Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Network Security

Exercise Sheet 03: Protocol Security

Prof. Dr. Mathias Fischer, August See, Finn Sell

Summer semester 2024

Goals and Objectives For programming tasks, submit the code as an extra file. Please write for each (sub) task approximately how much time you need. You should reach at least 50%.

The goal is to attack a remote network that uses insecure protocols by inspecting and manipulating network traffic.

- Be nice to and on the remote network!
- Always hand in the proof if the server gives you one.
- You don't need to attach a debugger to any provided binaires (and the exercise isn't designed for you to do this).

For the following tasks you will need:

- A network introspection tool e.g. Wireshark
- The ability to run binaries compiled for linux systems

1 Protocol Introspection (25%)

As part of this exercise, you have been given a *secure-download* program that implements a proprietary and secret download protocol.

1. In which ways can you observe the programs beahvior? Which tool can you use to aid your observation?
2. Run the *secure-download* binary and observe its behavior. Describe what you see on the different channels. Can you reconstruct the different steps performed by it?

   Hint: Only TCP traffic on port 7213 is relevant

3. What are the different fields (e.g. length, type tag, crc, protocol identifier, user id, checksums, ...) contained in the protocol spoken by the *secure-download* binary?[1]

   Hint: We know the target network is composed of a gateway server and a ***separate*** *download server. The connection information between the two must therefore be contained in the protocol.*

   Hint: There are strings visible in the protocol. Think about how you can encode them in a binary protocol.

   Hint: The protocol does not contain padding.

2 File Transfer Exploitation (25%)

You have gathered some information about the involved network protocols. the goal is to now exploit weaknesses in the protocol design.

1. Reproduce the *secure-download* programs behavior in a programming language of your choice.
2. Alter your programs behavior to request a different file from the server. Which fields in the protocol did you need to change and to what?

   Hint: This exercise is possible even if you haven't discovered the full protocol in Task 1.

3. Discuss how the protocol could be made more secure and avoid such a manipulation.

3 Proxy Exploitation

In Task 2 you have obtained additional information about the proxy protocol and the construction of its security mechanism. Your goal is to now exploit cryptographic weaknesses in that construction.

3.1 Integrity Protection (25%)

1. Give a summary of what a *Message Authentication Code (MAC)* is for and which security properties it should have.
2. Evaluate the MAC construction algorithm used by the proxy protocol in terms of its security. Explain possible attacks against the it.

[1] See `https://datatracker.ietf.org/doc/html/rfc768` for an example of how protocol header formats are usually described.

3.2 Network Enumeration (25%)

1. Develop a function that can utilize the discovered vulnerability to generate arbitrary proxy protocol headers.
2. Describe how you can use your function to enumerate the network behind the proxy server.
3. Run the developed exploit and describe your findings.

   Hint: The relevant network range is a `/20` subnet and you don't need to vary the port number.