Lester Artis

Cell: 757-770-2878 | larti001@odu.edu | LinkedIn | Chesapeake, VA 23321

SUMMARY

Currently, a 4th-year undergraduate student at Old Dominion University, pursuing a Bachelor of Science Degree in Cyber Operations. Seeking to join the workforce at an entry-level position or internship to gain real-world experience, recognized for being hard-working, responsible, and capable of completing tasks both independently and collaboratively, with a flexible schedule, and possessing the skills to secure computer networks and systems from potential cyber-attacks, such as conducting security assessments, implementing security measures and protocols, identifying vulnerabilities, creating and testing incident response plans, and continuously monitoring and analyzing security risks and threats.

EXPERIENCE

United States Military Entrance Processing Command **Aug/2023-Feb/2024**

Student Intern: Information System Security Officer North Chicago, IL

Clearance: Non-Sensitive

Hours per week: 40 hours per week, Full-Time

- Served as a mentee under Jodi L. Goss, Chief of Cybersecurity.
- Gained expertise in entry-level Cybersecurity Officer responsibilities, including managing System Access Requests, monitoring training and certification compliance, and conducting vulnerability scanning and compliance assessments.
- Acquired proficiency in utilizing Cybersecurity Tools commonly used by the Department of Defense (DoD) and the Army, such as Trellix, ENS, ACAS, Evaluate-STIG, and STIG Viewer.
- Assisted in entry-level tasks within eMASS, contributing to the successful completion of Authorization to Operate (ATO) renewals, conducting STIG research and analysis, and drafting government procedural documents.
- Assisted with installation of SolarWinds.
- Installed Nessus Scanner Servers and Drafted, Finalized, and obtained approval for SOP of process.

EDUCATION

Old Dominion University; Bachelor of Science in Cyber Operations; Expected Graduation Date: May 2024

CERTIFICATIONS

CompTIA Security+ (In Progress)
Endpoint Security Solution (ESS) Administrator 201 Certificate; 9/11/2023
Endpoint Security Solution (ESS) Advanced Administrator 301 Certificate; 9/11/2023
Assured Compliance Assessment Solution (ACAS) Certificate; 10/17/2023

TECHNICAL SKILLS

Relevant Coursework: Cybersecurity, Technology, and Society; Cybersecurity Techniques and Operations; Cybersecurity Strategy and Policy; Cyber Law; Computer Literacy: Communication and Information Technology; Foundations of Cybersecurity; Problem Solving and Programming I & II; Intro to Computer Architecture I & II; Intro to UNIX for Programmers; Data Structure and Algorithms; Intro to Discrete Structures; Intro to Computer Theoretical Science; Operating Systems; Cyber Defense Fundamentals; Intro to Networks/Data Communication; Digital Forensics; and Intro to Reverse Software Engineering; Microcontrollers; Embedded Systems; Cybersecurity Ethics; Principles and Practice of Cyber Defense; Cyber Physical System Security; Network Engineering and Design;

Operating Systems: Windows, Linux / Terminal, iOS, macOS, Redhat Linux, Kali Linux, Seed Ubuntu

Microsoft Suites

Programming Languages: C++, Python, Java, HTML, CSS, JavaScript, SQL

Security Tools: Wireshark, Nessus, Nmap, Splunk, PowerShell, Metasploit, Security Onion, pfSense, Maltego,

Burp Suite, Snort, GnuPG, Aircrack-ng, OSSEC, OWASP ZAP, Solarwinds

Security Concepts: Ethical Hacking, Intrusion Detection, Threat & Vulnerability Management, Penetration Testing, Malware Analysis, Incident Response, Information Security Management, Information Assurance, Network Security, Information Technology, Vulnerability Scanning/Assessment/Management, Password Cracking, EDR, CIA Triad, RMF

Firewalls: Packet Filtering, Application-level Gateway, Circuit-level Gateway,

Security Standards: NIST Cybersecurity Framework, Security Information & Event Management (SIEM), Security Orchestration, Automation, and Response (SOAR)

CYBERSECURITY PROJECTS

FILE INTEGRITY MONITOR (FIM)

- An internal control or process was in place to validate the integrity of the operating system and application software files. This was achieved by using a verification method to compare the current file state to a known good baseline.
- Introduced to PowerShell, Hashing, and Automation.
- Created a custom/proof of concept File Integrity Monitor
- Created an integrity baseline of target files/folders using the SHA-512 hashing algorithm.
- Continuously made a comparison of actual files vs baseline and raised alerts if any deviations occurred.
- Sent x-alert via y-means to allow further investigation of potential compromise.

VULNERABILITY MANAGEMENT LAB

- My experience in vulnerability management involved identifying, evaluating, treating, and reporting security vulnerabilities and misconfigurations within an organization's software and systems. This required a strong understanding of various processes, tools, and strategies used in vulnerability management.
- Introduced to:
    - Nessus essentials
    - Virtualization (Oracle Virtual Box)
    - Vulnerability remediation
- Performed the following tasks:
    - Installed and configured Nessus Essentials to perform credentialed vulnerability scan against Windows 10 Hosts
    - Implemented vulnerability management function on sandbox networks:
        - Discover, Prioritize, Assess, Report, Remediate, Verify
    - Conducted vulnerability assessments with Nessus, and remediated vulnerabilities.
    - Developed automated remediation process to preemptively deal with vulnerabilities stemming from Windows updates and third-party software.

CYBERSECURITY DETECTION & MONITORING LAB

- Designed a virtualized homelab network to test vulnerabilities and practice threat detection.
- Utilized Pfsense, Splunk, Kali Linux, Security Onion, and an Active Directory environment to simulate a small enterprise network
- Simulated offensive and defensive tactics for adversary emulation and incident response practice.

Cyber Risk Management Project

- Conducted cyber risk assessment of VM pool on CCI.
- Used Nmap/Nessus and Metasploit to perform the analysis.
- Vulnerability scan carried out on Kali Linux VM using Nmap or Nessus.
- Exploitation was done using Metasploit on Kali Linux for Windows XP, Windows 7, SEED UBUNTU, and Metasploitable 2 VMs.
- Delivered two reports: a status report and a final report.
- Supplementary documentation included automation scripts and relevant screenshots.
- Final report contained an executive summary, a summary of findings, methodology, and detailed findings.
- Used CVE and CVSS scores to rank vulnerabilities and determine risk levels.
- Recommended remediation solutions to fix identified issues.

Virtual Active Directory Environment in PowerShell

- Administered Active Directory: Utilized PowerShell scripts to automate the provisioning, maintenance, and de-provisioning of user accounts.
- Set up Remote Access Server (RAS) features: Established and configured RAS features to support Network Address Translation (NAT) and Port Address Translation (PAT), enhancing remote access capabilities.
- Implemented and maintained Windows DNS and DHCP services: Oversaw the implementation and ongoing maintenance of Windows DNS and DHCP services, ensuring efficient network resource allocation and name resolution.
- Configured Windows File Server, Implemented quotas and NTFS permissions on Windows File Servers, facilitating streamlined file management and bolstering security measures.

AI-Based Malware Detection

- Developed a deep learning model using 1D CNN architecture for malware detection.
- Used Google Colab
- Optimized hyperparameters including learning rate, batch size, input length, and epochs to maximize accuracy.
- Achieved 98.4% accuracy on test data using Adam optimizer.
- Model analyzes byte strings from malware samples as input.
- Used 1D convolutional layers for feature extraction from byte strings.
- Tested different input lengths, finding longer lengths improve accuracy by providing more byte string context.
- Compared optimizers RMSprop and Adam, with Adam yielding higher accuracy.
- Demonstrated experience applying deep learning for cybersecurity and malware detection.

Neural Backdoor Attack

- Implemented a backdoor attack on a CNN model for image classification.
- Used Google Colab
- Created a distinct red circular trigger in the top left corner to trigger misclassification.
- Model achieved 98.5% accuracy on clean test data, comparable to no backdoor.
- Backdoor attack had 86% success rate on causing images with trigger to be misclassified as zeros.
- Customized trigger demonstrated understanding of how to create an effective backdoor.
- Changing visual features like color and shape made trigger stand out.
- Positioning the trigger away from original location showed ability to configure backdoor.
- High success rate demonstrated proficiency in implementing neural backdoor attacks.

Neural Network Backdoor Detection

- Trained a backdoored MNIST image classification model injecting poisoned data with a trigger pattern during training. The backdoored model achieves 98.3% accuracy on clean test data.
- Used Google Colab
- Calculated the Attack Success Rate (ASR) of the backdoored model by testing it with poisoned data. The ASR was 99.9% indicating the backdoor attack is highly effective.
- Investigated the Neural Cleanse (NC) defense algorithm by recovering the trigger pattens. NC successfully identified the backdoor triggers, demonstrating its capability as a defense.
- Tested NC on different trigger sizes. It detected triggers of all tested sizes, showing robustness to trigger dimensions.
- Evaluated the impact NC step size on trigger recovery. Smaller step sizes enabled better precision at the cost of more iterations.
- Assessed the effect of NC iteration number on performance. Increasing the iterations improved trigger recovery but added computational overhead.
- Implemented backdoor triggers with varying opacity. The backdoored model reliably responded to even subtle transparent triggers.
- Experiment highlighted the potency of backdoor attacks and the promise of algorithmic defenses like NC. Careful parameter tuning was needed to optimize NC for operational employment.

Neural Network Adversarial Example Attack

- Used Google Colab
- Experimented with different Fast Gradient Sign Method (FGSM) step sizes and found 0.25 achieved the best trade-off, generating minimal visible perturbations while reducing accuracy to 17.68%. This step size likely found the optimal balance between perturbation scale and attack effectiveness.
- Tested various Projected Gradient Descent (PGD) attack parameters and determined 5 iterations, 0.002 step size, and 0.1 epsilon generated adversarial examples with minimal visible changes while decreasing accuracy to 97.45%. The small step size allowed precise perturbation accumulation over the 5 iterations.
- Constructed a black-box PGD attack using the optimized parameters. It achieved 97.45% accuracy compared to 17.68% for the FGSM attack. FGSM was more effective likely because the single-step black-box transferability was limited.
- Smaller step sizes and moderate iteration counts produced the best performing and subtlest adversarial attacks. Black-box attack transferability presented challenges compared to white-box attacks.

Web Server

- Developed a basic HTTP web server in Python using socket programming language to handle client requests.
- Server opened a TCP socket, bound it to port 6789 to listen for connections.
- Accepted client connections and received HTTP requests via socket recv().
- Parsed the requests to extract the filename being requested.
- Read the file content from the local system.
- Constructed an HTTP response with header and requested file content.
- Sent response back to client over the TCP socket connection.
- Returned 404 error if requested file was not found on server.
- Closed the socker connection after sending response.
- Tested successful serving of HTML files to browser clients.
- Verified 404 errors were returned appropriately when files were missing.
- Utilized Python socket communication for TCP connections, request handling, responses, and error handling.
- Project demonstrated client-server development skills and applied network programming for a basic web server.

Reliable Data Transfer Protocol

- Implemented a Reliable Data Transfer (RDT) protocol in Python to provide reliable delivery over UDP.
- Designed a delayed send approach to provide reliability without complex logic.
- Sender code spaced out UDP packet transmission by 1 second to allow receiver processing time.
- Used Python sockets programming for UDP packet sending and receiving.
- Tested code successfully transmitted a test string between sender and receiver.
- Demonstrated understanding of transport protocols, sockets programming, and achieving reliability over UDP.

Activities

- Member of CS2A – Old Dominion University's Cybersecurity Student Association 2019-Present
- Member of Coastal Virginia Cybersecurity Student Association 2019-Present