# IP Shuffle:
# Random IP Address Assignment for Network Interfaces

Hunter Thompson
*Eastern Washington University*

Chelsea Edwards
*Eastern Washington University*

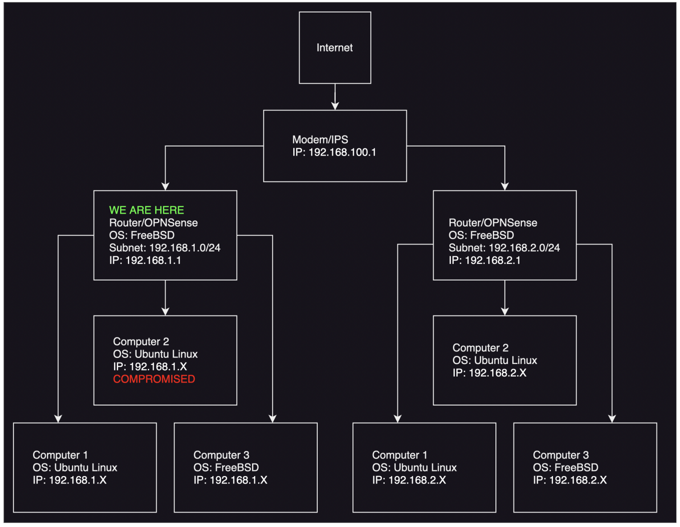
## Abstract

This paper introduces a Bash script designed to dynamically assign a random IP address to a computer's network interface. The script generates a random IP address within a specified range, checks its availability, and ensures proper configuration. It achieves efficient and reliable IP address assignment through distinct functions for IP address generation, availability verification, network configuration validation, and gateway reachability testing. The IP-shuffle script provides a practical solution for scenarios that require dynamic IP address allocation and streamlining network management processes. Its robustness is further enhanced by comprehensive error handling and compatibility with Linux and BSD systems.

## 1 Introduction

The Moving Target Defense (MTD) technique we're working towards is IP shuffling, aimed at complicating lateral movement reconnaissance. This strategy involves dynamically changing the IP addresses of systems on a network. In our model, we have a private subnet containing three virtual machines that perform IP address rotation, periodically or erratically shifting across 254 different IP addresses. Our diagram illustrates a scenario where one of these machines, denoted as Computer 2, has been compromised. By continuously changing IP addresses in an unpredictable manner, IP shuffling impedes attackers' reconnaissance efforts, making it difficult for them to identify and exploit vulnerabilities. The diagram delineates the intricate architecture of our network infrastructure, illustrating the hierarchical arrangement of networks, subnets, and their corresponding topological relationships. Within this schematic representation, the compromised computer is depicted, providing a visual reference to its position within the broader network.

Figure 1: Network Topology Diagram



## 2 Threat Model

Our threat model concerns the scenario in which a system is attacked. Specifically, we focus on the scenario depicted in Figure 1, where three interconnected computers form a subnet, with one of these computers compromised. Within this context, our threat model revolves around an attacker who has successfully gained access to one of the systems, as illustrated in the diagram. Once inside the network, the attacker's assumed objective is to scan other systems to identify vulnerabilities for lateral movement. The provided script, named ip-shuffle, plays a crucial role in this threat model, as it allows for the dynamic assignment of random IP addresses to network interfaces. The attackers' assumed capabilities are that they have basic user access to the compromised system, can perform network reconnaissance via scanning the network, and system persistence.

## 3 System Design

The IP-shuffle script offers a systematic approach to dynamic IP address assignment for network interfaces in Linux and FreeBSD environments. Built around Bash scripting, it seamlessly orchestrates the IP address allocation process. By default, the program runs every three minutes, based on the provided cronjob. During execution, the script dynamically configures the IP address, gateway, network interface details, and other parameters, providing a flexible framework for network configuration. Through dedicated functions such as

```
generate_random_ip (),
check_ip_availability(),
and validate_network_config(),
```

the script ensures that the assigned IP addresses are compatible with the network infrastructure. It also incorporates error-trapping mechanisms and support for common Unix signals to enhance reliability and resilience, safeguarding against potential errors or interruptions. The script's flexibility is maintained through adherence to modular design principles, allowing seamless adaptation to diverse network configurations and environments. However, since the IP addresses are not persistent after a reboot for DHCP-configured machines, the script includes functions like `reset_network()` for error recovery. The IP-shuffle script encapsulates a robust solution for automating network interface configuration tasks, embodying a sophisticated yet accessible approach to dynamic IP address management.

## 4 Evaluation

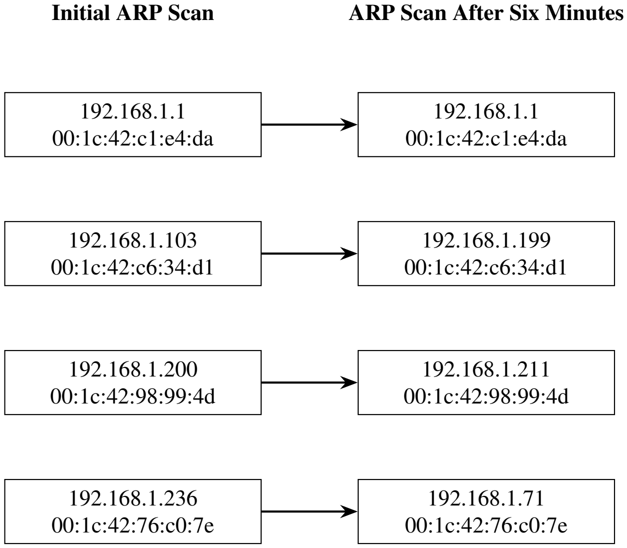
To evaluate the effectiveness of our IP shuffling script, we'll be adding an extra system to the virtual network with a static IP address of `192.168.1.10`. We'll be setting up an OPNsense instance as the default gateway for all virtual machines. This instance will be assigned the IP address `192.168.1.1`, and it will act as a simulated router that provides the DHCP service. As shown in Figure 1, there will be two Ubuntu Linux machines and one instance of FreeBSD. Each machine will receive an IP address on initial startup starting at `192.168.1.100`, assigned by OPNsense. We'll give each system six minutes after startup to begin changing its IP address, after which we'll use an arp-scan to obtain the following output:

```
192.168.1.1     00:1c:42:c1:e4:da       (Unknown)
192.168.1.103   00:1c:42:c6:34:d1       (Unknown)
192.168.1.200   00:1c:42:98:99:4d       (Unknown)
192.168.1.236   00:1c:42:76:c0:7e       (Unknown)
```

We can observe that the IP addresses, with the exception of the OPNsense instance, have been altered from their initially assigned addresses through DHCP. To check if the systems have changed their IPs once more, let's wait for another six minutes and then conduct another ARP scan. Here are the results of the said scan:

```
192.168.1.1     00:1c:42:c1:e4:da       (Unknown)
192.168.1.71    00:1c:42:76:c0:7e       (Unknown)
192.168.1.199   00:1c:42:c6:34:d1       (Unknown)
192.168.1.211   00:1c:42:98:99:4d       (Unknown)
```

Let's cross-check the modified IP addresses with their respective MAC addresses to better understand their altered IPs.



### 4.1 Analysis and Observations

The analysis of the IP-shuffling technique involves evaluating its ability to prevent an attacker from gaining valuable reconnaissance information about the network. Here's a summary of our observations:

- **IP Address Changes:** The IP addresses changed significantly over the two arp-scans, making it difficult for an attacker to establish a static view of the network.
- **MAC Address Consistency:** Each machine retained its MAC address throughout the scans, but the association between MAC addresses and IP addresses changed dynamically.
- **Impact on Reconnaissance:** An attacker would struggle to perform effective reconnaissance and lateral movement as the IP addresses of potential targets keep changing, requiring constant rescanning of the subnet.

### 4.2 Future Work and Limitations

- **MTD Evasion Analysis:** Attackers may attempt to fingerprint devices based on other network characteristics like latency or open ports. Future work could include an analysis of such evasion techniques.

- **MAC Address Fingerprinting Mitigation:** Although the IP addresses of the machines are shuffled, each machine retains its MAC address, which can be used to fingerprint it. Investigating ways to obscure MAC addresses or to randomly change them in addition to IP shuffling could significantly improve the effectiveness of the Moving Target Defense (MTD) strategy.

- **Impact on Legitimate Users:** Changing IP addresses might also impact legitimate network users. Evaluating how legitimate users handle these changes could be a valuable direction for research.

- **Integration with SDN:** Integration of this MTD technique with Software Defined Networking (SDN) could provide more robust and flexible defense mechanisms.

## 5 Conclusion

Moving Target Defense (MTD) is a "game-changing" theme in cybersecurity that involves creating mechanisms and strategies that are diverse, continually shifting, and changing over time to increase complexity and costs for attackers, limit the exposure of vulnerabilities, and increase system resiliency [1].. The IP-shuffle script provides a robust solution for dynamically allocating random IP addresses to network interfaces, a critical component of network security strategies aimed at deterring potential attackers. Leveraging Bash scripting, the IP-shuffle script offers functionalities for generating random IP addresses, checking their availability, and validating network configurations, ensuring efficient and reliable IP address assignment. It also incorporates error-handling capabilities and Unix signal responsiveness, enhancing reliability during execution and strengthening network resilience. Its modular design allows for easy adaptation to different network setups, making it a valuable tool for automating network interface configuration tasks. The IP-shuffle script embodies the concept of IP shuffling, a technique designed to complicate attackers' reconnaissance efforts by constantly changing IP addresses unpredictably. By dynamically assigning random IP addresses, IP-shuffle enhances proactive defense strategies, increasing the difficulty for attackers to identify and exploit vulnerabilities.

## References

[1] Guilin Cai, Baosheng Wang, Xiaofeng Wang, Yulei Yuan, and Sudan Li. An introduction to network address shuffling. In *2016 18th international conference on advanced communication technology (ICACT)*, pages 185–190. IEEE, 2016.