530 PROJECT FINAL REFLECTION

ARP Cache Poisoning Tool

Chenrui Niu

Fall 2023

PROJECT DESCRIPTION

The ARP Cache Poisoning Attack Tool project aims to develop a practical tool that simulates ARP cache poisoning attacks and their implications on network security. Address Resolution Protocol (ARP) cache poisoning is a real threat to network security, and this project seeks to provide a learning platform for network security students to understand the techniques, vulnerabilities, and consequences associated with these attacks. The tool will align with the "programming tool" category.

TARGET AUDIENCE

Network security students.

AUDIENCE SKILL LEVEL

Basic understanding of the network and familiarity with the ARP protocol is preferred.

LEARNING OBJECTIVES

After a participant views or interacts with the project, students should be able to

- **Objective 1 –** Demonstrate a comprehensive understanding of ARP cache poisoning attacks, including their mechanisms and potential consequences.
- **Objective 2 –** Simulate ARP cache poisoning attacks in a controlled network environment (virtual machines).

PROJECT IMPACT

The ARP Cache Poisoning Tool project holds significant implications for both its target audience and the broader network security community. By providing a practical and educational tool for simulating ARP cache poisoning attacks, the project aims to contribute valuable insights into network security practices and enhance the understanding of ARP-based vulnerabilities. ARP poisoning remains a persistent threat, and understanding its

mechanisms is crucial for effective defense strategies. The tool serves as a hands-on learning experience, allowing users to gain practical insights into ARP poisoning, a prevalent network attack. And the project has been tailored to accommodate users with beginner levels in network security. For instance, the tool offers a user-friendly interface and step-by-step guidance for installation, setup, and usage. Thus, students will not only acquire knowledge about ARP attacks but also gain practical skills in setting up virtual and controlled environments for testing such attacks.

Expected Outcomes, Evaluation, and Assessment

Deliverables will include the ARP Cache Poisoning Attack tool and a comprehensive user guide. The project's

success will be assessed based on the quality of the tool.

The primary deliverable, the ARP Cache Poisoning Attack tool, will be evaluated based on its functionality by whether it can successfully deploy an ARP cache poisoning attack to a virtual machine. This will include rigorous testing to identify and address any issues, ensuring the tool's stability and functionality. The lack of major unresolved issues will be a key indicator of project success. The tool will feature an interface with clear visualizations of ARP cache poisoning attacks such as live status updates during the attack, which will enhance the user experience by either the GUI or the command line-based output.

The quality of user documentation and tutorials will be assessed based on their clarity, completeness, and effectiveness in guiding users.

A comprehensive user guide will be developed to assist users in navigating and utilizing the ARP Cache Poisoning Attack tool effectively. It will include step-by-step instructions on the installation, setup, and operation of the tool. Moreover, the guide will also include educational materials explaining the underlying principles of ARP cache poisoning attacks, the significance of ARP in network security, and the potential risks associated with these attacks.

Project Evaluation and Outcomes

The project's documentation has been expanded to include a detailed user guide, installation instructions, and a comprehensive explanation of network security concepts related to ARP cache poisoning. The tool is optimized by identifying bugs and performance issues that have been addressed, resulting in a more stable and efficient tool.

In evaluating the tool, a thorough assessment of its functionality has been conducted, affirming its effectiveness in successfully deploying ARP cache poisoning attacks to virtual machines during testing. The tool's interface, primarily through command-line output, stands out for its clarity and user-friendly design, providing clear instructions upon program launch and presenting detailed information about the sent packets. The user documentation, encompassed within the guide, has been evaluated and found to be both clear and comprehensive. It adeptly covers all essential steps, from installation to operation, ensuring users can effortlessly follow instructions and make optimal use of the tool.

The accomplished outcomes of the project include the successful deployment of ARP cache poisoning attacks to virtual machines, showcasing the tool's efficacy in achieving its primary objective. The user experience is notably

enhanced through the tool's interface, whether accessed via GUI or command-line output, offering clear visualizations and real-time status updates during ARP cache poisoning attacks. Additionally, the project delivers a comprehensive user guide that provides step-by-step instructions for the tool's installation, setup, and operation. The user guide goes beyond operational guidance by incorporating educational materials, elucidating concepts such as ARP cache poisoning attacks, the significance of ARP in network security, and associated risks. However, some outcomes remain unmet, particularly with a side function of the tool designed to display the ARP cache table of the target host by its IP, which currently shows only part of the table.

The strengths of this project include the successful deployment of ARP cache poisoning attacks that align with the primary project goal and enhanced user experience by offering clear visualizations and a comprehensive user guide. However, there are identified weaknesses, such as the limited achievement of planned ARP cache displaying features, restricting the tool's scope for users desiring more profound exploration. Additionally, the environment setup is using the Seed Ubuntu 20.04 environment, potentially limiting accessibility for users on different platforms.

Personal Impact

The project topic was selected due to its profound relevance and significance in the realm of network security. ARP cache poisoning attacks represent a prevalent and potentially malicious technique that underscores vulnerabilities within network infrastructures. The choice of this topic was motivated by a desire to delve into a real-world cybersecurity challenge and develop a practical tool that simulates ARP cache poisoning attacks.

Personal Learning Goals

My personal learning goals for this project include gaining an in-depth understanding of network security, enhancing my programming skills, and creating a valuable educational tool. I aim to learn and apply practical knowledge to address real-world security challenges.

Skills and Knowledge Gained

The project significantly contributed to my personal growth and development, specifically by fostering knowledge in setting up and testing virtual environments. Through the research and content development process, I gained a comprehensive understanding of ARP cache poisoning attacks and the intricate mechanisms involved in simulating such attacks and also acquired practical skills related to configuring and testing virtualized networks. The hands-on experience gained during the project deepened my understanding of the intricacies involved in establishing virtual environments for network security testing.

One of my primary personal goals for this project was to deepen my understanding of network security concepts and gain hands-on experience in crafting a practical tool to simulate ARP cache poisoning attacks. This goal was successfully achieved as I navigated through the complexities of the ARP protocol, crafted functional code for the tool, and tested its effectiveness in virtualized environments.

While the majority of my personal learning goals were met, some contributing factors impacted the depth of exploration in certain areas. For instance, time constraints and the evolving nature of cybersecurity challenges also influenced the scope of the project, preventing an exhaustive research and examination.

Lessons Learned

Throughout the project, effective time management became crucial, especially when juggling research, development, and documentation tasks. The need for a well-structured plan and adherence to deadlines became apparent, reinforcing the importance of robust project management skills. Flexibility in adapting to unexpected challenges and adjusting timelines accordingly was a key lesson learned during this phase.

Also, during the project, one notable aspect was the transition in the testing environment from a regular virtual machine to the Seed VM (Virtual Machine). Initially, the project was set up and tested in a standard virtual environment. However, recognizing the advantages of utilizing the Seed VM, which is specifically designed for security education and labs, led to a strategic shift.

This transition to the Seed VM brought about several benefits, including a more standardized and controlled testing environment, aligned with cybersecurity education practices. The Seed VM provided a consistent platform with pre-installed tools, such as Scapy, eliminating the need for additional installations and ensuring a more streamlined testing process.

Future Impact

As I consider potential extensions and improvements, the prospect of incorporating Man-in-the-Middle (MITM) attacks and Denial of Service (DoS) capabilities could be one choice. The inclusion of MITM attacks would elevate the tool's capabilities, allowing users to delve into the complexities of intercepting, manipulating, and even hijacking network communications. Extending the tool to simulate DoS attacks opens up avenues for exploring the impact of overwhelming network resources. Features such as traffic flooding, resource exhaustion, and reflective amplification attacks would offer valuable insights into the dynamics of disrupting network availability.

The project has profoundly influenced my perspective on network security, transforming it from a theoretical concept into a tangible and dynamic field. The hands-on experience of developing a tool to simulate real-world attacks has deepened my understanding of the intricate balance between offensive and defensive strategies in cybersecurity.

Building upon the skills and knowledge gained from this experience, I envision future projects that delve into emerging trends in cybersecurity. Concepts like threat intelligence, zero-day vulnerabilities, and machine learning for anomaly detection captivate my curiosity, and I aspire to explore these areas to contribute to the ever-evolving landscape of network security.

Explanation of Submitted Materials

The ARP poisoning attack tool includes the following materials: ARP Cache Poisoning Attack Tool (source code)

User Guide (brief educational material to explain ARP cache poisoning attacks and documentation for tool usage)

Docker Compose configuration file

Project Content Deliverables

The ARP poisoning attack tool

- Item Description: The primary Python script for the ARP Cache Poisoning Tool.
- File name/path/link: arp poisoning.py
- Relevant information: The script includes functions for ARP cache poisoning, ARP cache display, and user prompts. It utilizes the Scapy library for crafting and sending ARP packets.

User Guide

- Item Description: A comprehensive user guide in PDF format.
- File name/path/link: README.PDF
- Relevant information: The guide covers step-by-step instructions on installing, setting up, and using the ARP Cache Poisoning Tool. It also includes educational content on ARP cache poisoning attacks, their significance, and associated risks.

Docker Compose configuration file

- Item Description: Users can use this file with Docker Compose to set up the required environment by following the provided instructions.
- File name/path/link: docker-compose.yml, obtained from https://seedsecuritylabs.org/Labs_20.04/Networking/ARP_Attack/
- Relevant information: Defines services, networks, and volumes for the Seed Ubuntu 20.04 environment. Configures the necessary containers for testing ARP cache poisoning attacks.

Source Code and READMEs

The ARP poisoning attack tool

- Item Description: The primary Python script for the ARP Cache Poisoning Tool.
- File name/path/link: arp poisoning.py
- Relevant information: The script includes functions for ARP cache poisoning, ARP cache display, and user prompts. It utilizes the Scapy library for crafting and sending ARP packets.

User Guide

- Item Description: A comprehensive user guide.
- File name/path/link: README.MD
- Relevant information: The guide covers step-by-step instructions on installing, setting up, and using the ARP Cache Poisoning Tool. It also includes educational content on ARP cache poisoning attacks, their significance, and associated risks.

Dependencies, Resources & Tools

Python and scapy library (SEED VM has it pre-installed).

Virtualization tool Oracle VirtualBox version 7.0 and SEED-Ubuntu20.04.

References

[1] D. Jacobson. *Introduction to Network Security*. Boca Raton, FL: CRC Press Llc, 2008.

[2] D. Sakhawat, A. N. Khan, M. Aslam. *Agent-based ARP cache poisoning detection in switched LAN environments*. IET Networks, 2019.

[3] "ARP Poisoning: Definition, Techniques, Defense & Prevention". June 1, 2023. [online]. Available: https://www.okta.com/identity-101/arp-poisoning/. [Accessed Nov. 24, 2023].

[4] Danchev, D. "Metasploit project's site hijacked through ARP poisoning". 2008. [online]. Available: https://www.zdnet.com/article/metasploit-projects-site-hijacked-through-arp-poisoning/. [Accessed Nov. 24, 2023].

[5] K. Stratvert. *How to Install VirtualBox - Tutorial for Beginners*, *www.youtube.com*. 2023. [Streaming Video]. Available: https://www.youtube.com/watch?v=nvdnQX9UkMY [accessed Nov. 24, 2023].

[6] M. Bose, "VirtualBox Network Settings: All You Need to Know," *Official NAKIVO Blog*, Jul. 16, 2019. [online]. Available: https://www.nakivo.com/blog/virtualbox-network-setting-guide/. [Accessed Nov. 24, 2023].

[7] P. Biondi, "Introduction — Scapy 2.4.3. documentation," Readthedocs.io, 2019. [online]. Available: https://scapy.readthedocs.io/en/latest/introduction.html. [accessed Nov.15, 2023].

[8] I. Akkila, "Black Hat Python — ARP Cache Poisoning with Scapy," Medium, Sep. 26, 2017. [online]. https://ismailakkila.medium.com/black-hat-python-arp-cache-poisoning-with-scapy-7cb1d8b9d242 [accessed Nov. 22, 2023].