

Student Project Seed Grant

Title:

**CyberShield:**

**Securing Websites with Embedded Electronic Ecosystem**

**Department of Electronics and Telecommunication Engineering**

1. **Title of the Project:**

CyberShield: Securing Websites with Embedded Electronic Ecosystem

1. **Broad Area of Research:**
2. Embedded Systems
3. Cyber Security
4. **Names of Students, Parent Department and USN**:
5. Aryan Singh – Electronics and Telecommunication (1MS20ET011)
6. Chaitra – Electronics and Telecommunication (1MS20ET014)
7. Chandan Kumar – Electronics and Telecommunication (1MS20ET015)
8. Pranayak Uniyal – Electronics and Telecommunication (1MS20ET035)
9. **Name of supervisor**

Dr S.G Shivaprasad Yadav (Associate Professor, Dept. of ETE, RIT)

1. **Expertise of supervisor in the domain of the proposed seed grant**:

Anil Kumar, Sanket S. Naik Dessai, S G Shivaprasad Yadav, “Design and implementation of an automated office environment system using embedded sensors”, International Journal of Reconfigurable and Embedded Systems (IJRES), Vol. 11, No. 1, March 2022, pp. 34~48, ISSN: 2089-4864,

DOI: 10.11591/ijres.v11.i1.pp34-48

### Abstract of proposal

### Vulnerability scanning is a security method used to identify security restrictions on a computer  or website. Vulnerability monitoring can be used byindividuals or network administrators for security purposes, or  by hackers trying to access unauthorized computers or websites. Scanning is the evaluation of potential vulnerabilities of computer networks to identify vul-nerabilities. Vulnerability scanning detects andclassifies weak functions in computers, networks and communication devices and estimate the effectiveness of protection.

### To address this challenge, this project introduces a novel hardware device centred around the Raspberry Pi 4, featuring an advanced terminal application dedicated to detecting and reporting vulnerabilities in websites. The primary goal of this project is to provide website administrators and developers with a comprehensive and efficient solution for identifying potential security weaknesses, enabling them to bolster their website defences proactively.

### 

### The hardware device integrates a selection of renowned security tools, including nmap, dnsrecon, wafw00f, uniscan, sslyze, fierce, lbd, theharvester, amass, nikto, and others, consolidating their functionalities into a unified framework. The terminal application leverages the capabilities of these tools to conduct in-depth vulnerability scans, targeting critical issues such as HEARTBLEED, FREAK, POODLE, CCS Injection, LOGJAM, OCSP Stapling, Commonly Opened Ports, hallow XSS, SQLi, and BSQLi Banners. By combining these powerful tools into a single entity, the hardware device enhances the efficiency and effectiveness of vulnerability assessment, ensuring comprehensive coverage of potential threats.

### Brief Introduction to the proposal

### The rapid digital transformation has revolutionized the way we interact with the world, relying extensively on web-based platforms for communication, commerce, and information access. However, this increasing dependence on the internet has also made web applications prime targets for malicious actors seeking to exploit vulnerabilities for their gain. Cyberattacks, data breaches, and security incidents have become commonplace, prompting a dire need for robust web application security measures.

### In light of these challenges, this project introduces an innovative hardware device built around the Raspberry Pi 4, a versatile single-board computer known for its processing power and versatility. The core feature of this device is a sophisticated terminal application designed to simplify and streamline the process of identifying and assessing vulnerabilities in websites.

### Rather than relying on separate, disparate security tools, the application combines a wide array of industry-leading software, such as nmap, dnsrecon, wafw00f, uniscan, sslyze, fierce, lbd, theharvester, amass, nikto, and more. The integrated terminal application brings together the functionalities of these tools, enabling website administrators and developers to conduct comprehensive vulnerability scans with ease. By scanning for critical issues like HEARTBLEED, FREAK, POODLE, CCS Injection, LOGJAM, OCSP Stapling, Commonly Opened Ports, hallow XSS, SQLi, and BSQLi Banners, the device empowers users to identify potential threats early on and implement appropriate security measures.

### The overarching objective of this project is to provide an all-encompassing solution that simplifies vulnerability assessment and assists website owners in securing their digital assets. With the hardware device and its terminal application, users can efficiently perform website scans, receive detailed vulnerability reports, and take proactive steps to safeguard their websites against potential cyber threats. By fostering a proactive security approach, this project aims to contribute to a safer and more resilient online ecosystem.

### Background (literature review)

### Prior research and existing literature emphasize the criticality of web application security due to the increasing frequency of cyberattacks and data breaches. Several studies have highlighted the significance of vulnerability scanning tools and methodologies in identifying potential threats to web-based systems. While individual security tools like nmap, dnsrecon, wafw00f, uniscan, sslyze, fierce, lbd, theharvester, amass, nikto, etc., have been widely used, their integration into a cohesive framework on a Raspberry Pi 4 device is a novel approach.

### Several studies have explored vulnerabilities such as HEARTBLEED, FREAK, POODLE, CCS Injection, LOGJAM, OCSP Stapling, Commonly Opened Ports, hallow XSS, SQLi, and BSQLi Banners, demonstrating the severity of their impact on web applications. However, existing solutions typically require manual execution of various tools, leading to inefficiencies and time-consuming procedures.

### This chapter introduces web attack research, countermeasures, black box web awareness and Nesses vulnerability detection tool Acunetix-Web Vulnerability Scanner, OWASP Zed Attack Proxy (ZAP), HTTP, Vulnerability Scanner. We will discuss some of these articles below.

### [1] Nessus

### The latest version of Nessus is 5.2.5 is one of the most popular electronic devices. It allows scanning of misconfigurations of software installed on the computer. It also includes checking the machine's open ports and the software version installed on the machine. It examines vulnerabilities that allow users to remotely control or access sensitive information on the system, denial of service against TCP/IP stacks, and PCI DSS inspection, among other things. This includes web application scanning; Like auditing SQL injection and cross-site scripting. Nessus has two versions: In-House Advertising and Professional Advertising Than Lee. For security reasons, Nessus scan results can be exported to various file formats such as HTML and CVS. As of 2010, 4,444 organizations use Nessus.

### 

### [2] Acunetix - Web Vulnerability Scanner

### Acunetix is web application vulnerability scanning. Web Vulnerability Scanner is a web application scanning tool that can detect vulnerability for example SQL injection, cross site scripting, flaws in the underlying operating system and misconfiguration of the web server. Acunetix also able to perform advanced penetration testing tool and testing for the password protected area. It also can detect port scanning. Acunetix runs on Windows operating system; the minimum is XP. For reporting purposes, Acunetix results can be export to PDF format file. Acunetix has come out with two different versions; Professional

### and trial version.

### [3] OWASP Zed Attack Proxy (ZAP)

### Zed Attack Proxy (ZAP) is a freeware vulnerability scanning tool. It was developed by Open Web Application Security Projector OWASP. OWASP ZAP Project or also known as Zed Attack Proxy is an integrated penetration testing tool for finding vulnerabilities in web applications. ZAP is an open-source tool that runs either on Linux or Windows platform. It also supports multiple languages, for example French, Spanish and Arabic. The example of the vulnerability that able to detect by OWASP ZAP is HTTP Parameter Pollution (HPP) extension and SQL injection.

### [4] Web Vulnerability Scanner by Using HTTP Method

### Web vulnerability scanner by using HTTP method basically works on URL crawling, Search engine, Remote Site, third party database and domain reputation. This vulnerability scanner scan URL and CMS. It scans for shells from client-side machine for commonly injected location and with their usual file names. It also checks mail server IP. Scan SQL injections for MySQL, MSSQL, PGSQL and Oracle database. It is trick that exploit poorly filtered or not correctly escaped SQL queries. It also scans XSS, Malware and directory indexing. but the vulnerabilities of this scanner seek to identify their efficiency in detecting different vulnerabilities.

### [5] SecuBat-A web vulnerability scanner

### SecuBat is a web application is used to find web vulnerability. Example of such vulnerability are SQL injection and cross site scripting(XSS). Using SecuBat identified a large number of potential vulnerable website. Also, SecuBat discover web vulnerabilities that could be used to launch phishing attacks that are difficult to identify even by technically more sophisticated users. SecuBat has crawling component to determine the door of attacks and four attacks are used.

### Form redirecting XSS attack

### SQL injection

### Simple reflected XSS attack

### Encoded Reflected XSS attack

### Problem Statement

### Securing data and implementing robust cybersecurity measures are of paramount importance in today's digital age. Our project aims to identify those critical issues or flaws that could be used by ill-intentioned people to exploit them.

### This system doesn’t only find the flaws but also creates a detailed report of critical level and measures to be taken to fix them.

### Objectives

* + Protection of Sensitive Information.
  + Data is a valuable asset, and its security is essential to protect sensitive information such as personal data, financial records, intellectual property, and trade secrets
  + Compliance and Legal Obligations
  + Various regulations and laws (e.g., GDPR, HIPAA, CCPA) require organizations to implement adequate security measures to protect user data.
  + Business Continuity and Reputation Management
  + A cyber incident or data breach can disrupt business operations, leading to downtime and financial losses. Additionally, a compromised reputation resulting from security incidents can erode customer trust and loyalty, impacting long-term business viability.
  + Prevention of Financial Losses
  + Cyberattacks and data breaches can result in significant financial losses, including costs associated with incident response, data recovery, legal expenses, and potential fines.

### Methodology of the proposed solution

### Creating a terminal application that scans vulnerabilities from a given website and produces a comprehensive report is a significant undertaking. To help you organize the development process, I'll outline a methodology for you:

### 

### Project Planning and Scope Definition:

### Define the specific objectives and goals of the terminal application. List the vulnerabilities and security issues you want to scan for. Determine the supported websites and their formats (HTTP, HTTPS). Define the tools you'll be using and their integration into the application.

### Research and Learning:

### Familiarize yourself with the Raspberry Pi 4 and its capabilities. Learn about Python 3 libraries and modules to interact with the tools (nmap, sslyze, nikto, etc.). Understand the vulnerabilities you plan to scan for and how to detect them.

### Environment Setup: Set up the Raspberry Pi 4 with the necessary dependencies and Python 3 environment. Install and configure the required tools like nmap, dnsrecon, wafw00f, etc.

### Design and Architecture:

### Plan the structure of your terminal application and its user interface. Create a flowchart or diagram to illustrate the interaction between the different tools. Design a report template to present the findings to the user.

### Coding and Implementation:

### Develop the terminal application using Python 3. Integrate the various tools into the application and define how they communicate. Implement functions to handle input validation, website scanning, and report generation.

### Testing:

### Perform unit testing for individual functions and modules. Conduct integration testing to ensure seamless interactions between the tools. Test the application on different websites and validate the accuracy of the results.

### Error Handling and Logging:

### Implement robust error handling to gracefully handle exceptions. Set up logging to record any issues or errors during the scanning process.

### User Interface (UI) Enhancement:

### Improve the user interface to make the application user-friendly. Add informative prompts, progress bars, and user instructions.

### Security Measures: Ensure that the terminal application itself is secure and free from vulnerabilities. Implement measures to protect user data and sensitive information.

### Documentation:

### Create detailed documentation on how to use the terminal application. Document the vulnerability scanning process, tools used, and their integration. Performance

### Optimization:

### Identify potential bottlenecks and optimize the application for speed and efficiency. Consider ways to reduce the scanning time and resource usage. Beta Testing and

### Feedback:

### Have a selected group of users test the application and provide feedback. Use the feedback to identify areas for improvement and bug fixing.

### Finalization and Deployment:

### Address any issues found during beta testing. Package the terminal application for easy installation and deployment. Make it available for download or distribution to the target audience.

### Maintenance and Updates:

### Plan for regular maintenance to keep the application up-to-date. Monitor the tools used and update them as new versions or security patches are released. Also, keep in mind the legal and ethical implications of scanning websites for vulnerabilities, and make sure you have proper authorization to conduct such scans.

### Preliminary Work Done (if any) and Project Execution Feasibility

### We have started implementing the related python code for taking the URL of website as input and performing operation on it. We are currently trying to run checks for all well-known bugs and implementing the already available relevant tools of security in Linux operating systems.

### Details of experimental set up

### Setting up the experimental environment for testing your terminal application that scans vulnerabilities from a given website requires careful preparation. Here's a detailed step-by-step guide for creating the experimental setup:

### Hardware Requirements:

### Raspberry Pi 4 (4GB or 8GB RAM model recommended) or higher.

### MicroSD card (16GB or more) with a compatible operating system installed (e.g., Raspbian).

### Power supply for Raspberry Pi.

### Keyboard, mouse, and HDMI cable for initial setup and configuration.

### Internet connection (Wi-Fi or Ethernet) for accessing websites.

### Software Requirements:

### Raspbian OS or any other compatible OS for Raspberry Pi.

### Python 3 (should be pre-installed on most Raspbian distributions).

### Required Python libraries and modules for interacting with the tools (e.g., subprocess, os, requests, etc.).

### Installation and setup of vulnerability scanning tools (nmap, dnsrecon, wafw00f, etc.).

### Download and integration of any additional libraries needed for tool interactions.

### Terminal Application Code:

### Develop the terminal application using Python 3.

### Ensure the code is organized, well-commented, and follows best coding practices.

### Store the application code in an appropriate directory on the Raspberry Pi.

### Vulnerable Test Website:

### Set up a test website specifically designed to have known vulnerabilities.

### Use a separate test environment or a virtual machine for the website hosting.

### Populate the website with test data and purposely introduce the vulnerabilities you plan to scan for.

### Test Websites:

### Gather a list of various websites with different configurations and security levels for real-world testing.

### Test Cases:

### Define a set of test cases that cover different vulnerabilities and scenarios.

### Create test cases for known vulnerable URLs, open ports, SSL issues, etc.

### Timeline of project execution

### Below is a timeline for the comfortable execution of the project from 1st of August 2023 to 31st of March 2024.

### Phase 1: Project Planning and Research (August 2023)

### Week 1 (August 1 - 7):

### Define project scope and objectives.

### Conduct initial research on Raspberry Pi 4 and relevant cybersecurity tools.

### Week 2 (August 8 - 14):

### Identify and finalize the list of cybersecurity tools to be integrated.

### Plan the hardware requirements for the Raspberry Pi-based device.

### Week 3 (August 15 - 21):

### Create a detailed project plan and timeline.

### Set up the development environment for the Raspberry Pi.

### Phase 2: Hardware and Software Development (September 2023 - November 2023)

### Week 4-6 (August 22 - September 10):

### Acquire the necessary components for the Raspberry Pi device.

### Start developing the hardware prototype.

### Week 7-10 (September 11 - October 8):

### Continue hardware development and testing.

### Begin software development for the integrated terminal application.

### Week 11-14 (October 9 - November 5):

### Integrate selected cybersecurity tools into the application.

### Perform initial testing of the integrated toolset.

### Week 15-16 (November 6 - November 19):

### Conduct the first round of vulnerability scans and fine-tune the application.

### Document the progress and outcomes for the mid-term evaluation.

### Phase 3: Testing and Optimization (November 2023 - January 2024)

### Week 17-20 (November 20 - December 17):

### Conduct comprehensive testing of the device and application.

### Identify and fix any bugs or issues.

### Week 21-23 (December 18 - January 7):

### Optimize the performance of the terminal application.

### Conduct in-depth security testing and penetration testing.

### Phase 4: Finalization and Documentation (January 2024 - March 2024)

### 

### Week 24-27 (January 8 - February 4): -

### Finalize the hardware and software components. –

### Conduct a full-scale evaluation of the integrated toolset.

### Week 28-30 (February 5 - February 25):

### Prepare the final project report and documentation.

### Create user manuals and guides for the Raspberry Pi device and application.

### Week 31-32 (February 26 - March 10):

### Prepare the project presentation for the final demonstration.

### Review and edit the project documentation.

### Phase 5: Project Submission and Presentation (March 2024)

### Week 33-34 (March 11 - March 24): -

### Submit the project documentation and codebase. –

### Practice and refine the project presentation.

### Week 35 (March 25 - March 31):

### Present the final project to the evaluation committee.

### Celebrate the successful completion of the project!

### Please note that the timeline is subject to changes based on the complexity of tool integrations, testing results, and any unforeseen challenges that may arise during the development process. It's important to keep track of progress and be flexible in adjusting the timeline when necessary.

### Ethical and Environmental considerations

### Our project is deeply committed to ethical and environmental considerations. Regarding ethics, we prioritize user privacy and data protection by adhering to best practices and responsible disclosure protocols for any vulnerabilities uncovered during website scans. We approach our project with respect for users, ensuring that its purpose is to enhance website security and safeguard their interests.

### On the environmental front, we have meticulously designed our hardware device, centred around the Raspberry Pi, to optimize energy efficiency and reduce environmental impact. We also make conscious efforts to use sustainable materials in construction and have laid out plans for responsible e-waste management. Moreover, we strive for inclusivity and accessibility, making our project adaptable to diverse user groups. As we continue this journey, we remain committed to fostering a positive social impact and continually improving our ethical and sustainable practices.

### Planning

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No**  **.** |  | **Item** | **Amount (In Rupees)** |
| 1. | Non- Recurring | Official Raspberry Pi 4 Desktop Kit - 8GB Ram(including case and keyboard) | **₹ 9,995 -/** |
|  |  | Waveshare 10.1 Inch Capacitive HDMI LCD Display (B) with Case 1280x800 | **₹ 15,249 -/** |
|  |  | USB C Hub Multiport Adapter- 6 in 1 Portable Aluminum Type C Hub with 4K HDMI Output, USB 2.0/3.0 Ports, SD/Micro SD Card Reader Compatible | **₹ 1,189 -/** |
|  |  | Geekworm Raspberry Pi 4 UPS, X703 V1.2 18650 UPS Shield with Auto Power On Function for Raspberry Pi 4 Model B Only(Support one 18650 Battery) | **₹ 7398 -/** |
| 2. | Recurring | Consumables and Components   * Ultra Alkaline 9V Battery, 12 Pcs | **₹2,532 -/** |
|  | * MCIGICM 2Pcs 830 Point and 2Pcs 400 Point Solderless Breadboard (4cs) | **₹799 -/** |
|  |  | * 120 Pieces Male To Male, Female To Female, Male To Female Dupont Cable Jumper Wire - Multicolour - 40 Pieces Each | **₹258-/** |
| 3. |  | Contingency |  |
|  |  | * Duracell Ultra Alkaline 9V Battery, 12 Pcs | **₹2,532 -/** |
|  |  | * 120 Pieces Male To Male, Female To Female, Male To Female Dupont Cable Jumper Wire - Multicolour - 40 Pieces Each | **₹258-/** |
|  |  | **Grand total** | **₹40,210 –/** |

**Justification for Equipment:**

1. Official Raspberry Pi 4 Desktop Kit – 8GB RAM:

This kit provides the core computing platform for our project. The Raspberry Pi 4 with 8GB RAM ensures sufficient memory to handle complex vulnerability scans efficiently. It comes with essential components such as the case and keyboard, making it a comprehensive solution for our device.

1. Waveshare 10.1 Inch Capacitive HDMI LCD Display (B) with Case 1280x800:

The display is a critical component that allows users to interact with the integrated terminal application effectively. Its large size and capacitive touch capability enhance the user experience, providing a user-friendly interface for vulnerability scanning results and configuration.

1. USB C Hub Multiport Adapter - 6 in 1:

This adapter serves as a bridge between the Raspberry Pi and various external devices. Its multiple USB ports enable easy connectivity for additional tools, peripherals, or external storage, expanding the capabilities of the device.

1. Geekworm Raspberry Pi 4 UPS, X703 V1.2 18650 UPS Shield:

The UPS shield is essential for maintaining uninterrupted power supply to the Raspberry Pi during vulnerability scans. It ensures that power failures or fluctuations do not disrupt the scanning process and safeguards against data loss. Consumables and Components:

1. The ultra-alkaline 9V batteries provide backup power to the UPS shield, ensuring continuous operation even during power outages.
2. The solderless breadboards and Dupont jumper wires offer a convenient and versatile way to connect various components without the need for soldering, making the setup flexible and adaptable.

### Justification for Consumables:

### The consumables used in our project are essential components that contribute to the functionality, ease of use, and adaptability of the device. Below are justifications for the consumables chosen:

### Ultra-Alkaline 9V Batteries:

### These batteries are selected for their reliable power supply to the Geekworm Raspberry Pi 4 UPS. They provide backup power to the Raspberry Pi during vulnerability scans, ensuring continuous operation and preventing data loss in the event of power outages. The use of alkaline batteries ensures long-lasting performance, minimizing the need for frequent replacements and promoting sustainability.

### MCIGICM Solderless Breadboards:

### Solderless breadboards are vital for quick and flexible prototyping during the development phase. They allow us to assemble and connect electronic components without the need for soldering, enabling rapid iteration and testing of different configurations. The ease of use and reusability of these breadboards facilitate a smooth development process.

### Dupont Jumper Wires - Male To Male, Female To Female, Male To Female:

### These jumper wires serve as vital connectors between different components in the project. Their versatility allows us to create custom connections without permanent alterations, making it easy to modify and reconfigure the device as needed. The various types of jumper wires (male to male, female to female, and male to female) provide flexibility in connecting different components and interfaces.

### The chosen consumables align with the project's objectives of creating a functional, adaptable, and user-friendly device for vulnerability scanning. They promote ease of use during development, quick iterations, and efficient testing. Additionally, the consumables contribute to the overall sustainability of the project by reducing electronic waste (e-waste) and minimizing the environmental impact through the use of reliable, long-lasting batteries.

### Justification for Contingency:

### The inclusion of contingency items in our project is essential to ensure smooth and uninterrupted progress throughout the development and execution phases. Justifications for the contingency items are as follows:

### Duracell Ultra Alkaline 9V Batteries (12 Pcs):

### The contingency supply of 9V batteries is a precautionary measure to account for any unexpected power-related issues during vulnerability scanning. Batteries might drain faster than anticipated or encounter unforeseen technical problems. Having a surplus of high-quality Duracell batteries ensures that we can quickly replace depleted batteries and continue the scanning process without delays. This safeguard guarantees the reliability and continuous operation of the device, assuring accurate and comprehensive vulnerability assessments.

### 120 Pieces Dupont Jumper Wires - Male to Male, Female to Female, Male to Female:

### The contingency supply of jumper wires serves as a backup for the primary connectors used in our project. These wires are vital for establishing connections between various components and interfaces. In case of wire breakage, damage, or loss during the development or testing phase, having an ample contingency supply allows us to swiftly replace the affected jumper wires without interrupting progress. This readiness ensures a seamless and efficient development process, preventing potential setbacks.

### By incorporating contingency items, we demonstrate proactive planning and risk mitigation strategies. These items act as safety nets, providing reassurance that unforeseen challenges will not hinder the project's timeline and successful completion. The availability of contingency supplies enhances the reliability and sustainability of our project, allowing us to address any unexpected issues promptly and effectively.

### Expected Outcomes (publications, patents etc.)

### The expected outcomes of this project could go beyond the development of the vulnerability finding device itself. Here are potential outcomes related to publications, patents, and other contributions:

### Technical Documentation:

### The project may result in technical documentation detailing the development process, methodologies, integration of vulnerability scanning tools, and the application's functionality.

### Research Papers or Technical Reports:

### These publications could be submitted to conferences, journals, or technical forums related to cybersecurity, ethical hacking, or software development.

### Open-Source Contribution:

### If the terminal application and associated code are released as open-source software, the project's outcomes could include contributions to the open-source community.

### Patents or Intellectual Property:

### In certain cases, the project might involve the development of novel algorithms, methodologies, or processes that could be patented. If the project results in unique solutions or approaches, seeking patents to protect intellectual property might be considered.

### Educational Resources:

### The project's outcomes could include the creation of educational resources, tutorials, or workshops related to vulnerability scanning, cybersecurity best practices, or Python programming. These resources could be shared with the broader community or used for educational purposes.

### References

### Abdulqader, F. B., Thiyab, R. M., & Ali, A. M. (2017). The impact of SQL injection attacks on the security of databases. In Proceedings of the 6th International Conference on Computing and Informatics (pp. 323-331).

### Kirti Randhe, "Security Engine for prevention of SQL Injection and CSS Attacks using Data Sanitization Technique", IJIRCCE, 2015.

### J. Fonseca. Seixas, M. Vieira, and H. Madeira, "Analysis of Field Data on Web Security Vulnerabilities", IEEE Transaction on dependable and secure computing, vol. 11, no. 2, March/April  2014.

### Jose Fonseca, Marco Vieira, and Henrique Madeira, "Evaluation of Web Security Mechanisms Using Vulnerability and Attack Injection", IEEE transaction on Dependable and secure compu-ting, Vol. 11, No. 5, 2014

### Zoron Djuric, "Black Box Testing tool for Detecting SQL Injection Vulnerabilities", IEEE, 2013.

### Abdul Razzaq, Khalid Latif, H.Farooq ahmad, Ali Hur, Zahid anwar and Peter Charles Bloodsworth, "Semantic security against web application attacks", ACM, 2013.

### Nilima R. Patil and Nitin N. Patil, April, 2012. "A comparative study of network vulnerability analysis using attack graph", in Proceedings of National Conference on Emerging Trends in Computer Technology (NCETCT-2012).

### Peng Li and Baojiang Cui, December, 2010, "A Comparative Study on Software Vulnerability Static Analysis Techniques and Tools", in Proceedings of the IEEE International Conference on Information Theory and Information Security (ICITIS), pp.521-524.

### Nilsson J., 2006, "Vulnerability Scanners", Master of Science Thesis at Department of Computer and System Sciences, Royal Institute of Technology, Kista, Sweden

### H.S. Venter, J.H.P. Eloff, (2004), Vulnerability forecasting -a conceptual model, University of Pretoria, SA, Department of Computer Science.

### Oleg Sheyner, Joshua Haines, Somesh Jha, R. Lippman and J. M. Wing, May, 2002. "Automat-ed generation and analysis of attack graphs", in Proceedings of IEEE Symposium on Security and Privacy.

**Supervisor(s) Head of Department(s)**