Project

USF UNIVERSITY OF SOUTH FLORIDA

UNDER CONSTRUCTION

This page describes the project for Computer Networks for fall 2018.

The goal of this project is to design, document, implement, and demonstrate an invisible web server with port knocking. The port knocking should be secure with respect to playback attack and denial-of-service attack. You will design the protocol for the port knocking.

Prizes will be offered for the "most secure", "most elegant", and "most innovative" designs (with matching implementations). The logistics of this contest will be discussed in class.

Service to be provided

The service to be provided is a hidden (or invisible) web server that is visible only after a valid port knocking protocol has been completed by a client.

Assumptions about the environment

The environment is the Internet where the web server is assumed to be running on a reachable (so, fixed IP address) host. The host supports the full set of Internet protocols.

Vocabulary of messages

This is for you to design.

Encoding of messages

This is for you to design.

Procedure rules

This is for you to design.

Technical requirements

The technical requirements are listed below.

1. Implement a simple webserver that supports only GET that can be enabled (made visible by opening its port) and disabled (made invisible by closing its port) on command.
2. Create a port knocking protocol that when executed enables the web server for 10 seconds from the time of the port knock. After 10 seconds from the last port knock, the web server is disabled even if in the middle of a file transfer.
    a. The protocol should be secure against playback attacks. It is acceptable to have a shared secret between the client and server system.
    b. The protocol should be reasonably secure against denial of service attacks.
3. The implementation should support up to 10 users.

Project requirements

The project requirements are listed below.

1. The project may be completed by individual students or in teams of two. For teams both students will earn the same grade is all cases.
2. Students may use weblite as their webserver
3. A firewall-based protocol and implementation is not acceptable (that would be too easy).
4. Existing security protocols (e.g., SSH and HTTPS) may not be used in this project.
5. Project deliverables are:
    a. Design document describing the protocol (notably, the messages, message encoding, and message rules). The design document must include a discussion of design choices made with appropriate references to the literature.
    b. Software source code for both client and server side. The client generates the knock, the server receives and processes the knock.
    c. Demonstration of the implementation to the TA in a secheduled 20 minute time period. The TA will have available Windows 10 systems - if the implementation is for any other OS, the students must provide the test bed.

Grading

Project grading is described below. The base grade of 100 is as follows:

- 30 points - a complete and correct protocol design meeting all technical requirements. Points will be deducted if the design is not described to a level where it can be fully implemented by someone "skilled in the art".
- 10 points - a coherent and technically correct explanation of the design choices made. Points will be deducted if the students do not understand the design choices inherent in their design. Points will be deducted if students do not demonstrate an understanding of existing designs/implementations of port knocking.
- 50 points - successful operation of the protocol meeting all technical requirements as demonstrated to the TA.
- 10 points - source code that is readable and understandable (so, documented!).

Additional (aka "extra credit") grade points can be earned as follows:

- 10 points extra - Develop an implementation that does not require any changes to the source code or build of the web server program (and so can use other web servers than just weblite).
- 10 points extra - The goal of this project is to hide a web server from attackers. But, is the web server fully hidden? Explain how the web server could still discovered and attacked. Develop and demonstrate a way to minimize this attack potential.

Hints, notes, and remarks

1. Tumbler can serve as an inspiration for this project.
2. The strength of any hashing algorithms used and/or how passwords (if any) are be stored in the system are not of much emphasis in this project - the emphasis is on the protocol. This project is not the forum for debating the strengths of hashing algorithms (e.g., MD5 versus SHA).
3. You may choose to work in any language that you think is best suited for this project. The use of "C" is recommended.
4. You may work together within your team of two students. Submitted code (between teams) will be compared for copied code. Copying code is cheating - cheaters will earn (at minimum) a zero in the project and (at most) an FF in the class.
5. If and when in doubt, ask a question.
6. Have fun! As with all things, this project is what you make of it  .