

Attack Lab: Attacks on TCP/IP Protocols

The development of this document is funded by the National Science Foundation's Course, Curriculum, and Laboratory Improvement (CCLI) program under Award No. 0618680 and 0231122. Permission is granted to copy, distribute and/or modify this document.

Overview

The learning objective of this lab is for students to gain first-hand experience on vulnerabilities, as well as on attacks against these vulnerabilities. Wise people learn from mistakes. In security education, we study mistakes that lead to software vulnerabilities. Studying mistakes from the past not only help students understand why systems are vulnerable, why a “seemly-benign” mistake can turn into a disaster, and why many security mechanisms are needed. More importantly, it also helps students learn the common patterns of vulnerabilities, so they can avoid making similar mistakes in the future. Moreover, using vulnerabilities as case studies, students can learn the principles of secure design, secure programming, and security testing.

The vulnerabilities in the TCP/IP protocols represent a special genre of vulnerabilities in protocol designs and implementations; they provide an invaluable lesson as to why security should be designed in from the beginning, rather than being added as an afterthought. Moreover, studying these vulnerabilities help students understand the challenges of network security and why many network security measures are needed. Vulnerabilities of the TCP/IP protocols occur at every layer.

Lab Tasks

In this lab, you need to conduct attacks on TCP/IP protocols. You can use netwox tools and/or other tools in your attacks. Your attacks should be performed on on both Minix and Linux systems. To simplify the “guess” of TCP sequence numbers and source port numbers, we assume that attacks are on the same physical network as the victims. Therefore, you can use sniffers to get those information. The following is the list of attacks that need to be implemented.

1. ARP cache poisoning: change the target host's ARP cache.
2. ICMP Redirect Attack: change the target host's routing table using ICMP redirect message.
3. Attacks on TCP
 - *SYN Flooding Attacks*: use SYN flooding to achieve a Denial-of-Service attack on a target machine.
 - *TCP RST Attacks*: use TCP Reset packet to break an existing TCP connection.
 - *TCP Session Hijacking*: hijack an existing TCP connection.
4. ICMP attacks against TCP
 - *ICMP Blind Connection-Reset Attacks*: The host requirements RFC1122 states that a host SHOULD abort the corresponding connection when receiving an ICMP error message that indicates a “hard error”, and states that ICMP error messages of type 3 (Destination Unreachable), and 4 (fragmentation needed and DF bit set) should be considered to indicate hard errors. Can you use these ICMP error messages to break connections between two machines?

- Do `Linux` and `Minix` conduct validity check on the sequence number of the TCP segment contained in the ICMP payload? For example, can stale ICMP error messages be acted upon by the receivers?
5. TCP Initial Sequence Numbers (ISN) and window size
 - How do `Linux` and `Minix` assign ISNs? Are ISNs predictable?
 - What are the initial window size of `Linux` and `Minix`?
 6. TCP source ports
 - How do `Linux` and `Minix` allocate source ports for TCP connections? Are source port numbers predictable?
 7. Port scanning using Nmap: understand the techniques used by Nmap. Try at least 5 techniques on a target machine, and report your observations.
 8. OS Fingerprinting: use Nmap to fingerprint `Linux`, `Minix`, and `Windows` (if you also use `Windows`); report your observations.

It should be noted that because some vulnerabilities have already been fixed in `Linux`, some of the above attacks will fail in `Linux`, but they might still be successful against `Minix`. You should draw a table in your lab report to summarize the difference between `Linux` and `Minix`, in terms of whether the above attacks are successful.

Optional (up to 30 bonus points) For `Minix`, if an attack is successful, you need to decide whether the vulnerability is caused by implementation errors or by design errors. If it is caused by implementation errors, you need to find the errors from the `Minix` source code, and explain why the errors can lead to the security breaches. You will receive more points if you can fix those errors.

Lab Report

You should submit a lab report. The report should cover the following sections:

- **Design:** The design of your attacks, including the attacking strategies, the packets that you use in your attacks, the tools that you used, etc.
- **Observation:** Is your attack successful? How do you know whether it has succeeded or not? What do you expect to see? What have you observed? Is the observation a surprise to you?
- **Explanation:** Some of the attacks might fail. If so, you need to find out what makes them fail. You can find the explanations from your own experiments (preferred) or from the Internet. If you get the explanation from the Internet, you still need to find ways to verify those explanations through your own experiments. You need to convince us that the explanations you get from the Internet can indeed explain your observations.