced Course) of minimum 20 hours and certification exam completed or
   c. NPTEL (Advanced Course) of minimum 20 hours and certification exam completed or
   d. Valid International Certifications (Prometric, Pearson, Certiport, Coursera, Udemy and the like)
   e. Certification marks of one completed exam shall be awarded to one course only. For four courses, the students will have to complete four certifications.
   (Note: Only those certification/courses suggested by the department shall be deemed valid,
   Student cannot do any certification on their own)
2. **10 marks**
   10 marks from every course (Two 4 credits mandatory courses, one 2 credits mandatory course, one 4 credits elective course) coming to a total of 40 marks, shall be awarded on publishing of research paper in UGC approved / Other Journal with plagiarism less than 15%. The marks can be awarded as per the impact factor of the journal, quality of the paper, importance of the contents published, social value.

3. **10 marks**
   Open Book examination based on problem solving related to the respective subject.

i. **Suggested format of Question paper of 30 marks for the written test.**

| Q1. | Attempt *any two* of the following: | 16 marks |
|---|---|---|
| a. | | |
| b. | | |
| c. | | |
| d. | | |
| | | |
| Q2. | Attempt *any two* of the following: | 14 marks |
| a. | | |
| b. | | |
| c. | | |
| d. | | |

B. **External Examination: (50 marks) Duration : 2 hrs**

| | | |
|---|---|---|
| | All questions are compulsory | |
| Q1 | (Based on all units) Attempt *any two* of the following: | 10 marks |
| a. | Unit 1 | |
| b. | Unit 2 | |
| c. | Unit 3 | |
| d. | Unit 4 | |
| | | |
| Q2 | (Based on Unit 1) Attempt *any two* of the following: | 10 marks |
| Q3 | (Based on Unit 2) Attempt *any two* of the following: | 10 marks |
| Q4 | (Based on Unit 3) Attempt *any two* of the following: | 10 marks |
| Q5 | (Based on Unit 4) Attempt *any two* of the following: | 10 marks |

**Theory courses of 2 credits:** Total marks 50. Out of the total, 50 % each for internal and external evaluation.

A. **Internal Evaluation (25 Marks)**

The internal assessment marks shall be awarded as follows:

1. 10 marks from every course (Two 4 credits mandatory courses, One 2 credits mandatory course, One 4 credits elective course) coming to a total of 40 marks, shall be awarded on publishing of research paper in UGC approved / Other Journal with plagiarism less than 15%. The marks can be awarded as per the impact factor of the journal, quality of the paper, importance of the contents published, social value.
2. 10 marks - Open Book examination based on problem solving related to the respective subject.
3. 5 marks - Assignment/Group discussion.

B. **External Examination: (25 marks) Duration : 1 hr**

| | | |
|---|---|---|
| | All questions are compulsory | |
| Q1 | (Based on Unit 1) Attempt *any two* of the following: | 13 marks |
| Q2 | (Based on Unit 2) Attempt *any two* of the following: | 12 marks |

**Practical courses of 2 credits:** Total marks 50. Out of the total, 50 % each for internal and external evaluation.

A. **Practical Evaluation Internal (25 marks)**

| | | |
|---|---|---|
| 1. | Performance during all practical sessions | 10 |
| 2. | Problem solving with the acquired programming skills | 10 |
| 3. | Viva Voce | 5 |

B. **Practical Evaluation External (25 marks)**

A Certified copy of hard-bound journal is essential to appear for the practical examination.

| | | |
|---|---|---|
| 1. | Practical Question | 15 |
| 2. | Journal | 5 |
| 3. | Viva Voce | 5 |

## Letter Grades and Grade Points

| **Semester GPA/Program CGPA Semester/Program** | **Percentage of Marks** | **Alpha-Sign/Letter Grade Result** |
|---|---|---|
| 9.00 – 10.00 | 90.00-100.00 | O (Outstanding) |
| 8.00 -<9.00 | 80.00-<90.00 | A+ (Excellent) |
| 7.00-<8.00 | 70.00-<80.00 | A (Very Good) |
| 6.00-<7.00 | 60.00-<70.00 | B+ (Good) |
| 5.50-<6.00 | 55.00-<60.00 | B (Above Average) |
| 5.00-<5.50 | 50.00-<55.00 | C (Average) |
| 4.00-<5.00 | 40.00-<50.00 | P (Pass) |
| Below 4.00 | Below 40.00 | F (Fail) |
| Ab(Absent) | - | Absent |

**Sign of HOD**
Dr. Mrs. R. Srivaramangai
Dept of Information Technology

## Team for Creation of Syllabus

| Name | Organization | Sign |
|---|---|---|
| Dr. Mrs. R. Srivaramangai | Dept of Information Technology<br>Head, UDIT | Originally Signed |
| Dr. Hiren Dand | Head, Dept of Information Technology<br>Mulund College of Commerce | Dand |
| Dr. Rajendra Patil | Principal, Anna Leela Bunt's College | Patil |
| Mr. Mandar Bhave | Head, Dept of Information Technology & Computer Science<br>D.G. Ruparel College<br>(Special Invitee) | mandar |

**Sign of HOD**
Dr. Mrs. R. Srivaramangai
Dept of Information Technology

Garje
**Sign of Dean**
Prof. Shivram Garje
Science & Technology

**Appendix B**

## Justification for (M.Sc (Information Technology))

| | | |
|---|---|---|
| 1. | Necessity for starting the course: | A M.Sc(Information Technology) is a specialized postgraduate degree program that offers in-depth knowledge and expertise in various aspects of information technology. The key factors are advancements in technology, Specialization and Depth of Knowledge, career advancement, research and innovation etc |
| 2. | Whether the UGC has recommended the course: | Yes |
| 3. | Whether all the courses have commenced from the academic year 2023-24 | The program has commenced from 2004-2005 academic year onwards |
| 4. | The courses started by the University are self-financed, whether adequate number of eligible permanent faculties are available?: | Yes. |
| 5. | To give details regarding the duration of the Course and is it possible to compress the course?: | 2 years. Not possible to compress the program |
| 6. | The intake capacity of each course and no. of admissions given in the current academic year: | 20 seats minimum given. Seats capacity varies from college to college as per the sanction given. 2023-2024 admission is yet to start |
| 7. | Opportunities of Employability / Employment available after undertaking these courses: | The employability prospects for individuals with an M.Sc(Information Technology) are highly favorable due to the growing reliance on technology in various industries. Graduates with an M.Sc(Information Technology possess specialized knowledge and skills that make them attractive to a wide range of employers. The versatility of an M.Sc(Information Technology allows graduates to pursue various career paths across diverse industries, including finance, healthcare, e-commerce, education, government, and more. As technology continues to advance, the demand for skilled IT professionals with an M.Sc degree is expected to grow, making M.Sc(Information Technology graduates highly sought after in the job market. |

**Sign of HOD**
Dr. Mrs. R. Srivaramangai
Dept of Information Technology

**Sign of Dean**
Prof. Shivram Garje
Science & Technology